BEES: Bayesian Ensemble Estimation from SAS, a SASSIE-web Module

Samuel Bowerman[1], Joseph E. Curtis[2], Joseph Clayton[1], Emre H. Brookes[3], and Jeff Wereszczynski[1]

[1]Department of Physics and the Center for Molecular Study of Condensed Soft Matter, Illinois Institute of Technology, Chicago, IL 60616
[2]NIST Center for Neutron Research, National Institute of Standards and Technology, Gaithersburg, MD 20899
[3]University of Texas Health Science Center, San Antonio, TX 78229

August 24, 2018

1 Abstract

Many biomolecular complexes exist in a flexible ensemble of states in solution which are necessary to perform their biological function. Small angle scattering (SAS) measurements are a popular method for characterizing these flexible molecules due to their relative ease of use and ability to simultaneously probe the full ensemble of states. However, SAS data is typically low-dimensional and difficult to interpret without the assistance of additional structural models. In theory, experimental SAS curves can be reconstituted from a linear combination of theoretical models, although this procedure carries significant risk of overfitting the inherently low-dimensional SAS data. Previously, we developed a Bayesian-based method for fitting ensembles of model structures to experimental SAS data that rigorously avoids overfitting. However, we have found that these methods can be difficult to incorporate into typical SAS modeling workflows, especially for users that are not experts in computational modeling. To this end, we present the "Bayesian Ensemble Estimator from SAS" (BEES) program. Two forks of BEES are available, the primary one existing as module for the SASSIE webserver, and a developmental version that is a standalone python program. BEES allows users to exhaustively sample ensemble models constructed from a library of theoretical states and to interactively analyze and compare each model's performance. The fitting routine also allows for a secondary data sets to be supplied, thereby simultaneously fitting models to both SAS data as well as orthogonal information. The flexible ensemble of K63-linked ubiquitin trimers is presented as an example of BEES' capabilities.

2 Introduction

Biological molecules rely heavily on their conformational dynamics to conduct their cellular function, and the characterization of these flexible ensembles of states remains a key challenge in modern biophysics.[1] As a result, many different experimental and computational techniques have been developed to probe and model configurational ensembles. Of these, small angle scattering (SAS) measurements are an increasingly popular technique due to their relative ease of use and ability to simultaneously probe the full solution ensemble.[2,3] Moreover, SAS measurements are able to probe systems at room temperature, free from packing forces induced by the lattice and cryogenic effects of crystallography, and they can measure the solution of states in both equilibrium ensembles and time-dependent processes,[4] such as protein and RNA folding,[5,6] or the allosteric coupling of enzymatic activity and large-scale domain movement.[7,8] However, the low-dimensional nature of SAS data can often cause the interpretation of scattering profiles to be relatively difficult, and reconstituting a three-dimensional molecular structure solely from scattering curves can often be misleading, as multiple reconstitutions of varying shapes may result from the same scattering profile.

In contrast, model structures can also be identified from all-atom or coarse-grained simulations, and their calculated scattering profiles can be compared against empirical curves.[9–12] Since SAS profiles are measurements of the full solution ensemble and therefore may not be fully described by a single structural state, these *in silico* profiles can also serve as a basis set to construct an ensemble model through a linear combination of states.[13–16] While this ensemble reconstitution approach is conceptually straightforward, in practice it can be quite difficult to identify the "best" ensemble model. For instance, it is not known *a priori* what the number of underlying states should be in the ensemble. It is also possible for ensemble models to overfit experimental data through the inclusion of too many underlying populations. Furthermore, altogether different combinations of states may yield similarly performing models, in respect to their goodness-of-fit values.

For these reasons, a Bayesian-based approach has many advantages over more traditional methods. For instance, Markov Chain Monte Carlo posterior sampling methods will not only estimate model parameters but will also allow for the direct assessment of their errors.[17] Moreover, Bayesian formalism allows for the comparison of a population of models as a solution to parameterization, rather than only identifying a single set of parameters.[18–20] This is exceptionally useful for SAS modeling, where information regarding the model is underdetermined. However, the ability to construct a large population of solutions can also be a disadvantage, as both the computational resources to construct a complete array of model parameters, as well as tools for comparing models, can be daunting for many systems.

To this end, we have developed an iterative Bayesian method to use small angle scattering (SAS) using either x-ray scattering (SAXS) or neutron scattering (SANS) profiles to re-weight the population of states from molecular dynamics trajectories. This approach, which is an extension of the BSS-SAXS technique,[13] compares solution ensembles of a variety of sub-ensembles from a combination of candidate states. Previously, we have used this method to fit ensembles of covalently linked ubiquitin trimers, and we observed that the algorithm could produce ensemble models that robustly resisted overfitting.[21]

Here, we present an update to this method as an open source program called "Bayesian Ensemble Estimator from SAS" (BEES, henceforth). Two versions of this code have been developed. The primary version, and the focus of this manuscript, is an open-access module on the SASSIE-web server, which provides a convenient graphical user interface for controlling the module. The BEES-SASSIE module is designed for users that are both new and experienced in biophysical modeling, and, through SASSIE, it provides access to the computational resources required to calculate and analyze large combinations of states. The second, developmental, version is a stand-alone python code that is designed to be run from the command line, and is intended for experienced computational scientists. We also provide two example use cases, one in which we fit profiles of K63-linked ubiquitin trimers to SAXS data alone and another in which we add a second data set to the fitting procedure.

3 Design and Implementation

3.1 SASSIE-web Framework

The primary version of BEES is contained within the SASSIE-web server. SASSIE-web utilizes the GenApp framework to create a graphical user interface (GUI) and facilitate the use of several independent scientific applications into modular workflows.[22,23] These applications, including the BEES module, are written primarily in Python (v2.7.13) and utilize libraries contained within the Anaconda distribution. BEES, both in the SASSIE and stand-alone versions, makes use of the mpi4py (v2.0.0),[24] NumPy (v1.13.1),[25] and SciPy (v0.19.1)[26] packages to optimize performance. It also utilizes the Bokeh (v 0.12.16)[27] library to create interactive plots and html files to assist with the interpretation of the ensemble models identified by the BEES algorithm.

3.2 BEES Algorithm

The BEES algorithm is designed to find the smallest possible ensemble that accurately describes the experimental data. This algorithm is briefly presented here (Fig 1), but further details can be found in the supplemental text and elsewhere.[21] In short, experimental data are gathered and post-processed prior to using the BEES module. For example, users may wish to screen their data for low-q beam smearing effects or to extrapolate their scattering profile to I(0), the later of which can be accomplished using the "Data Interpolation" module of SASSIE. A collection of theoretical profiles for candidate solution states are also input to the module in the form of a ZIP archive. These profiles can be calculated within SASSIE via the "SasCalc" module, or they can be computed from standalone programs such as Crysol,[28] FoXS,[29] or many other scattering prediction software.[30–33] The only requirements of the BEES algorithm is that the theoretical profiles are formatted as two-column files, where the first column is q and the second column is I(q), and that the theoretical q-values are identical to the experimental q-space.

Once initiated, the BEES routine first determines the goodness-of-fit values of each individual profile. It then identifies all possible sub-bases containing combinations of two theoretical profiles, and it conducts a Bayesian Monte Carlo routine on each combination to identify the population of states in each sub-basis. Each Monte Carlo routine is conducted according to user-defined parameters: number of walkers, number of Monte Carlo iterations per walker, and amount of population change per iteration. Notably, the BEES likelihood function (L) includes the ability to simultaneously fit the scattering profiles and an auxiliary set of measurements:

$$L = \mathrm{e}^{\chi^2_{\mathrm{total}}/2.0} \tag{1}$$

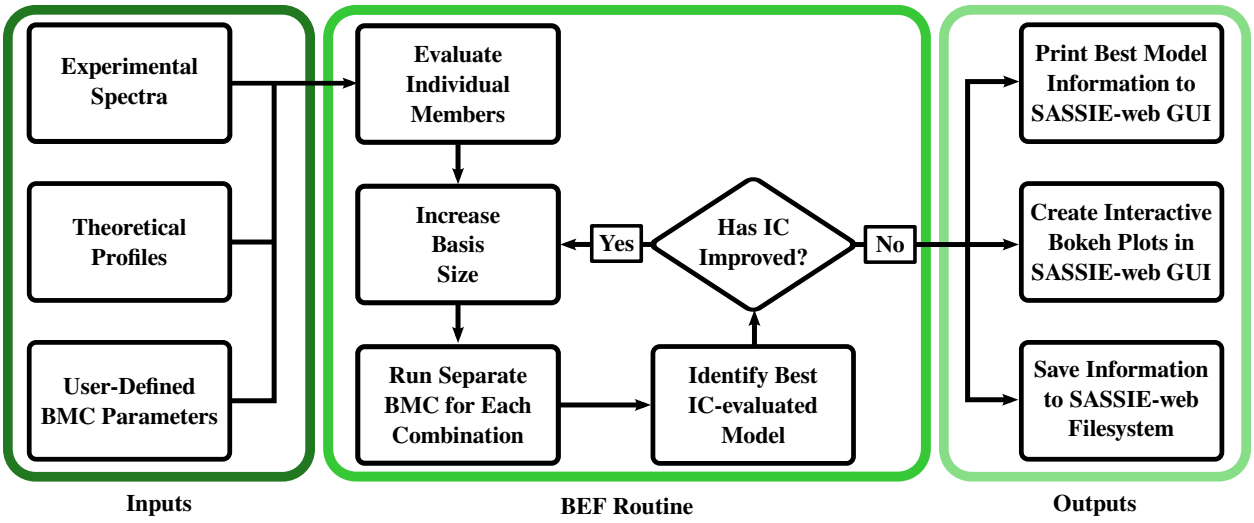


Figure 1: Workflow schematic of the BEES routine. Users upload empirical data and analogous theoretical profiles for potential ensemble members to SASSIE-web, as well as set several parameters associated with the Bayesian Monte Carlo (BMC) parameter search. After the performance of individual theoretical state is evaluated, ensemble populations are fit by BMC routines conducted iteratively on increasing sized sub-ensembles, until the addition of another member population does not improve the IC value and overfitting is observed. Alternatively, users can bypass the IC-comparison step to compare all possible combinations of states. The routine then relays information regarding the resulting models to the SASSIE-web GUI and further stores the information on the SASSIE-web filesystem, for subsequent retrieval by users.

where the total model goodness-of-fit (χ^2_{total}) is the linear combination of the model scattering goodness-of-fit (χ^2_{SAS}) and the model goodness-of-fit to the auxiliary data set (χ^2_{aux}): $\chi^2_{total} = \chi^2_{SAS} + \chi^2_{aux}$.

Once the ensemble of states for each two-member sub-basis has been identified, the best two-member state is selected in accordance to the information criteria (IC) selected by the user, either the Akaike information criterion[34] or the Bayesian Information Criterion[35] (see Supporting Information for more details). If the IC value of the best two-member state is worse than that of any single theoretical profile, then the module reports the best single profile as the most likely model. However, if the IC value of this two-member state is instead an improvement over all individual profiles, then the BEES module conducts the Bayesian Monte Carlo routine on every three-member sub-basis, and the best three-state IC value is similarly compared to the two-state ensemble. This iterative increase in sub-basis size and comparison of IC values is conducted until either the IC metric does not improve or every possible combination of states is considered. Alternatively, users also have the option to override the IC-comparison and force the construction of all combinations of sub-ensembles. Once the desired number of models have been identified, the BEES module will also calculate each model's "relative performance" metric to determine its likelihood over the best IC-identified model:[36]

$$RP(m) = \mathrm{e}^{(\mathrm{IC_m} - \mathrm{IC_o})/2} \tag{2}$$

where $RP(m)$ and IC_m are the the relative performance and IC values of model m, and IC_o is the minimum IC value of all observed models. The relative performance metric is more commonly known as the relative likelihood of a model. Here, we opt for the change in nomenclature to assist non-experts in the interpretation of the metric, as well as to avoid confusion with the likelihood function used by the Bayesian Monte Carlo parameter fitting routine. While the relative performance provides a quantitative result, it is admittedly an approximation of the more rigorous Bayes Factor.[35,37] As such, it is intended to be interpreted loosely and to assist the user in applying their intuition toward the performance of alternative ensembles to the best identified one.

Once the best model has been identified, the BEES module prints text information regarding ensemble members of the IC-identified model, its model population weights, goodness-of-fit information for the full

ensemble model and each individual, and the IC value of the model. This information is also stored to the filesystem as a text file, along with a separate two-column file containing the theoretical scattering profile of the ensemble. If an auxiliary set of experimental data was supplied to the module, then a text file containing the model spectra in this auxiliary dimension is also printed. Beyond the best identified model, information regarding every model identified for each sub-basis is also saved to the filesystem. In the SASSIE-web GUI, plots of the model ensemble fit to the experimental data, along with the associated residual errors, are displayed at the bottom of the page once the fitting routine is completed. These plots are also included in a multi-tab HTML page that is saved by the module, and this page provides graphical and table presentations to allow users the ability to compare different models and performances. Users can download this HTML page and access it locally in their preferred web browser at any time, allowing them to compare previous results with future BEES analyses.

3.2.1 BEES Parallelization Scheme

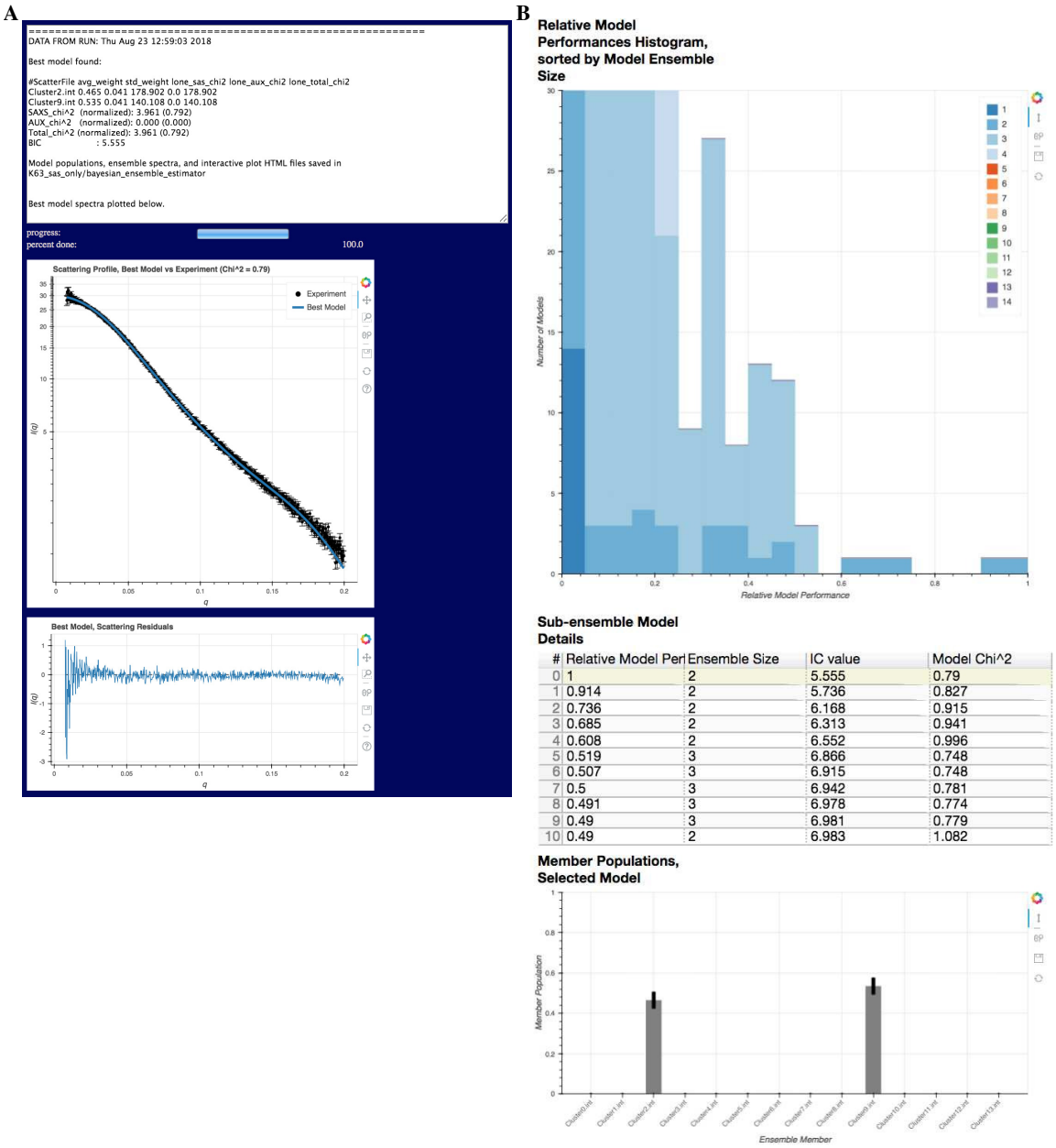
Since the BEES algorithm identifies the best ensemble model as a sub-basis of the full list of supplied potential states, the required CPU time for a single BEES analysis relies mainly on two parameters: the total number of theoretical profiles and the sub-basis size at which over-fitting is observed. While the number of theoretical profiles can be reduced by conducting a clustering analysis or by screening the profiles in accordance to orthogonal information before being input to BEES, the number of states before overfitting is observed is not known *a priori* and may likely be affected by the signal-to-noise ratio of the experimental data. However, because the Monte Carlo reweighting procedure of each sub-ensemble is independent of the others, the algorithm benefits substantially from a trivial parallelization scheme. Therefore, BEES utilizes mpi4py to separate sub-ensembles at each basis-size iteration in an Open MPI-compatible manner.[24,38] In this way, the BEES module on SASSIE-web provides users that possess limited parallel resources with the opportunity to conduct ensemble re-weightings on large basis sizes ($n > 10$ members) in a reasonable amount of time (∼2 hours on 6 processors for 14 candidates that overfit at three member states, ∼36 hours on 6 processors for calculation of the full array of models; see Supporting Information for more details on parallel performance). Similarly, users of the command line version may take advantage of their local parallel computational resources to reduce the required computational time.

4 Results

Here, we describe the usage of BEES and its resulting data, with a focus on the SASSIE-based BEES module. The necessary data files for this test set are included in the Supporting Information. Users can thereby re-create the analyses presented here by unpacking the archive locally and uploading the relevant files for each case to the BEES module in SASSIE-web. In the first example, we model the populations of states of K63-linked ubiquitin trimers using clusters identified from accelerated molecular dynamics trajectories.[21] In the second example, we showcase the effects of simultaneously fitting the SAS spectra and an auxiliary data set by including simulated measurements of an inter-domain distance and angle.

4.1 Building Ensembles of SAS Data

The BEES module requires the user to upload two files: the experimental scattering curve and a ZIP archive of theoretical profiles. Also contained within the ZIP archive is a one-column text file that lists the profiles that should be considered in the BEES fitting routine. In addition to providing these files, the user will also define the D_{max} of the molecule, which can be determined from the experimental profile using pre-existing software.[39] Here, a D_{max} of 83.6 Å was determined using the Shanum program of the ATSAS package.[40] Furthermore, five Monte Carlo walkers were used for each sub-basis ensemble, and each walker was conducted for 10,000 iterations. The first 1,000 iterations were neglected when determining the model populations so as to remove any influence of the randomly selected initial values from the final result. Parallel processing can also be used (here, 6 processors were used), but using multiple processors will only enhance the speed



#	Relative Model Perf	Ensemble Size	IC value	Model Chi^2
0	1	2	5.555	0.79
1	0.914	2	5.736	0.827
2	0.736	2	6.168	0.915
3	0.685	2	6.313	0.941
4	0.608	2	6.552	0.996
5	0.519	3	6.866	0.748
6	0.507	3	6.915	0.748
7	0.5	3	6.942	0.781
8	0.491	3	6.978	0.774
9	0.49	3	6.981	0.779
10	0.49	2	6.983	1.082

Figure 2: (a) SASSIE-web GUI outputs of the BEES module. (top) Text output displaying the contributing populations of the best IC-identified ensemble and the associated error in population estimates, as well as goodness-of-fit for each member. Total model goodness-of-fit and IC value are also printed by the module. (middle) Ensemble scattering profile of the best identified model, shown in blue, fit to the experimental spectrum, shown in black. (bottom) Residual errors of the best model against the experiment. (b) The third tab ("Compare All Models") of the saved HTML file, which contains the relative performances histogram as well as a table of all the constructed ensemble models and their relative performance, ensemble size, selected IC metric, and goodness-of-fit values. Selecting a particular model in the table will also visualize the constituent populations on the bar graph below (best identified model selected here). The full interactive HTML file can be accessed by downloading the "K63_SASonly.html" file from the Supporting Material.

of the calculation and has no effect on the final result. In addition, the full array of sub-ensembles has been calculated to display the depth of analysis available. In this example, truncation of the algorithm via the IC parameter would save a significant amount of computational time without effecting the best IC-identified model; however, models with lower χ^2_{free} would not have been observed. At the conclusion of the BEES routine, the best identified model is printed to the SASSIE-web GUI (Fig 2A), and an interactive plot interface is spawned (Fig 2B). In this example, the best model is a two-state solution that is approximately equal parts clusters 2 and 9. This model has a χ^2_{free} of 0.79 and a BIC value of 5.56. While this is the best model according to BIC comparisons, roughly 50 models of varying sizes possess better χ^2_{free} values, and the model with the best goodness-of-fit ($\chi^2_{\text{free}} = 0.74$) is a 4-member state comprised of clusters 2 (∼45%), 4 (∼22%), 10 (∼15%), and 11 (∼18%). This lowest χ^2_{free} model has an IC value of 8.46, which yields a relative performance of 0.23 when compared to the IC-identified two-state model. As such, the improved χ^2_{free} value of this model is unwarranted, as it is likely the result of overfitting by too many basis members. Indeed, inspection of the model performance histogram (Figure 2B, top) shows that the best performing models are largely two-state solutions, but some three-state solutions perform moderately well. Furthermore, many of the two- and three-state solutions are a significant improvement over each of the single-state models.

4.2 Building Ensembles with Auxiliary Data

Some users may desire to use BEES to build theoretical solution states by fitting solely to SAS data, and then use these states to predict measurements of future experiments. However, others may already possess such data and may prefer to create models that are consistent with both these measurements as well as the observed SAS profiles. For example, an experimenter may desire to simultaneously model both a scattering profile and a catalogue of NMR-derived distances. For the benefit of this class of users, we have included this functionality within the BEES module. To demonstrate how including such data might affect the modeling results, we discuss here an extension of the previous tri-ubquitin example in which we provide a simulated data set that contains the ensemble-averaged center-of-mass distance between distal monomers and the angle formed by the trimer arrangement (Fig 3). These data were created by taking the ensemble-averaged measures of the best model from the previous example with the inclusion of a Gaussian noise factor, resulting in a target distance of 53.0 ± 1.6 Å and a target angle of 117.7 ± 8.3°. Inputs to the BEES routine are identical to the previous example, with the exception of the auxiliary data set.

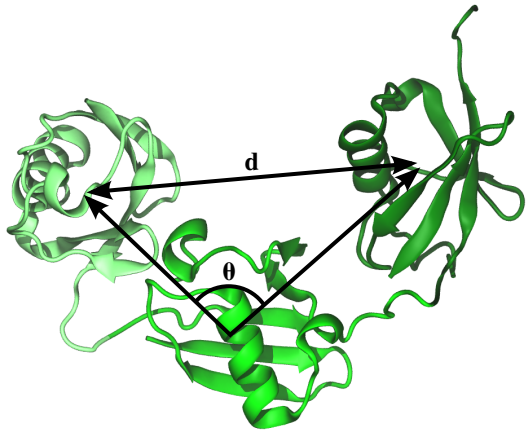


Figure 3: A visual representation of the two auxiliary measurements included in the second BEES routine. Both the distal monomer separation distance (d) and angle (ϑ) are measured in accordance to each monomer's center-of-mass. The target value for the distance was 53.0 ± 1.6 Å, and the target angle was 117.7 ± 8.3°.

With the addition of the distance and angle measurements, we find a shift in the best IC-identified model. While it is still a two-state solution, the contributing members are now clusters 3 (43 ± 5%) and 4 (56 ± 5%). This model yields a χ^2_{total} of 0.80, with a χ^2_{free} of 0.96 and a χ^2_{aux} of 0.39. As was the case in the last example, there are a plethora of models containing three or more members in which better goodness-of-fits are observed, and the best goodness-of-fit model is a mixture of clusters 2, 4, and 11 and has a χ^2_{total} of 0.64. While this model is arguably a better fit to the data than the two-state ensemble of clusters 3 and 4, the IC value of this model is larger due to the addition of a third population. As such, this model is only the eighth most probable model, and possesses a relative performance of 0.65.

When we inspect the ten best ensembles, we once again find the best model from the previous example, which possesses a total χ^2_{total} of 0.80, a χ^2_{free} of 0.78, and a χ^2_{aux} of 0.84. Differences between the exact values of the χ^2_{free} metric in this example and the previous example are a result of the random-sampling nature of the χ^2_{free} metric, but these values are statistically indistinguishable. Similarly, the total goodness-of-fit in the clusters 3 and 4 ensemble is comparable to the ensemble containing clusters 2 and 9. As both models are two-state solutions, this results in very similar IC metrics and a relative performance value of 0.98, which suggests that neither model is significantly more accurate than the other. However, the 3+4 ensemble significantly outperforms the 2+9 ensemble in the context of the distance and angle measurements, while the 2+9 ensemble is a better fit to the the scattering curve.

5 Discussion

Here, we have presented the Bayesian Ensemble Estimator from SAS (BEES) program and highlighted its use as a module in SASSIE-web with two example use cases. In the first example, we used the module to reweight states of K63-linked tri-ubiquitin that were obtained from accelerated molecular dynamics simulations. The BEES module identified a two-state solution as the model that best balanced the fit to experimental data with the fewest number of states. However, the analysis also found a plethora of models that had improved goodness-of-fits to the experimental scattering profile, but each of these models had more ensemble members than the two-state solution. The BEES module provides users with a convenient interface to both find and compare these other candidate ensembles with the IC-identified best state. This allows researchers the option to either rigorously trust the IC statistics to identify the most appropriate scattering model or to use the "ensemble of ensembles" constructed by the BEES module to guide their understanding of datasets separate from the fitting procedure.

The second use case discussed here demonstrated how BEES performs when simultaneously fitting populations to both SAXS and auxiliary data (here, simulated distance and angle measurements). In this example, and the best identified model was still a two-state solution. However, a three-member ensemble was observed to have a better goodness-of-fit, but the improvement to χ^2_{total} was not sufficient to also improve the IC parameter, yielding a relative performance of 0.64. Since the two-state solution has strong agreement with both measurements (χ^2_{free}, $\chi^2_{\text{aux}} < 1.0$), this relative performance value suggests that a conservative estimate for the solution ensemble would favor the two-state model over the χ^2 three-state case. However, the performance is of high enough quality that this ensemble could also be considered as a solution for future measurements. In this way, we emphasize that the relative performance metric should aide the intuition of researchers, rather than completely replace it.

As we have shown, BEES can be used to construct ensemble models of scattering data from a library of candidate states, and the iterative algorithm of BEES quantitatively resists overfitting of the data from the addition of unnecessary populations. The program is designed for use by both new and expert users of computational ensemble modeling, with a web-based module for the SASSIE-web platform that provides structural and computational biophysicists with the resources necessary to construct molecular models in a Bayesian-based manner. Furthermore, it provides visual tools for quickly interpreting not only the quality of the best IC-identified model, but also for the full ensemble of sub-basis models available from the candidate populations. This feature allows users to inspect many different potential solutions and to compare their ability to model both SAS and auxiliary data sets. In this way, BEES serves the intuition of structural researchers in building ensembles of states for their systems of interest.

6 Future Directions and Availability

While the current BEES implementation assumes uniform prior distributions across all parameters, future versions of the module will include the ability to provide user-defined prior distributions to ensemble populations as an option for advanced users. This will allow users to access the full capability of the Bayesian inference approach, and populations determined from a previous BEES run may potentially serve as prior distributions for future BEES analyses.

Furthermore, the BEES SASSIE-web module is presently limited to fitting every combination of ensembles containing up to 14 candidate states in a reasonable timeframe (∼36 hours on 6 processors and minimal server load). In the future, SASSIE-web will be integrated as a scientific gateway to XSEDE supercomputing resources, allowing for even larger model ensemble libraries to be considered. By combining with highly parallel computing environments, BEES users will be able to find minimum-sized solution ensembles from dozens of candidate members, as well as have the ability to construct the complete array of sub-ensembles from even larger candidate pools.

SASSIE-web, including the Bayesian Ensemble Estimator from SAS (BEES) module, is open source, and persistent user accounts are available (https://sassie-web.chem.utk.edu/sassie2/). The developmental BEES implementation is licensed under the GNU GPLv3 license and is available from a GitHub repository `https://github.com/WereszczynskiGroup/BEES`.

7 Acknowledgements

The authors would like to thank Dr. Susan Krueger for valuable discussions in designing the plotting interface. EB's work is supported by National Science Foundation grant number OAC-1740097 and NIH grant GM120600 (to Borries Demeler), SB, JC and JW are supported by the National Institute of General Medical Sciences (NIGMS) of the National Institutes of Health under award number R35GM119647. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. This work benefited from CCP-SAS software developed through a joint EPSRC (EP/K039121/1) and NSF (CHE-1265821) grant as well as interactions and data collection at the Biophysics Collaborative Access Team, which is supported by NIGMS grant P41GM103622.

References

[1] Henzler-Wildman, K.; Kern, D. Nature **2007**, 450, 964–972.

[2] Boldon, L.; Laliberte, F.; Liu, L. Nano Rev **2015**, 6, 25661.

[3] Trewhella, J. Curr. Opin. Struct. Biol. **2016**, 40, 1–7.

[4] Graceffa, R.; Nobrega, R. P.; Barrea, R. A.; Kathuria, S. V.; Chakravarthy, S.; Bilsel, O.; Irving, T. C. J Synchrotron Radiat **2013**, 20, 820–825.

[5] Nasedkin, A.; Marcellini, M.; Religa, T. L.; Freund, S. M.; Menzel, A.; Fersht, A. R.; Jemth, P.; van der Spoel, D.; Davidsson, J. PLoS ONE **2015**, 10, e0125662.

[6] Plumridge, A.; Katz, A. M.; Calvey, G. D.; Elber, R.; Kirmizialtin, S.; Pollack, L. Nucleic Acids Res. **2018**,

[7] Cross, P. J.; Dobson, R. C.; Patchett, M. L.; Parker, E. J. J. Biol. Chem. **2011**, 286, 10216–10224.

[8] Fetler, L.; Kantrowitz, E. R.; Vachette, P. Proc. Natl. Acad. Sci. U.S.A. **2007**, 104, 495–500.

[9] Howell, S. C.; Qiu, X.; Curtis, J. E. J Comput Chem **2016**, 37, 2553–2563.

[10] Datta, S. A.; Curtis, J. E.; Ratcliff, W.; Clark, P. K.; Crist, R. M.; Lebowitz, J.; Krueger, S.; Rein, A. J. Mol. Biol. **2007**, 365, 812–824.

[11] Chen, P. C.; Hub, J. S. Biophys. J. **2014**, 107, 435–447.

[12] Hub, J. S. Curr. Opin. Struct. Biol. **2018**, 49, 18–26.

[13] Yang, S.; Blachowicz, L.; Makowski, L.; Roux, B. Proc. Natl. Acad. Sci. U.S.A. **2010**, 107, 15757–15762.

[14] Tria, G.; Mertens, H. D.; Kachala, M.; Svergun, D. I. IUCrJ **2015**, 2, 207–217.

[15] Pelikan, M.; Hura, G. L.; Hammel, M. Gen. Physiol. Biophys. **2009**, 28, 174–189.

[16] Schneidman-Duhovny, D.; Hammel, M.; Tainer, J. A.; Sali, A. Nucleic Acids Res. **2016**, 44, W424–429.

[17] Hines, K. E. Biophys. J. **2015**, 108, 2103–2113.

[18] Fisher, C. K.; Huang, A.; Stultz, C. M. J. Am. Chem. Soc. **2010**, 132, 14919–14927.

[19] Voelz, V. A.; Zhou, G. J Comput Chem **2014**, 35, 2215–2224.

[20] Ge, Y.; Voelz, V. A. J Phys Chem B **2018**, 122, 5610–5622.

[21] Bowerman, S.; Rana, A. S. J. B.; Rice, A.; Pham, G. H.; Strieter, E. R.; Wereszczynski, J. J Chem Theory Comput **2017**, 13, 2418–2429.

[22] Brookes, E. H.; Anjum, N.; Curtis, J. E.; Marru, S.; Singh, R.; Pierce, M. Concurrency and Computation: Practice and Experience **2015**, 27, 4292–4303.

[23] Perkins, S. J.; Wright, D. W.; Zhang, H.; Brookes, E. H.; Chen, J.; Irving, T. C.; Krueger, S.; Barlow, D. J.; Edler, K. J.; Scott, D. J.; Terrill, N. J.; King, S. M.; Butler, P. D.; Curtis, J. E. J Appl Crystallogr **2016**, 49, 1861–1875.

[24] Dalcín, L.; Paz, R.; Storti, M. Journal of Parallel and Distributed Computing **2005**, 65, 1108 – 1115.

[25] Oliphant, T. E. Computing in Science Engineering **2007**, 9, 10–20.

[26] others,, et al. SciPy: Open source scientific tools for Python. 2001–; `http://www.scipy.org/`, [Online; accessed ¡today¿].

[27] Bokeh Development Team, Bokeh: Python library for interactive visualization. 2014.

[28] Svergun, D.; Barberato, C.; Koch, M. H. J. Journal of Applied Crystallography **1995**, 28, 768–773.

[29] Schneidman-Duhovny, D.; Hammel, M.; Sali, A. Nucleic Acids Res. **2010**, 38, W540–544.

[30] Stovgaard, K.; Andreetta, C.; Ferkinghoff-Borg, J.; Hamelryck, T. BMC Bioinformatics **2010**, 11, 429.

[31] Ravikumar, K. M.; Huang, W.; Yang, S. J Chem Phys **2013**, 138, 024112.

[32] Virtanen, J. J.; Makowski, L.; Sosnick, T. R.; Freed, K. F. Biophys. J. **2011**, 101, 2061–2069.

[33] Chen, P. C.; Hub, J. S. Biophys. J. **2015**, 108, 2573–2584.

[34] Akaike, H. In International Encyclopedia of Statistical Science; Lovric, M., Ed.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2011; pp 25–25.

[35] Schwarz, G. The Annals of Statistics **1978**, 6, 461–464.

[36] Burnham, K. P.; Anderson, D. R. Model Selection and Multimodel Inference, 2nd ed.; Springer, 2002; p 76–123.

[37] Kass, R. E.; Raftery, A. E. Journal of the American Statistical Association **1995**, 90, 773–795.

[38] Gabriel, E.; Fagg, G. E.; Bosilca, G.; Angskun, T.; Dongarra, J. J.; Squyres, J. M.; Sahay, V.; Kambadur, P.; Barrett, B.; Lumsdaine, A.; Castain, R. H.; Daniel, D. J.; Graham, R. L.; Woodall, T. S. Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation. Proceedings, 11th European PVM/MPI Users' Group Meeting. Budapest, Hungary, 2004; pp 97–104.

[39] Franke, D.; Petoukhov, M. V.; Konarev, P. V.; Panjkovich, A.; Tuukkanen, A.; Mertens, H. D. T.; Kikhney, A. G.; Hajizadeh, N. R.; Franklin, J. M.; Jeffries, C. M.; Svergun, D. I. J Appl Crystallogr **2017**, 50, 1212–1225.

[40] Konarev, P. V.; Svergun, D. I. IUCrJ **2015**, 2, 352–360.

Supporting Information for BEES: Bayesian Ensemble Estimation from SAS, a SASSIE-web Module

Samuel Bowerman[1], Joseph E. Curtis[2], Joseph Clayton[1], Emre H. Brooks[3], and Jeff Wereszczynski[1]

[1]Department of Physics and the Center for the Molecular Study of Condensed Soft Matter, Illinois Institute of Technology, Chicago, IL 60616
[2]NIST Center for Neutron Research, National Institute of Standards and Technology, Gaithersburg, MD 20899
[3]University of Texas Health Science Center, San Antonio, TX 78229

S1 Information Criteria

The rigorous comparison of theoretical models to experimental data requires creating models that are rich enough to describe the underlying physical structures that generated the data while simultaneously avoiding overfitting. Biomolecules exist in an ensemble of conformations in solution, therefore an ensemble of theoretical structures is typically required to interpret SAS data. However, it is imperative that the final model does not achieve a strong goodness-of-fit value by including of an arbitrary number of parameters (here, the number of scattering profiles). As a result, the true "best model" must be a balance between optimizing the goodness-of-fit metric and minimizing the number of underlying scattering states. To this end, the BEES module utilizes "Information Criterion" (IC) in order to penalize model goodness-of-fit values according to their ensemble size. Users have the option to use one of two different IC metrics during fitting — the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC):[34,35]

$$AIC = 2k - 2 \cdot \log\left(\hat{L}\right) \tag{1}$$

$$BIC = \log(n) \cdot k - 2 \cdot \log\left(\hat{L}\right) \tag{2}$$

Here, k is the number of model parameters (number of scattering states), $\hat{L}$ is the maximum observed likelihood value during the Bayesian Monte Carlo parameter fitting, and n is the number of points in the experimental data set.

Both the BIC and AIC have forms that reward models with improved experimental fits (higher values of $\hat{L}$) and penalize those with more parameters (higher values of k). The BIC is closely related to the AIC; however, it is derived from Bayesian principles rather than the frequentist foundation of the AIC. In both metrics, smaller values are indicative of better model performance, with the defining separation between them being the strength of the penalty term. In the AIC, the penalty is always double the number of states, whereas the BIC penalty will become increasingly larger for a larger number of data points. In reality, both metrics are an approximate way to identify the true model, and the AIC may be more prone to false positive estimations (including too many states), while the BIC metric may be more prone to false negatives (rejecting too many states), depending on the number of experimental data points. However, it is often possible that both metrics converge upon the same solution, as is the case with the K63 example presented in the main text.

The model with the minimum IC value can be interpreted as the most likely, best performing, model. While it may be tempting to accept this model and reject all others, Bayesian principles dictate that there is a possibility that one of these other models might actually be more accurate to the true nature of the system, even though each one possesses a weaker IC value. The probability that a model is, in fact, a better assessment of the data can be calculated by comparing the model IC values to the lowest IC value:[36]

$$RP(m) = \mathrm{e}^{(\mathrm{IC_m} - \mathrm{IC_o})/2} \tag{3}$$

where $RP(m)$ and IC_m are the "relative performance" and IC values of model m, respectively, and IC_o is the minimum IC value across all observed models.

Because the BIC and AIC apply different penalties to the number of states, they may also produce different relative performance values for the same set of models. Depending on the number of data points, the BIC will produce relative performance values for competing models that are either closer to ($n \leq 7$) or further from ($n \geq 8$) the performance of the model with the lowest BIC. That is, if the number of independent data points is seven or fewer, then more models will have a relative performance closer to 1.0 than if evaluated by AIC. On the other hand, if the number of observed data points is greater than eight, then more models will have relative performances closer to 0.0 if they are evaluated by the BIC in place of the AIC. In the end, the choice of BIC vs AIC evaluation is up to the user, and it may sometimes be appropriate to use both to determine upper and lower bounds for relative model performances.

S2 Parallel Performance

The BEES algorithm suffers from two potential bottlenecks: the number of profiles in the user-supplied candidate pool and the ensemble size at which overfitting is observed. While knowledge of the latter cannot be determined *a priori*, benchmark of the BEES parallel performance can help guide users in determining how much data reduction is necessary before running the BEES module. These benchmarks were performed under minimal server load, and therefore represent a best case scenario (i.e., a lower limit to the computational wall clock) for the BEES algorithm. Detailed further below, we find that users can calculate the full combination of models from 14 in ∼36 hours using 6 processors, and candidate pools of 22 members can be fit to three-member states (overfitting observed in four-member models) in the same timescale using 6 processors.

To perform these benchmarks, we constructed candidate states of varying sizes (n = 2, 6, 8, 10, 12, 20, 25) in addition to our original 14-member candidate pool. Each BEES routine was run using five Monte Carlo replicas, where each replica was conducted for 10,000 iterations. It can be seen that the total number of combinations scales exponentially with the number of candidate profiles (Figure S1), as does the required wall clock. Furthermore, the time required to build the models scales almost linearly with the number of processors: for n = 10, the required time was 6.4 hours on two processors, in comparison to the 2.2 hours required for six processors. From these data, we observe that users can calculate full combinations of states from candidate pools of 14 members in a reasonable (∼36 hours) time frame when run on 6 processors.

In contrast to the previous discussion, many users may not desire to calculate the full combination of states, but rather find the single best model that avoids overfitting. In this case, the required computational cost may be drastically reduced, especially if overfitting is observed at a relatively small (≤ 3) ensemble size. For this reason, we have conducted a secondary benchmark of BEES routines containing 10, 14, 20, and 25 candidate profiles (Figure S2). Each BEES run utilized five Monte Carlo replicas of 10,000 iterations each on six processors. Using these benchmarks, we see that a three-member solution (overfitting observed at four states) from a candidate pool of 25 members can be found in ∼30 hours under ideal server load.

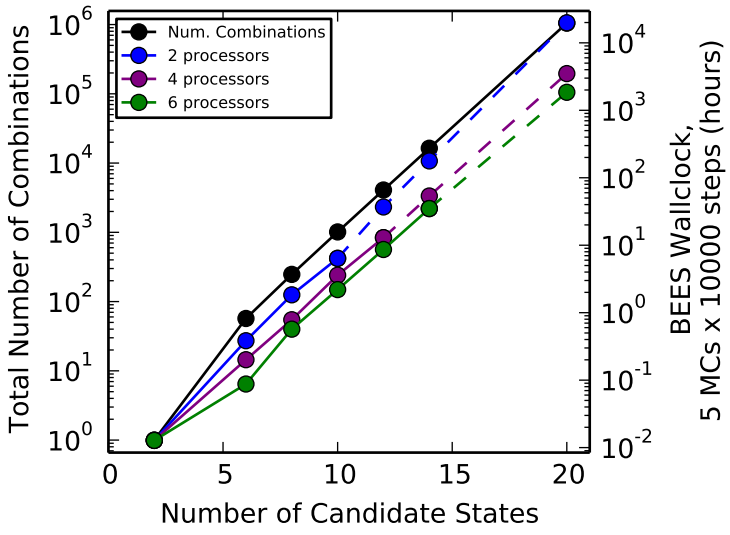


Figure S1: Performance of the BEES module with respect to candidate pool size when evaluating every possible combination of states. Benchmarks were determined for BEES routine of five Monte Carlo replicas that were run for 10,000 iterations each on two (blue), four (purple), and six (green) processors. Solid lines connect benchmarks that were explicitly measured, whereas dotted lines connect to points that are extrapolated from an exponential fitting of the measured benchmark data. From these data, even on six processors, the BEES module would require over 1,000 hours (∼42 days) to examine every possible combination of states from 20 candidate members.

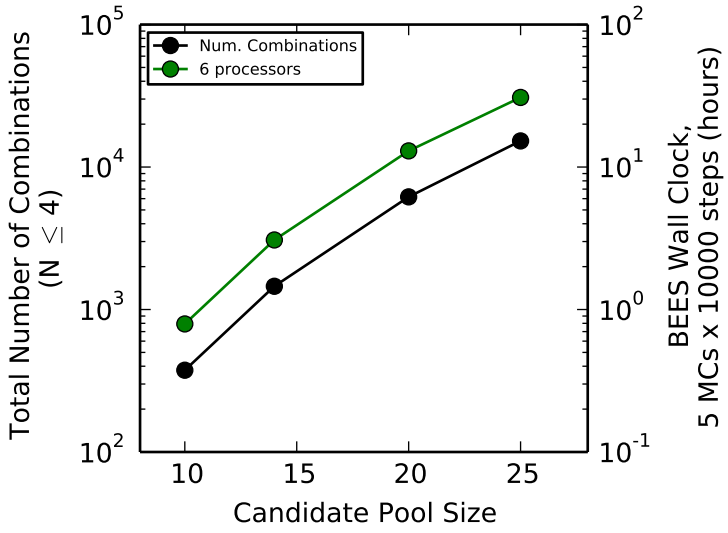


Figure S2: Performance of BEES module with respect to candidate pool size when the experimental data is best described by a three-state solution (overfitting observed at four-member solutions). Benchmarks were determined for a BEES routine that employed five Monte Carlo replicas that were run for 10,000 iterations on six processors. From the interpolation of this data, a three-state solution could be identified as the best model from a candidate pool of 25 members in ∼30 hours when run on six processors.