

# Tios: The Internet of Simulations. Turning Molecular Dynamics into a Data Streaming Web Application.

Athina Meletiou,<sup>§</sup> James Gebbie-Rayet<sup>†</sup>, Charles Laughton<sup>\*,§</sup>

<sup>§</sup>School of Pharmacy and Centre for Biomolecular Sciences, University of Nottingham,  
University Park, Nottingham NG7 2RD, United Kingdom

<sup>†</sup>Scientific Computing Department, STFC Daresbury Laboratory, Warrington WA4 4AD, United  
Kingdom

## ABSTRACT

The configuration of most current academic high-performance computing (HPC) resources tends to enforce ways of working with, and thinking about, molecular dynamics (MD) simulations that are not always optimal. For example, when the aim of the simulation(s) is to produce a representative sample of a Boltzmann weighted ensemble, the ideal scenario would be to be able to do just that – i.e. to tap into a running simulation of indefinite length, collect data from it in real time, and only terminate the simulation once the quality of a sample was assured. Current approaches, based on batch jobs of proscribed maximum length, and a post-processing style of data analysis, inhibit this. In the spirit of the Internet of Things, we have developed Tios, a Python application that turns MD simulations into remotely discoverable and accessible streaming web applications to which researchers can connect and download data as they please. Tios is freely available, works with standard MD codes, and requires no modifications to them. In this paper we outline how Tios works and present a number of test cases that demonstrate its capabilities.

## INTRODUCTION

MD simulations are some of the heaviest consumers of academic HPC resources; a recent analysis<sup>1</sup> of workloads across the US XSEDE facilities revealed that two of the top five applications in terms of Service Units (SUs) consumed are NAMD<sup>2</sup> and GROMACS.<sup>3-4</sup> Despite this, one can argue that MD simulations are not always well suited to the constraints placed on them by such services. The almost ubiquitous batch job model makes sense for computing tasks that proceed towards a well-defined result, however most frequently<sup>5-9</sup> the purpose of MD is to generate an ensemble of molecular configurations that represents an adequate sample for comparison with "reality". One can therefore easily identify at least three problems when running MD simulations within conventional HPC environments and all three refer to the total required amount of simulation, namely that: a) it cannot be known a priori, b) it is not easily evaluable as the simulation runs, so in most current practice is tested for a posteriori, and c) it often exceeds typical HPC batch job lengths (24-48 hours), thus creating the need for checkpointing and

restarting mechanisms which often leads to data being fragmented across multiple output files (though GROMACS has options to allow restarted simulations to append data to existing files) .

In recent years, the Internet of Things<sup>10-11</sup> has been gaining increasing attention from researchers because of its focus on highly distributed networks of discoverable devices communicating in a resilient way with humans as well as other devices. Tios is designed to provide MD simulation-based research with an alternative way of working inspired by this: in the spirit of the Internet of Things, MD simulations become transmitters that stream live trajectory data to the internet, while independent receivers connect to a data stream and download information (trajectory data, and meta-data) as and when they wish.

The Tios model is that data production and collection are decoupled. An MD simulation broadcasts its trajectory data to the internet whether or not one or more receivers are “listening” to it. The resource where the data is collected does not need to be the same as the one on which it is being generated. The reverse is also true: if an MD simulation is stopped while a receiver is collecting data, the receiver only stalls. When the simulation is restarted, provided the receiver is still listening, data collection seamlessly resumes from where it left off. Importantly, the Tios model also permits the MD job to be restarted on a different resource from the one it first ran on (as long as that resource has all the capabilities the job requires, e.g. number of cores/nodes if this is set explicitly for the simulation). It is also possible for multiple receivers to collect from the same data stream concurrently, and different receivers may collect different subsets of the data.

A key feature of the Tios philosophy is that it takes no responsibility itself for trajectory data storage, only for data streaming. It is up to the simulation owner, or other users who connect to the simulation, to create trajectory files from this data. So, if a user comes across a simulation started by someone else that has already run for 20 microseconds, there is no way within Tios itself to access past data (otherwise Tios would need to be backed by an inexhaustible data store). One might assume though that the simulation owner, who created it in the first place, has been monitoring the simulation up to now, and has data files they could be willing to share.

## **TECHNICAL DESCRIPTION**

Currently, a typical MD simulation of a biomolecular system might involve collecting the coordinates of 200,000 atoms every 10 picoseconds of simulated time. On a modern HPC resource, simulation speeds of 50 nanoseconds a day are fairly standard.<sup>12</sup> Assuming that the Cartesian coordinates are encoded as single precision floats, this equates to a data production rate of about 140 kB/s, or 1.1 Mbit/s. In practice, significant (up to 50%) data compression is achievable. For comparison, a 720p YouTube video stream is 2.5 Mbit/s. Therefore, it becomes clear that the idea of live streaming MD simulation data is realistic.

The Tios model has three basic components: data transmitters, data receivers, and data hubs. Tios transmitters are attached to conventional MD simulation jobs and stream their data to a Tios data hub. Tios receivers discover the data streams that are available, and connect to them, through querying a Tios data hub. The lifecycle of a Tios simulation is as follows.

a) Set-up

The necessary coordinate, topology, run control, etc. files are uploaded to a Tios hub, where they are given a unique ID. This can be done from any resource – e.g. the user's own laptop/workstation.

b) Job launch

On the HPC resource of choice, a Tios transmitter job is started (e.g. as a batch job). This downloads the required files from the Tios hub, and starts the job.

c) Data transmission

As the MD simulation proceeds, each new frame of trajectory data (coordinates and energy meta-data) is transmitted back to the Tios hub. The hub only stores the most recent snapshot of information.

d) Simulation query

The Tios hub may be queried from any remote resource to find out details of the jobs it holds. These may be currently running or not. A variety of meta-data may be available.

e) Data collection

A Tios receiver can be started on any resource to connect to a remotely running job. The connection is actually to the entry in the Tios hub, rather than direct to the transmitter itself. The receiver polls the hub, downloading each fresh snapshot of data as it is uploaded by the transmitter. Data collection can be terminated and restarted as required. Obviously, any data that is transmitted while a receiver is not active is lost.

f) Job termination

The MD job will typically have been configured originally to run for as long as possible – some codes (e.g. GROMACS<sup>3-4</sup>) may allow an indefinite run time. However, in reality the job may terminate due to exceeding the run time in the batch job queue, or be stopped by the user. When this happens, any final checkpoint files are uploaded to the hub before the job terminates. Jobs can subsequently be restarted from where they left off by issuing the job launch command again in a new batch job script (step b above).

Tios is written in Python and it provides the command line tools required to create, launch, control, discover and query MD simulations. These commands call functions provided by the Tios library which has a public API that may be incorporated into user's own code. More details of the command lines tools and Python library are provided in the Supplementary Information. Most importantly, Tios does not require any modifications whatsoever to the MD codes (GROMACS<sup>3-4</sup> and NAMD<sup>2</sup>) it supports, making it very user-friendly. Further, Tios has only minimal impact on

their performance. Tios leverages the socket-based Interactive Molecular Dynamics (IMD) protocol<sup>13</sup> implemented in the aforementioned MD codes. The IMD protocol<sup>13</sup> was developed originally to permit real-time viewing of MD simulations in a separate graphics program, as well as potentially interacting with the simulation (steering). It enables (i) monitoring of an ongoing MD simulation, (ii) identifying when a new set of coordinates has been generated, and (iii) publishing them as well as their associated metadata. The advantage of using IMD is that it provides unbuffered access to the current state of the system. Without IMD, one would have to interrogate the output files the simulation produces, which a) are inconsistent in format from simulation package to simulation package, while the IMD protocol provides a standard; b) are frequently heavily buffered, so are unsynchronised with the actual progression of the simulation, and c) could become very large in size for the sort of multi-microsecond simulation that Tios enables – if IMD is being used it is not necessary to ask for any trajectory file to be written at all.

The Tios library presents much of the MD trajectory-specific data to the user as standard NumPy<sup>14</sup> arrays. We have chosen MongoDB for the hub database system as it is free and open-source, and its NoSQL nature makes it adaptable to the various data structures required for different MD codes. Simultaneously, we have designed Tios in a way that alternative database back-ends could be implemented fairly straightforwardly.

## **ACCESS CONTROL AND SECURITY**

Before using Tios, users must register with a Tios hub. As Tios focuses on promoting collaborative science, user permissions are set in such a way that any user registered on a particular hub may collect data from any simulation running on the same hub; not only the ones that they personally have launched. While Tios simulations are “readable” by any user registered to that hub, only the owner has “write” permissions, e.g. simulations can only be (re)started and stopped by their owner. Data transfers to and from the hub make use of the encryption and security facilities built in to Mongo, so that even if the security of a hub was compromised, this would provide no access to the HPC service.

## **CASE STUDIES**

### *System size capability*

We have been able to stream a simulation of beta1-adrenoceptor<sup>16</sup> (PDB code: 4BVN) in a lipid bilayer and box of TIP3P water (183,589 atoms in total) from ARCHER, the UK national HPC service, to a local (University of Nottingham) workstation without errors or frame drops, and the same simulation was later restarted and streamed back to our workstations from Stampede2 at the University of Texas Advanced Computing Center (TACC). The second collection ran for twelve hours during which 404 frames of data were gathered.

### *Impact of Tios on MD code performance*

A GROMACS<sup>3-4</sup> portable run input (.tpr) file for a simulation of FOXO3<sup>15</sup> (PDB code: 2uzk) in a cubic box of SPC/E water (57,037 atoms in total) was prepared and submitted to Tios. The MD job was run for 1 ns, broadcasting coordinates every 10 ps, on 1 Intel® Xeon® CPU E5-1620 v4 @ 3.50GHz node with a total of 4 cores, 8 logical cores, and 1 NVIDIA Quadro P5000 GPU, using GROMACS version 2018. On average, over six test runs performed, the jobs completed in 779 seconds (111 ns/day, with a broadcast rate of 7.5 frames/minute). The same tests were then run natively (outside Tios), and with the IMD feature turned off, on the same resource. These completed on average in 699 seconds (124 ns/day). Thus, interfacing GROMACS with Tios gave only a 10% reduction in code performance. Additionally, a short MD simulation of ubiquitin (PDB code: 1ubq) in a TIP3P water box (7,052 atoms in total) was performed using NAMD.<sup>2</sup> The MD job was run for 1 ns, broadcasting coordinates every 10 ps, on the same resource. On average, over three test runs performed, the jobs completed with a performance of 2.24 ns/day. The same MD job was then rerun natively on the same resource and it completed with an average performance of 2.25 ns/day. Presumably, the reason the impact on GROMACS performance is more noticeable is because its baseline (GPU-accelerated) speed with the chosen test-case is greater.

### *Simultaneous data collection*

In another test, ten Tios receivers were established on Amazon EC2 t2.small instances in different regions around the world (Figure 1). Simultaneous *tios-collect* commands were issued on each of the ten instances and data were collected from the same FOXO3<sup>15</sup> simulation discussed above running at the University of Nottingham.

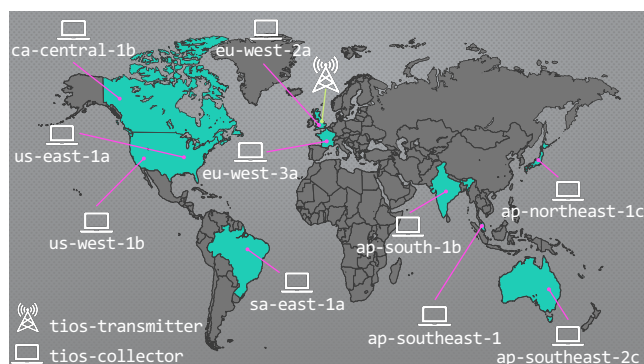


Figure 1. Tios data collection test. Ten tios-collectors established on Amazon EC2 t2.small instances around the world concurrently collected data from a simulation of FOXO3 running on a workstation at the University of Nottingham without any data loss (see text for details).

The receivers downloaded trajectory data every 10 ps, meaning that without any loss of frames the generated portable trajectory (.xtc) file would be populated with 100 frames over the course of the 1 ns simulation. No frame loss was observed at any one of the ten receivers.

### *Impact of simulation data generation rate on Tios performance*

In another test, the same FOXO3 simulation was launched on the Condor pool at the University of Nottingham using a variety of interval rates (-r flag). Subsequently, collection commands were issued from a different terminal window on the same resource. This test aimed to examine the performance of Tios at high simulation frame production rates. We hypothesised that there could be two “pinch-points”: the speed at which frames produced by the simulation could be uploaded to the Tios hub (the upload speed), and the speed with which frames could be downloaded from the Tios hub by receivers (the download speed). If the upload speed is less than the simulation production frame rate, we will see the effective speed of the simulations (the broadcast frame rate) throttled. If the download speed was less than the broadcast frame rate, we would see dropped frames. Running transmitter, hub, and receiver on the same local network ensured performance metrics related to the characteristics of the code, rather than network limitations. We observed that the Tios transmitter was able to “keep up” with the simulation if it produced up to 20 frames per minute; above this some throttling of the simulations was evident (Figure S1a, supplementary information) Similarly, Tios receivers could “keep up” with simulations broadcasting at up to 20 frames per minute before frame drops were seen (Figure S1b).

### *Science demonstrator*

As part of a project looking at the long timescale dynamics of DNA, a simulation of the “Drew-Dickerson” DNA dodecamer (dCGCGAATTCGCG)<sup>17</sup> was prepared (AMBER parmBSC1 forcefield and TIP3P water box, sodium ions for electroneutrality), and run using Tios with GROMACS version 2018.5. The Tios sampling interval was set at 10 ps and on a GPU-enabled resource this produced a broadcast frame rate of about 24 frames/minute.

Intermittent observation of the simulation (without regular data collection) revealed that, as commonly reported, the simulation showed intermittent reversible “fraying” of the terminal base pairs, during which their normal Watson-Crick hydrogen bonding was lost. At 2.2 microseconds we noted a non-canonical conformation at the 5'-end of the helix featuring C-N3:H-N2-G and C-N4-H:N3-G H-bonds, i.e. between the Watson-Crick edge of the C and the sugar edge of the G. Non-canonical conformations are more common in RNA than DNA but this particular geometry is not well documented (it does not appear in the Tinoco<sup>18</sup> and Westhoff<sup>17,19</sup> compilations of RNA H-bonding motifs, but does appear as “GC4” in the Jeffrey/Saenger<sup>20</sup> compilation).

We observed that at the other end of the helix, the 3'-terminal G:C base pair retained its canonical interactions. We were interested to see if the formation of this alternate conformation at the 5'-end had an effect on the structural stability of the rest of the helix. A Python script (see supplementary data) was written that, using the Tios library, began to monitor the variation in the length of each “central” H-bond in the helix over time (i.e., between N1 of A/G and N3 of

C/G). We monitored convergence in the distance distributions for each H-bond by discretising their values into ten bins, and then calculating the Kullback-Leibler divergence between the distribution of the first half of the accumulating set of distance data, and the second. The collection was terminated automatically when all distance distributions had satisfied the convergence threshold (set at 0.1). In total, 3,588 snapshots were collected, representing 35.88 nanoseconds of simulation time.

During this period, the 5' base pair retained its non-canonical interactions (Figure 2a, top). Comparing the distributions in H-bond distances for the first and second halves of the data collection phase (Figure 2b) confirms that the Kullback-Leibler divergence criterion has resulted in a well-converged sampling. The mean N1:N3 separations for each base pair (Figure 2c) show that all pairs other than the first retain standard Watson-Crick geometry. From the root mean square fluctuations (RMSF) in the N1:N3 distances (Figure 2c) we see that the 5'-non-canonical base pair shows increased dynamics compared to the other positions, but that this has no effect on the stability of the base pair at position 2. The dynamics of this base pair is the same as its palindromic counterpart at position 11.

The RMSF also reveal a c. 30% greater flexibility of the central A:T base pairs compared to the canonical G:C base pairs, as is expected due to the former being stabilised by just two H-bonds rather than three. Unexpectedly, we observed a high RMSF for the 3'-terminal base pair; plotting the time series of the H-bond separation shows this is due to occasional large but short-lived "fraying" dynamics at this end of the helix (Figure 2a, bottom, and Figure 2d). In summary, we can conclude that the non-canonical base paired conformation at the 5'-terminus of the sequence is stable over multi-nanosecond timescales but is intrinsically more dynamic than a normally-H-bonded base pair. This increased dynamics is however very localised, with no detectable effect on the stability of the neighbouring base pair.

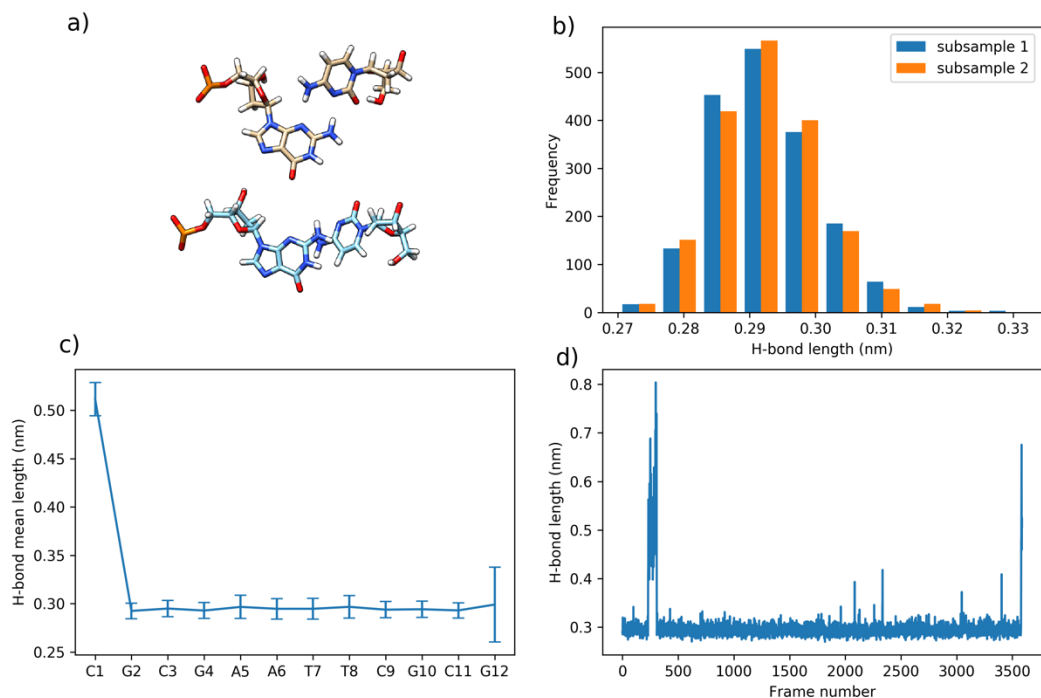


Figure 2. Application of Tios to the study of DNA dynamics. Panel a): the non-canonically H-bonded structure of the 5'-terminal GC base pair observed throughout the data collection phase (top), and the transient "frayed" structure observed for the 3'-terminal GC base pair (bottom). Panel b): evaluation of sampling convergence. Distributions in the H-bond distance data for base pair 2 generated from the first and second subsamples of the data have a Kullback-Leibler divergence of less than 0.1 when discretised into 10 bins. Panel c): mean H-bond lengths for each base pair. Error bars show +/- one standard deviation. The large error bars for base pair 12 are the result of intermittent short-lived "fraying" dynamics (Panel d)).

## LIMITATIONS

As any streaming application, any MD data that gets transmitted while no receiver is active to collect them, will be lost. A small percentage of frames may still be lost depending on network performance, even when a receiver is active (e.g. if the rate at which frames are being uploaded to the hub by the transmitter exceeds the rate they can be downloaded from the hub by a receiver). This means that Tios is best suited to MD simulations that aim to generate a sample representative of an equilibrated ensemble, rather than a perfect time series. Additionally, Tios can currently only be used for non-periodic or constant volume (NVT) simulations, because the IMD protocol (see above) does not provide the ability to access periodic box data on a per-frame basis. Although not ideal, it is the case that for many extended "vanilla" MD simulations where the aim is purely to generate a good sample of a simulation at equilibrium (the key target for



Tios), once equilibrated in an NPT ensemble the production phase can be run in the NVT ensemble without any issues.

Another limitation of Tios is that it only streams coordinate and limited energy-related metadata: other simulation outputs (e.g. log files, velocity information, etc.) are not currently captured. Additionally, Tios does not currently support ensemble type simulations (e.g. replica exchange or other multiple walker-based methods). Tios requires that the resource the simulation is running on has internet access (e.g. in the case of this meant establishing a concurrent port forwarding process from the compute nodes). Finally, though installing the client code for transmitters and receivers is very straightforward, a little more effort is required to launch a persistent Tios hub, if an existing one is not available or suitable.

As mentioned above, Tios currently only works with NAMD and GROMACS but would be straightforward to adapt to other MD codes that support the IMD protocol. Currently this would include LAMMPS,<sup>21</sup> and HOOMD.<sup>22-23</sup>

## **CONCLUSIONS**

Tios allows researchers to rethink their ways of working with molecular dynamics simulations, without compromising on MD code performance. It promotes a more interactive approach to working with ‘live’ simulations, and to data sharing. The simulation can be regarded as an independent entity, not tied to any physical computing resource, which can leverage whatever resource is available at any given time, wherever it might be in the world, to move it forward. It has been demonstrated to work effectively on a range of resources from laptops to national HPC services. In future work we plan to extend Tios to support other MD codes (e.g. OpenMM<sup>24</sup>), and to other types of simulation, particularly constant pressure ones.

## **AUTHOR INFORMATION**

### **Corresponding Author**

\*Email: [charles.laughton@nottingham.ac.uk](mailto:charles.laughton@nottingham.ac.uk)

### **Author Contributions**

C.L. conceived the research and wrote most of the software. G.B contributed to the software for the Tios “hub” component. A.M. completed testing and C.L. did bug correction. A.M. and C.L. wrote a draft of the paper and the final version.

### **Funding**

This work was supported by EPSRC Grants EP/L000253/1 and EP/P011993/1, by the University of Nottingham, and by Amazon Web Services via their AWS Cloud Credits for Research program.

### **Notes**

The authors declare no competing financial interest.

### ARCHER Software availability

Tios is freely available under the BSD “three clause” license at <https://bitbucket.org/claughton/tios.git> (date of access: 8<sup>th</sup> July 2019) which includes the Tios source code and additional documentation on installation and usage.

### ACKNOWLEDGMENTS

This work used the ARCHER UK National Supercomputing Service (<http://www.archer.ac.uk>), the University of Nottingham HPC service, and the Extreme Science and Engineering Discovery Environment (XSEDE) computational facility Stampede2 which is supported by National Science Foundation grant number ACI-1548562, via allocation TG-MCB090174. We thank Shantenu Jha (Rutgers) for support and advice on accessing XSEDE resources, Adrian Jackson at the Edinburgh Parallel Computing Centre for help in implementing the software on ARCHER, and Keverne Louison (University of Nottingham) for the beta-adrenoceptor simulation files.

### ASSOCIATED CONTENT

Supporting Information Available: Details of the Tios command line tools and Python API; further details of Tios performance analysis (Figure S1), and a copy of the Jupyter Notebook used for the “science demonstrator” example.

### REFERENCES

1. Simakov, N. A.; White, J. P.; DeLeon, R. L.; Gallo, S. M.; Jones, M. D.; Palmer, J. T.; Plessinger, B. D.; Furlani, T. R. A Workload Analysis of NSF's Innovative HPC Resources Using XDMoD. *eprint arXiv:1801.04306* **2018**, arXiv:1801.04306.
2. Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. Scalable Molecular Dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781-1802.
3. van der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. GROMACS: Fast, Flexible, and Free. *J. Comput. Chem.* **2005**, *26*, 1701-1718.
4. Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High Performance Molecular Simulations through Multi-Level Parallelism from Laptops to Supercomputers. *SoftwareX* **2015**, *1-2*, 19-25.
5. Fischer, N. M.; Polêto, M. D.; Steuer, J.; van der Spoel, D. Influence of Na<sup>+</sup> and Mg<sup>2+</sup> Ions on RNA Structures Studied with Molecular Dynamics Simulations. *Nucleic Acids Res.* **2018**, *46*, 4872-4882.
6. Ilie, I. M.; Nayar, D.; den Otter, W. K.; van der Vegt, N. F. A.; Briels, W. J. Intrinsic Conformational Preferences and Interactions in  $\alpha$ -Synuclein Fibrils: Insights from Molecular Dynamics Simulations. *J. Chem. Theory Comput.* **2018**, *14*, 3298-3310.

7. Srivastava, A.; Hirota, T.; Irle, S.; Tama, F. Conformational Dynamics of Human Protein Kinase CK2 $\alpha$  and its Effect on Function and Inhibition. *Proteins: Struct., Funct., Bioinf.* **2018**, *86*, 344-353.
8. Meng, Y.; Gao, C.; Clawson, D. K.; Atwell, S.; Russell, M.; Vieth, M.; Roux, B. Predicting the Conformational Variability of Abl Tyrosine Kinase Using Molecular Dynamics Simulations and Markov State Models. *J. Chem. Theory Comput.* **2018**, *14*, 2721-2732.
9. Balos, V.; Marekha, B.; Malm, C.; Wagner, M.; Nagata, Y.; Bonn, M.; Hunger, J. Specific Ion Effects on an Oligopeptide: Bidentate Binding Matters for the Guanidinium Cation. *Angew. Chem., Int. Ed.* **2019**, *58*, 332-337.
10. Xia, F.; Yang, L. T.; Wang, L.; Vinel, A. Internet of Things. *Int. J. Commun. Syst.* **2012**, *25*, 1101-1102.
11. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A Survey. *Comput. Netw.* **2010**, *54*, 2787-2805.
12. Kutzner, C.; Páll, S.; Fechner, M.; Esztermann, A.; de Groot, B. L.; Grubmüller, H. Best Bang for your Buck: GPU Nodes for GROMACS Biomolecular Simulations. *J. Comput. Chem.* **2015**, *36*, 1990-2008.
13. Stone, J. E.; Gullingsrud, J.; Schulten, K. A System for Interactive Molecular Dynamics Simulation. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, ACM: New York (NY, USA), 2001; pp 191-194.
14. Oliphant, T. *A Guide to NumPy*, Trelgol Publishing: USA, 2006.
15. Tsai, K.-L.; Sun, Y.-J.; Huang, C.-Y.; Yang, J.-Y.; Hung, M.-C.; Hsiao, C.-D. Crystal Structure of the Human FOXO3a-DBD/DNA Complex Suggests the Effects of Post-Translational Modification. *Nucleic Acids Res.* **2007**, *35*, 6984-6994.
16. Miller-Gallacher, J. L.; Nehmé, R.; Warne, T.; Edwards, P. C.; Schertler, G. F. X.; Leslie, A. G. W.; Tate, C. G. The 2.1 Å Resolution Structure of Cyanopindolol-Bound  $\beta$ 1-Adrenoceptor Identifies an Intramembrane Na<sup>+</sup> Ion that Stabilises the Ligand-Free Receptor. *PLoS One* **2014**, *9*, e92727.
17. Westhof, E. Re-refinement of the B-dodecamer d(CGCGAATTCGCG) with a Comparative Analysis of the Solvent in it and in the Z-hexamer d(5BrCG5BrCG5BrCG). *J. Biomol. Struct. Dyn.* **1987**, *5*, 581-600.
18. Tinoco, I., Jr. Appendix 1: Structures of Base Pairs Involving at Least Two Hydrogen Bonds. In *The RNA World* Gesteland, R. F.; Atkins, J. F., Eds.; Cold Spring Harbor Laboratory Press: 1993; pp 603-607.
19. Dirheimer, G.; Keith, G.; Dumas, P.; Westhof, E. Primary, Secondary, and Tertiary Structures of tRNAs. In *tRNA: Structure, Biosynthesis, and Function*, Söll, D.; RajBhandary, U., Eds.; ASM Press: Washington, DC, 1995; pp 93-126.
20. Jeffrey, G. A.; Saenger, W. *Hydrogen bonding in biological structures*, Springer-Verlag: Berlin, 1991.
21. Plimpton, S. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *J. Comput. Phys.* **1995**, *117*, 1-19.
22. Anderson, J. A.; Lorenz, C. D.; Travesset, A. General Purpose Molecular Dynamics Simulations Fully Implemented on Graphics Processing Units. *J. Comput. Phys.* **2008**, *227*, 5342-5359.

23. Glaser, J.; Nguyen, T. D.; Anderson, J. A.; Lui, P.; Spiga, F.; Millan, J. A.; Morse, D. C.; Glotzer, S. C. Strong Scaling of General-Purpose Molecular Dynamics Simulations on GPUs. *Comput. Phys. Commun.* **2015**, *192*, 97-107.
24. Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. OpenMM 7: Rapid Development of High Performance Algorithms for Molecular Dynamics. *PLoS Comput. Biol.* **2017**, *13*, e1005659.

## For Table of Contents Use Only

Tios: The Internet of Simulations. Turning Molecular Dynamics into a Data Streaming Web Application.

Athina Meletiou, James Gebbie-Rayet, Charles Laughton \*

