Supplementary Data Legends

Data 1 - Excel spreadsheet containing a subset of randomly selected 41,200 predicted structures with no counterparts in the RCSB PDB, with their calculated properties.

Video 1 - Animation showing the 100 DMD-generated structures for the O88338 Cadherin-16 from *M. Musculus*, with their calculated R_s and $[\eta]$ and the $p(r)$ vs. r distributions reported in each video frame.

Data 2 - Excel spreadsheet containing the calculated parameters for the 100 DMD-generated structures of the O88338 Cadherin-16 from *M. Musculus*.

Supplementary Methods

US-SOMO-AF database and data organization background

We chose a NoSQL MongoDB (https://www.Mongodb.com) database to store the hydrodynamic calculations and metadata due to its familiarity to the authors and its native support by the chosen website framework. Structural and CD properties were kept as separate files to simplify website direct download access. All steps processing large numbers of structures were performed by creating and running command line scripts. All work was done on the PDB files and not the mmCIF files, as we knew the hydrodynamic, structural and CD calculation programs we utilized supported the PDB format.

Collecting and pre-processing the AF database entries

To collect the AlphaFold entries, we began by downloading the complete 52 GB set of AlphaFold tar files and the sequences.fasta file (http://ftp.ebi.ac.uk/pub/databases/alphafold). We noted that the 365,198 AlphaFold predicted structures contained 362,312 unique UniProt accession codes. Multiple frames (designated -F2, -F3 etc) were present for 208 of the accession codes. A Perl (https://www.perl.org) script was written to: *a*-get the UniProt accession code from each AlphaFold first frame (-F1) PDB file name; *b*-extract the signal peptide information, utilizing Curl (https://man7.org/linux/man-pages/man1/curl.1.html) to pull the UniProt html content; *c*-extract the FASTA sequence from either the sequences.fasta file, or for those accession codes not present in the sequences.fasta file, perform an additional Curl pull from UniProt for the FASTA sequence; *d*-scan the downloaded UniProt html content for the Signal Peptide string and capture its value; *e*-and insert the Signal Peptide value and FASTA sequence keyed on the UniProt accession code into the database. This step took approximately five days to complete, as we ran the Curl calls serially to avoid overtaxing the UniProt servers.

Preparing the structures for calculations

To prepare the structures for hydrodynamic, structural and CD calculations, we produced additional scripts. As the signal peptide would not be present in most purified actual samples, we needed to remove it from the first frame PDB files. To remove it, a Perl script was written which, for each entry in the database that had a signal peptide: *a*-found the associated frame 1 PDB file; *b*-validated the residues present in the PDB file against the database retrieved FASTA sequence, and if so, removed the signal peptide residues from the PDB and annotated the PDB with a REMARK recording indicating that the signal peptide was removed. This step took a few hours to run. Note: there were only 45,064 of the 362,312 entries with a signal peptide.

Our next preparation step was to add disulphide bond information as SSBOND records to the PDB. For this step we utilized the disulphide bond identification feature recently developed in US-SOMO[23], and modified the US-SOMO code to expose this feature via the command line. To this end, an additional script was written which called the modified US-SOMO code to compute the SSBOND records and these were inserted by the script into the PDB files. This was run on the complete set of AlphaFold entries (all frames). To add α-helix and β-sheet information into the PDB files, we create a UCSF-Chimera[26] command script to simply open and write back the PDB file. Chimera utilizes a DSSP-based algorithm[25] to identify stretches of secondary structure. Running Chimera with the --nogui option on this script added the HELIX, SHEET and CONECT records to the PDB file. We added REMARK records to the PDB files with another script to detail our additions. Finally, we produced mmCIFs for the website using RCSB's MAXIT software (https://sw-tools.rcsb.org/apps/MAXIT). This required two processing runs as MAXIT needed to first convert the PDB file to a CIF file and subsequently convert the CIF file to an mmCIF file.

Hydrodynamic and structural parameters computations

To compute the hydrodynamic parameters and structural properties, we used the US-SOMO software, that includes several modeling options and multiple hydrodynamic parameter calculation algorithms[21-23]. For this work, we chose the "SoMo with overlaps" bead modeling method, that coupled with the ZENO computational method[27-29] has produced the best matching between experimental and computed parameters for an extended set of test proteins[19,22]. The SoMo with overlaps method is based on the original SOMO method in which each main- and side-chain segments for every amino acid are represented by a bead whose volume includes that of the theoretically bound hydration waters[40]. Although US-SOMO now includes the state-of-the-art GRPY method[23,41] to compute the hydrodynamic parameters of bead models with overlaps, it is significantly more computationally intensive, and its main advantage would be to produce also the rotational diffusion parameters (such as the rotational correlation time(s)), which are, however, more difficult to accurately measure. All the computations were carried out under standard solvent conditions (water at 20 °C, pH 7).

The US-SOMO software includes a batch mode, which can compute parameters and a SAXS $p(r)$ vs. r distribution on a list of structures. Initial performance testing revealed that each structure’s calculations would require about one minute and would therefore take approximately 250 days to complete the entire set. Therefore, to run the calculations in a reasonable time, we chose to run on 50 cores. To simplify execution and ensure against graphical user interface interaction mistakes, the development US-SOMO code was enhanced to support the execution of script files. We wrote a Perl script to produce US-SOMO scripts to run the hydrodynamic and structural calculations. Each produced script was limited to processing 250 structures, resulting in 1461 US-SOMO scripts. These were run in parallel on 50 cores, each producing an output CSV file detailing the results for up to 250 structures and associated SAXS $p(r)$ vs. r data files.

Circular Dichroism spectra

To compute the CD spectra from our prepared PDB files, we used SESCA[18]. For this step we wrote a Perl script to run SESCA_main.py on our PDB files. SESCA produces a CD_comp.out file containing the CD spectrum. As we wished to run in parallel and the SESCA CD_comp.out file name is not unique, we create a unique temporary directory within the script to run SESCA and subsequently rename the CD_comp.out output file.

Database implementation

To populate the database with the hydrodynamic and structural parameters, we wrote a Perl script to extract them from the US-SOMO produced CSV files and insert them into a new table keyed on the UniProt accession code concatenated with the AlphaFold frame number.

To add metadata to the records of the new table, we wrote another Perl script to: *a*-read the processed PDB and extract the TITLE and HEADER; *b*-compute the percentage of residues in each structure identified as α-helix and β-sheet; *c*-compute the mean per residue confidence from the values, as reported by AlphaFold, in the PDB ATOM records' temperature factor field. This script produces a MongoDB command script to set these values on the existing table records, which was subsequently processed by the MongoDB command line interface program.

An additional Perl script was written to extract the hydrodynamic parameters and metadata from the database and produce a CSV format file for website user download. This script was run on the entire database.

The products at this stage included a MongoDB database populated with the hydrodynamic parameters and metadata and a collection of 365,198 prepared PDB, mmCIF, CSV, $p(r)$ vs. r and CD data files. To provide convenient complete user downloads, we created a Perl script to produce zip and compressed tar files for each entry.

Website implementation

The website was created with the GenApp framework[42]. The website runs in a Docker (https://www.docker.com) container based on Ubuntu 20.04.3 LTS (https://ubuntu.com) with PHP 7.4.3 (https://www.php.net), MongoDB 4.2.17 and Apache 2.4.41 (https://httpd.apache.org). GenApp application development works by creating various definition files and provides a rich user interface including support for atomic structure display (JSmol, https://sourceforge.net/projects/jsmol) and advanced plotting (Plotly, https://plotly.com). The

GenApp Docker container was built with the GenApp provided Dockerfile. A JSON (https://www.json.org) formatted module definition file was written detailing all the inputs, outputs, user interface layout and a reference to the underlying executable. GenApp provides limited constraints on the language of the module's executable. PHP was chosen for the module's executable due to its fast startup, good support for JSON and available MongoDB interface. A PHP executable file was created which: *a*-consumes an input object as described in the module definition file; *b*-does the appropriate lookups from the MongoDB, including, if multiple records are found matching the search string, an interactive user refinement to a single result; *c*-populates the output object, as described in the module definition file, with the hydrodynamic parameters, metadata and links to the PDB, mmCIF, CSV, $p(r)$ vs. r, CD, zip and compressed tar files for user download and a reference to the PDB for the JSmol viewer; *d*-and finally outputs the JSON output object. JSON formatted directives and menu files were modified from provided templates to, respectively, specify the overall website details and include a reference to the created module. The GenApp framework engine was run to build the complete website. Refining the website was an iterative procedure consisting of modifying the definition files and/or the module's executable, running the GenApp engine, and testing in a web browser.

Computational resources

All production steps except the website creation and hosting were performed on a shared 128 core dual EPYC 7742 system with 512 GB of RAM, 73 TB of storage and a 1 Gb internet connection running CentOS 8.4 located at the University of Lethbridge, Canada. Website creation was and hosting is done on the NSF supported Jetstream cloud[43] made possible by an XSEDE[44] allocation to E.B.

DMD simulations of AF-O88338

To expand the conformational space of AF-O88338, we used US-SOMO's interface[45,46] to a Discrete Molecular Dynamics (DMD) program[31.32]. Operations were carried out with the Linux version of US-SOMO operating on a cluster[43,44]. Relaxation was run for 5 ps at 0.7 kcal/mol/k_B, the production was run for 5 ns at 0.6 kcal/mol/k_B. The Andersen thermostat was used for both the relaxation and run stages.

Graphs, figures, and movie preparation

Non-website graphs were prepared with Origin v. 6.0 (Microcal, now OriginLab, https://www.originlab.com). Figures were assembled using PaintShopPro v. 5.3 (JASC Software, now Corel, https://www.paintshoppro.com). The movie's protein structures images were generated by UCSF Chimera. The movie's *p*(*r*) vs. *r* distribution plots were generated by Gnuplot (http://www.gnuplot.info). The movie's plots and text were inserted using ImageMagick (https://imagemagick.org). The final movie was assembled with FFmpeg (https://www.ffmpeg.org).

Supplementary references

40. Rai, N. et al. SOMO (SOlution MOdeler) differences between X-Ray- and NMR-derived bead models suggest a role for side chain flexibility in protein hydrodynamics. *Structure* **13,** 723-734 (2005). Doi: https://doi.org/10.1016/j.str.2005.02.012
41. Zuk, P.J., Cichocki, B. & Szymczak, P. GRPY: an accurate bead method for calculation of hydrodynamic properties of rigid biomacromolecules. *Biophys. J.* **115,** 782–800 (2018). Doi: https://doi.org/10.1016/j.bpj.2018.07.015
42. Savelyev, A. & Brookes, E. GenApp: Extensible tool for rapid generation of web and native GUI applications. *Future Gener. Comput. Syst.* **94,** 929-936 (2019). Doi: https://doi.org/10.1016/j.future.2017.09.069
43. Stewart, C.A. et al. Jetstream: a self-provisioned, scalable science and engineering cloud environment. In: *XSEDE '15: Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*. Association for Computing Machinery, New York, NY, USA, pp. 1-8 (2015). Doi: https://doi.org/10.1145/2792745.2792774
44. Towns, J. et al. XSEDE: Accelerating Scientific Discovery. *Comput. Sci. Eng.* **16,** 62-74 (2014). Doi: https://doi.org/10.1109/MCSE.2014.80

45. Brookes, E. et al. US-SOMO cluster methods: year one perspective. In: *XSEDE '13 Proceedings of the conference on extreme science and engineering discovery environment: gateway to discovery*, Article 65, pp 1-2 (2013). Doi: https://doi.org/10.1145/2484762.2484815

46. Rocco, M. & Brookes, E. Dynamical aspects of biomacromolecular multi-resolution modelling using the UltraScan Solution Modeler (US–SOMO) suite. In: *The Future of Dynamic Structural Science. NATO Science for Peace and Security Series A: Chemistry and Biology*. Howard, J., Sparkes, H., Raithby, P. & Churakov, A. (eds). Springer, Dordrecht, pp 189–199 (2014). Doi: https://doi.org/10.1007/978-94-017-8550-1_13