Protocols for Multi-Scale Molecular Dynamics Simulations in Amber and Gromacs: a Case Study of Intrinsically Disordered Amyloid Beta

Pamela Smardz, Midhun Mohan Anila, Pawel Rogowski, Mai Suan Li, Bartosz Różycki, Pawel Krupa*

Institute of Physics Polish Academy of Sciences, Al. Lotnikow 32/46, 02-668, Warsaw, Poland

*corresponding author, email: pkrupa@ifpan.edu.pl

SUMMARY:

As intrinsically disordered proteins are often challenging to study experimentally, we present step-by-step protocols for conducting all-atom and coarse-grained molecular dynamics simulations using two popular packages, Amber and Gromacs. We illustrate the protocols with the monomeric form of 1-42 amyloid-β (Aβ42) as an example, which can yield insights into their structure, dynamics, and physicochemical properties.

ABSTRACT:

Intrinsically disordered proteins (IDPs) pose challenges to conventional experimental techniques due to their large-scale conformational fluctuations and transient occurrence of structural elements. This work presents computational methods for studying IDPs at various levels of resolution. The included simulation protocol provides a step-by-step guide on how to conduct molecular dynamics (MD) simulations and analyze the results using the Amber and Gromacs packages, employing both all-atom and coarse-grained approaches. This protocol can be easily adapted to study other biomacromolecules, including folded and disordered proteins and peptides.

Furthermore, we describe how to perform standard molecular modeling operations, such as amino-acid substitutions (mutagenesis) and insertion of missing residues, and how to incorporate post-translational modifications into the simulations, such as disulfide bonds, which are often crucial for proteins to attain their proper structure. In conventional MD studies, disulfide bonds are typically fixed at the preparation step and remain unchanged throughout the simulations, unable to break or reform. In contrast, we introduce a dynamic approach that mimics bond breaking and reforming by applying additional distance restraints to the sulfur atoms of selected cysteine residues, allowing disulfide bonds to break and reform during the simulation.

We demonstrate the effectiveness of these methodologies by examining a model IDP, the monomeric form of 1-42 amyloid-β (Aβ42), both with and without disulfide bonds, at different levels of resolution. This study not only contributes to our understanding of the role of disulfide bonds but also provides detailed simulation protocols that can serve as a foundation for future investigations.

INTRODUCTION:

Proteins are one of the most important biological macromolecules, playing a variety of roles in organisms, such as serving as building blocks, catalyzing biochemical reactions as enzymes, facilitating molecular transport as transporters, and governing essential cellular processes as regulatory molecules[1]. For a considerable time, it was believed that every protein possessed a single unique structure, which was considered crucial for its functional role and believed to be deducible solely from its sequence (Anfinsen Dogma[2]). However, with advancements in science, it was discovered that many proteins and peptides contain disordered regions or exist entirely in a disordered state[3]. Such systems are referred to as intrinsically disordered proteins (IDPs), although it may be more precise to classify them as 'multi-structure proteins' with low-energy conformational transitions or simply 'intrinsically flexible systems'[4]. These systems are challenging to study due to their highly dynamic nature and their properties' strong dependence on various environmental conditions, including pH, temperature, redox potential, and the presence of other molecular entities such as receptors or ligands[5]. They are also prone to oligomerization, aggregation or nanodroplet formation[6]. One example of such a system is the monomeric form of 1-42 amyloid-β (Aβ42), a fundamental component of toxic Aβ42 oligomers[7] and fibrils associated with neurodegenerative diseases, including Alzheimer's disease[8]. An interesting approach to expedite the study of Aβ fibril formation is the mutation of residues Leu17 and Leu34, which can be efficiently cross-linked in the fibril form, to cysteine with subsequent disulfide bond formation, forcing the peptide conformation to obtain a mature fibril structure without disturbing the fibril formation process [9].

Disulfide bonds are the most common post-translational modification of proteins, occurring in over 20% of proteins with known structures[10]. They fulfill a myriad of roles, including structural and enzymatic stabilization, cycle regulation, and response to stress conditions and are popularly used in industry[11]. Despite their high importance, disulfide bonds are often marginalized or even omitted in studies. Therefore, in this work, we present methods for simulating the presence or absence of disulfide bonds in proteins during simulations, as well as a novel approach allowing for the dynamic formation and disruption of disulfide bonds during simulations based on finite distance restraints[12].

Among the numerous options and parameters available for use in molecular dynamics (MD) simulations, two of them hold paramount importance and can significantly impact the results: the choice of the force field and the sampling method[13]. Together, they influence how efficiently the conformational space of the studied system is explored and whether the obtained results are reliable. While most force fields for proteins have been developed over many years primarily based on well-structured molecules, the study of IDPs necessitates the consideration of methods optimized and well-benchmarked for such systems[14, 15]. For instance, it has been demonstrated that older force fields, such as ff99sb[16], ff14sb[17], CHARMM22[18], and CHARMM36[19], yield results that more deviate from experimental data than newer parameter sets, such as ff19sb[20] and CHARMM36m[21]. Notably, parameters for elements other than proteins, particularly water[22] and ions[23], are especially significant in the case of studies involving IDPs[24]. To achieve satisfactory sampling of the conformational space, more advanced techniques, such as replica exchange MD[25] or conventional MD simulations consisting of multiple independent trajectories, each of reasonable duration, should be employed[26]. However, running simulations with more and longer trajectories can be very costly, especially for large systems comprising more than a single protein chain. Consequently, one can leverage powerful supercomputers or GPUs[27] to accelerate classical all-atom simulations or opt for simplified models, including coarse-grained ones, in which groups of atoms are represented by single interaction centers or beads. Such simplifications not only expedite the calculation of a single MD step but also permit the use of longer time steps and, by reducing the number of degrees of freedom, smooth out the energy landscape, thereby accelerating processes within the simulation[28]. However, most coarse-grained force fields are based on statistical analyses of databases and are thus less suitable for studying IDPs, which are underrepresented in the training data[29]. One approach to address this issue is the utilization of a physics-based coarse-grained force field, such as UNRES[30], a component of the UNICORN package[31], or coarse-grained methods developed for IDPs, such as AWSEM-IDP[32]. Nevertheless, with the development of the Martini model[33], the most popular coarse-grained force field to date[33], it has been discovered that accurate scaling of interactions between solutes and solvents can be a key factor enabling MD studies of IDPs[34].

In this work, using the monomeric form of Aβ42 and its L17C/L34C mutant as examples, we demonstrate how to employ all-atom and coarse-grained force fields in two of the most popular simulational packages, Amber[35] (with all-atom ff19sb and coarse-grained SIRAH[36] force fields) and Gromacs[37] (with the coarse-grained Martini 3 force field[33] corrected for protein-water interactions[34]), to study the dynamics of both folded and disordered proteins, with the option to incorporate static or dynamic treatments of disulfide bonds[38].

PROTOCOL:

When following any tutorial on molecular dynamics, it is important to always consult appropriate manuals, in this protocol for Amber (https://ambermd.org/Manuals.php) or for GROMACS (https://manual.gromacs.org/). They thoroughly explain all program syntax, flags and scope of use.

0. Preparation of the PDB file:
   a. download a PDB file from the PDB (https://www.rcsb.org/), PMDB (http://srv00.recas.ba.infn.it/PMDB/) or other database or other source
   b. remove all of the molecules and atoms that should not be included in the studies
   c. it is usually safer to remove hydrogen atoms (by e.g. using reduce -Trim *.pdb) or to use external tool to predict correct protonation state for the system (PDB2PQR[39]: https://server.poissonboltzmann.org/pdb2pqr)
   d. for convenience, external tools and server, such as CHARMM-GUI[40] (https://www.charmm-gui.org/) can be used for the system generation, especially if we would like to include other molecules, such as lipids (micelles, lipid bilayers), or post-translational modifications
   e. it may be necessary to rebuild missing fragments of the structure: if they are small (2-3 amino-acid residues missing) it is trivial and can be done manually (approximation of the C⍰ position in text editor), with visualization tool (PyMol) or with any popular software for the structure prediction (Modeller, I-TASSER, AlphaFold); however, if a larger fragment is missing it may significantly impact the results
   f. mutagenesis (more details are provided in the Method 1 section about mutagenesis to include non-native disulfide bonds):
      i. changing mutated amino-acid residues to the standard ones
      ii. point substitutions
      iii. deletions
   g. non-standard amino-acids and other molecules not present in the force field has to be manually parameterized in case of all-atom force field or approximated by existing bead types in coarse-grained representations; for all-atom Amber force fields AnteChamber, which is a part of Amber package, or external server, such as PyRed[41] (https://upjv.q4md-forcefieldtools.org/REDServer-Development/) can be used.

METHOD 1 - Amber Package

Software and hardware

This protocol is based on the use of the Ubuntu 22.04 distribution of Linux, but analogous steps should be applicable to other UNIX-based operating systems. The protocol presents how to efficiently run Amber MD calculations on a desktop computer equipped with a CUDA-compatible GPU; however, similar calculations can be performed analogously with other GPUs or CPU calculations on supercomputers. To execute the procedures described in this protocol on a desktop computer, it is necessary to install and set up the following software.

1. Software, that can be downloaded from the Ubuntu repository:
   a. Bash shell program, tcsh, cmake, gcc, gfortran, flex, bison, patch, bc, xorg-dev, libz-dev, libbz2-dev.
   b. The latest version of the proprietary NVIDIA graphics card driver is required. Alternatively, AMD graphics cards can be used if a special version of Amber is utilized (https://www.amd.com/en/technologies/infinity-hub/amber).
   c. Molecular visualization program. Here, we used PyMOL (for more information see: https://pymol.org/dokuwiki/doku.php?id=installation).
2. Software requiring external sources:
   a. CUDA Toolkit without the graphic card CUDA Driver (note: installation of drivers outside the official repository may cause problems with future upgrades of the operating system). (https://developer.nvidia.com/cuda-downloads)
   b. Visual Molecular Dynamics (VMD) (https://www.ks.uiuc.edu/Research/vmd) that will be used for backmapping from a SIRAH coarse-grained representation to all-atom models.

   c. Latest version of Amber, a software package that facilitates the setup of simulation systems and enables the execution and analysis of MD simulations (https://ambermd.org/GetAmber.php).
      i. License for academic use is currently free.
      ii. It is important to check if all versions of programs listed in point 1 of Software are compatible with the current version of Amber (especially compilers and CUDA, for more information see: https://stackoverflow.com/questions/6622454/cuda-incompatible-with-my-gcc-version).
      iii. After downloading and unpacking the files, run CMake. In the **run_cmake** script, change the flag "DCUDA=TRUE" to enable CMake to install pmemd.cuda, a program that allows running programs on the GPU. Since not all programs are GPU-compatible, it is recommended to first compile the code on the CPU and then on the GPU.
         1. If one would like to use multiple CPU cores for MD simulations in the Amber package, another compilation with the flag "DMPI=TRUE" should be performed, and pmemd.MPI used instead of pmemd to run the simulations. In this case, ensure that MPI or OpenMPI is installed.
      iv. Build and install the code with ***make install*** and then ***make test.serial***, which is a time consuming process.
   d. SIRAH force field package for Amber (http://www.sirahff.com/)
3. The folders ***amber, amber_disul, amber_dyn_disul, amber_analysis, amber_sirah, amber_sirah_analysis, amber_sirah_disul*** can be found in our Supporting Materials. These folders contain scripts that utilize all the aforementioned software for setting up and conducting Amber MD simulations of Aβ42.

System setup, minimization and equilibration

System setup, minimization and equilibration are automatically performed by ***amber/setup_amber.sh*** script, which performs the following operations:

1. Generation of the topology and coordinates files based on the selected force field:
   a. The initial structure of Aβ42, as given in the PDB file ***Ab42_input.pdb*** (due to disordered character of the monomeric Aβ42, there are no experimental structures available, so we used one of the snapshots obtained in the previous MD studies as the initial model[13]), is converted to amber ff19SB topology and coordinate files with the use of **tLeap** program.
   b. The protein is solvated with an OPC water and co-optimized ions (sodium and chloride) are added to neutralize the system and to reach the physiological salt concentration of about 0.15 M. Counter-ions are calculated automatically if *-1* number is specified in the tLeap input file (e.g. "addions2 system Na+ -1"), but salt concentration needs to be inserted manually. There are descriptions of steps made by Amber developers: (https://ambermd.org/tutorials/basic/tutorial8/index.php). Alternatively, on the CHARMM-GUI site, the input generator tool allows for calculation of the salt concentration (https://www.charmm-gui.org/).
   c. The system is placed in a selected type of a periodic boundary condition (PBC) box (in this case: truncated octahedron), the size of which is determined based on the added water layer.
2. Energy minimization is performed. Parameters of the energy minimization procedure are specified in the input file ***md_min.in***, and the bash file ***start_min.sh*** is used to copy generated topology and coordinates files and run the minimization.
   a. During minimization, the optimization method is switched from the steepest descent to the conjugate gradient method, combining stability with efficiency.
   b. Restraints are typically imposed to maintain the initial positions of solute heavy atoms.
3. The system is equilibrated in two successive steps. In the first one, equilibration is performed using the CPU to increase stability in the NPT ensemble. The second step utilizes the GPU to expedite the process without risking instabilities. All parameters for the MD simulation are specified in the input file ***md_eq1.in***, and the bash file ***start_eq1.sh*** is used to initiate the first equilibration run. For the second equilibration run (on GPU), the input file ***md_eq2.in*** and bash file ***start_eq2.sh*** are used. The

simulation time step is set to 2 fs, and the total equilibration time is 1 ns, which should be adequate for a typical system without a lipid bilayer to reach the proper temperature and density. This time can be increased if necessary.

a. During the equilibration process, initial velocities are generated, and the system is heated up to the production temperature (300K) to prevent potential steric clashes.
b. In systems with multiple components, such as ligands or lipids, restraints are applied to groups of atoms to ensure a proper system preparation and to prevent component dissociation resulting from minor structural imperfections that could lead to localized high energy.
c. Equilibration in the NPT ensemble establishes the correct system density, allowing the production phase of the simulation to proceed in the NVT ensemble.
d. A cutoff distance of 0.9 nm is set, and the simulation uses a Langevin thermostat with a collision frequency of 2 ps^{-1} and a Berendsen barostat with isotropic position scaling.
e. To calculate the long-ranged electrostatic interactions, the Particle-Mesh Ewald (PME) method is used, which gives a speed-up to the calculations.
f. For GPU calculations on NVIDIA GPUs, one can use the 'CUDA_VISIBLE_DEVICES' flag to select which GPU unit to use when multiple GPUs are available.

4. Additional equilibration with the use of GPU can be performed by modification of the second equilibration run to be continued as the third subsequent run.

The scripts ***start_min.sh***, ***start_eq1.sh*** and ***start_eq2.sh*** copy the necessary input coordinates and topology files from the respective subdirectories of the previous steps. They then initiate simulations that generate trajectory files, which contain coordinates of the system (snapshots) saved at specified intervals, a 'restart' file with coordinates and velocities (after equilibration), output with simulation information, and details regarding the simulation's progress.

Option 1: Inclusion of disulfide bonds

1. Turning on and off disulfide bond in existing cysteine residues

When simulating a protein that contains disulfide bonds in its wild type form, the amino-acid three-letter code in the PDB file must be changed from CYS to CYX to reflect these bonds. Additionally, the tLeap input file should include a line specifying which sulfur atoms are bonded together. If the simulation aims to reduce or remove these disulfide bonds, the amino-acid code should remain as CYS.

2. Mutation of the residues to cysteines and other similar operations

Mutations in the PDB file can be performed manually with a text editor or using visualization, modeling, or structure prediction tools. If a single point mutation is planned, then the fastest approach should be to use a manual method, however, if more extensive modifications are planned, it may be necessary to use more advanced methods such as I-TASSER or AlphaFold, described briefly in point c below.

a. Manual mutation

To manually mutate any amino acid residue in the PDB file, except for Gly, remove all atoms from the side chain except for the CB atom and change the three-letter code of the residue to the desired amino acid. For Gly, change the amino-acid code for the CA atom only. During system preparation, tLeap will automatically assign any missing atoms. Be cautious as this process may lead to steric clashes; therefore, perform the minimization and equilibration steps with extra care.

b. Visualization-tool mutation (PyMol)

To mutate an amino-acid residue using PyMOL, utilize the Mutagenesis Wizard tool. Firstly, select the new residue name you desire from the list and then choose the existing amino-acid from the sequence that you want to mutate. The tool also allows you to choose a rotamer, but it's important to note that they may still result in steric clashes (https://pymolwiki.org/index.php/Mutagenesis).

c. Use of structure prediction tool

If extensive mutations or deletions are planned, using advanced software like I-TASSER[42] or AlphaFold[43], which predicts structures based on provided sequence and known protein structures, may be advisable. Such tools not only have extensive checks to avoid steric clashes caused by the mutation but, to some extent, can also incorporate local or even global conformational changes upon mutation; therefore, they should provide more reliable initial structures than in case of manual modifications.

3. Insertion of a disulfide bond

a. Selection of the conformation with cysteine residues in the proximity

In some cases, there may be a conformation in the trajectory of the wild-type protein where amino acid side-chains that need to be mutated are in close proximity. To check this, you can use cpptraj with the 'distance' command to measure the distance between the desired CB or CG atoms. If a suitable conformation is found, it can be extracted and then mutated as described above to create the initial conformation.

b. SMD of a structure to bring cysteines together

If the previous method fails or there is no trajectory without mutation available, another option is to run a short steered MD (SMD) simulation. After mutating the chosen amino-acid residues to cysteines, the initial steps (system setup, minimization, and equilibration) should follow the same procedure as for the wild-type version. Then, run a short production run of SMD with distance restraints imposed on the sulfur atoms in cysteine side-chains, reducing the distance between them to 0.2-0.3 nm. Convert the last snapshot to a pdb file, and after verifying the structure with PyMOL or other molecular visualization software, use it as the initial conformation. Scripts and input files are in ***amber_disul/disul_smd***.

4. Selection of a disulfide-bond model

There are two options for applying disulfide bonds in MD simulations using the Amber software: one can use simple covalent bonds, which can be present or absent throughout the entire simulation (static treatment), or one can use a pseudopotential that allows for breaking and reforming disulfide bonds during the simulation (dynamic treatment).

a. Static disulfide-bond treatment

To set up simulations of the Aβ42 double cysteine mutant Leu17Cys/Leu34Cys, execute ***amber_disul/setup_amber_disul.sh***. This script performs operations analogous to those described in points 1 to 4 above with two modifications: (i) The initial structure in the PDB file ***amyloidB_withSS.pdb*** is used as input. This structure of Aβ42 contains a disulfide bond between Cys17 and Cys34, indicated as CYX in the PDB file. (ii) The tleap program has an additional command ***bond*** to set disulfide bonds between sulfur atoms. Tleap detects and incorporates disulfide bonds into the topology file.

b. Dynamic disulfide-bond treatment

To set up simulations of the Aβ42 with dynamic disulfide bond, execute ***amber_dyn_disul/setup_amber_dyn_disul.sh***. This script performs operations analogous to those described in points 1 to 4 above with three modifications: (i) The initial structure in the PDB file ***amyloidB_dynSS.pdb*** is used as input. This structure of Aβ42 contains a disulfide bond between Cys17 and Cys34 which is indicated as 'CYS' in this PDB file, in contrast to the static disulfide bond 'CYX' version. (ii) There is a third equilibration step with restraints placed on sulfur atoms to mimic a breakable bond. (iii) Both the third equilibration and production steps have an additional input file that indicates the sulfur atoms that are restrained.

Option 2: Use of the SIRAH coarse-grained model

To use the coarse-grained SIRAH force field, execute ***amber_sirah/setup_sirah.sh***, which will convert the .pqr file using ***sirah_convert.sh***. The .pqr file can be prepared with charge assignment using the PROPKA tool, for example, *via* the PDB2PQR server. These prepared input files are now ready to be used in tLeap, with syntax analogous to that of all-atom simulations. The only changes required are in the names of the protein and water parameters, which are also explained in the ***README*** file provided by the SIRAH developers.

Similarly, to apply the coarse-grained SIRAH force field to the Aβ42 double Cys mutant, and after preparing a .pqr file with the mutation, run ***amber_sirah_disul/setup_sirah_disul.sh***. In the case of coarse-grained simulations using ***amber_sirah_disul***, the tleap input includes an additional command to set a disulfide bond between the appropriate coarse-grained beads.

Production simulations

To start the production phase, execute the bash script ***amber/MD1/dyn1/start_dyn1.sh***. All the parameters for the MD simulation are specified in the input file ***md_dyn1.in***. In this phase, the simulation time step is set to 2 fs for all-atom simulations, and the production time is 1 µs. For coarse-grained simulations, the simulation time step is set to 20 fs, and the production time is 20 µs. To enhance the simulation speed and stability, an NVT ensemble is typically used for production trajectories (https://ambermd.org/GPULogistics.php). The cut-off distance remains at 0.9 and 1.2 nm for all-atom and coarse-grained simulations, respectively, and Langevin dynamics with a collision frequency of 2 ps^{-1} at a temperature of 300K are applied. The only difference compared to the equilibration phase is the absence of pressure control since a constant volume, not pressure, is maintained. The SHAKE algorithm is used to constrain hydrogen bond lengths to allow for longer timestep and higher stability of the simulation.

If one needs to extend the simulations, navigate to the folder ***MD1/dyn2***, set the desired production steps using the 'nstlim' flag, and execute the bash script ***start_dyn2.sh***. This extension can be performed multiple times, but make sure to use appropriate file naming.

To run multiple trajectories, duplicate the input folder ***MD1*** (i.e ***MD2***, ***MD3***, etc.) as many times as needed (a reasonable minimum for averaging is 3 trajectories) and follow analogous steps as for MD1.

During the simulations, the following files are generated:

1. **traj.nc** file containing atom/bead coordinates,
2. **system_dyn*.nc** restart file with coordinates and velocities, which allows to restart the simulation,
3. **min.out** file with simulation information output,
4. **mdinfo** file with information about the simulation progress, speed of the simulation and the estimated time to finish the production run.

Analysis

There are several analyses that allow to study structural properties of the system based on the MD trajectories. To run some of the most common analyses, execute the bash script ***amber_analysis/run_analysis.sh***. This script performs the following operations:

1. Creates folders for analysis for each trajectory listed in the script, and runs the ***cpptraj*** program with commands provided in the input file ***ptraj_traj.in***. This process generates topology and coordinate files in the **ptraj_MD*** folders after centering the solute in the periodic boundary conditions box (using the 'autoimage' command), followed by the removal of water molecules and ions. Caution while using the ‘autoimage’ command in more complex systems is recommended, as it may cause unwanted shifts of solute molecules. This step is important for speeding up all the following analyses. (Note: if there is any analysis that needs to incorporate initial coordinates or water molecules and ions, it should be run on the initial coordinates and topology files).
    a. be careful not to run any further analysis in a single script if the RMSD calculation has been performed as it may impact the results. Conversely, be cautious about the orientation of the solute in the PBC box when calculating RMSF, or other analyses requiring superpositioning with a reference structure.
2. Runs **cpptraj** program with the commands included in input files **ptraj_rmsd_rg.in**, ***ptraj_pdb.in*** and ***ptraj_image_distance.in***. ***Ptraj_rmsd_rg.in*** calculates RMSD, R_g and RMSF; ***ptraj_pdb.in*** creates PDB files from Amber topology and trajectory files, and ***ptraj_image_distance.in*** calculates the minimum non-self imaged distance, which is used to determine if the periodic box is of sufficient size to prohibit interactions between periodic images.
3. Copies all the resulting text and PDB files with overwritten names of each trajectory to the corresponding folders (***rg***, ***rmsd***, ***rmsf***, ***pdb***, ***mindist***).

Visualizations and backmapping

To visualize snapshots for the all-atom simulations with the minimum and maximum R_g values, run ***amber_analysis/rg_min_max_snap.sh*** script which will:

1. search for the minimum and maximum values of R_g and identify the corresponding structures,
2. extract those structures from the trajectory file and save them in the pdb format.

To visualize snapshots for the coarse-grained SIRAH simulations, run the bash script ***amber_sirah_analysis/rg_min_max_cg_snap.sh***. This script performs operations analogous to those described in points 1 and 2 above with two modifications: (i) The extracted structures are saved in Amber topology and coordinate files. (ii) The VMD script ***sirah_vmdtk.tcl*** is used to map the simulation structures from the coarse-grained representation back to the all-atom representation. This script is provided by SIRAH developers, and can be found in the directory in which SIRAH was extracted during the installation process. After opening VMD with the trajectory, the backmaping is performed with the command ***sirah_backmap***, which backmaps every snapshot in the trajectory, or with the command ***sirah_backmap now***, which backmaps only the current snapshot.

The conformations from pdb files can be visualized by means of a chosen molecular graphics program. To visualize the system, one can use PyMol software. Open the terminal and type **pymol** to execute the program. Then, click **File → Open** and select the PDB file to be visualized. In the side menu on the right hand side of the object name (which, in this case, is the name of the PDB file without the extension), click **A → remove waters**, and then **S → cartoon**. One can use a rainbow representation to visualize the N- to C-termini of the polypeptide chain by clicking **C → spectrum →rainbow**.

METHOD 2 - GROMACS package

Software and hardware

1. To execute the procedures described in this protocol on a personal computer, install and set up the following software:
   a. Python versions 2 and 3 together with mdtraj library (https://mdtraj.org/1.9.4/installation.html). Here, we used Python 2.7 and Python 3.10.
   b. Any molecular visualization program such as VMD (https://www.ks.uiuc.edu/Research/vmd) or Pymol (https://pymol.org/dokuwiki/doku.php?id=installation) or Chimera (https://www.cgl.ucsf.edu/chimera/download.html).
   c. Latest version of GROMACS (we used version 2022.4), a package of programs to set up, carry out and analyze MD simulations (https://manual.gromacs.org/current/install-guide/index.html).
   d. Program martinize2 to generate coarse-grained structures and topology files from atomic structures (https://github.com/marrink-lab/vermouth-martinize).
2. Download the following scripts and packages:
   a. Martini 3 force-field parameter files (packed into a zip file available from webpage http://cgmartini.nl/images/martini_v300.zip).
   b. ***insane.py***, a python script to solvate molecular systems and add ions to the simulation box (http://cgmartini.nl/images/tools/insane/insane.py).
   c. ***PW_rescaling_martini3.py***, a python script to rescale water-protein interactions (https://github.com/KULL-Centre/papers/blob/main/2021/Martini-Thomasen-et-al/PW_rescaling_example/PW_rescaling_martini3.py).
   d. ***initram-v5.sh***, a tool to perform backmapping, i.e. to convert coarse-grained systems to the all-atom representation (http://md.chem.rug.nl/images/tools/backward/backward-v5.zip).
   e. folders ***ABeta*** and ***ABeta_CC*** from our Supporting Materials that contain files that use all the aforementioned scripts and software pieces to set up and perform the Martini simulations of Aβ42. Make sure to update paths in the scripts in folders ***ABeta*** and ***ABeta_CC*** to indicate the aforementioned programs installed on the personal computer.

3. Access to a computer cluster is useful but not necessary to perform the Martini simulations.
    a. The MD simulations can be run on a desktop computer with a multi-core CPU. However, the main advantage of using a computer cluster is that many simulations can be run simultaneously on separate compute nodes.
    b. In this protocol, we provide bash scripts to execute the Martini simulations on a computer cluster equipped with the Slurm queue system.
    c. On a supercomputer, or computer cluster, the installation of necessary software is typically performed by the system administrator. Therefore, one may not need to install the programs mentioned above, or one can inquire with the administrator about the software installation.

System setup

Execute the bash script ***setup_scripts/setup_martini_AB.sh***. This script performs the following operations:

1. The initial structure of Aβ42, as given in the PDB file ***Ab42_input.pdb***, is converted to a coarse-grained representation using the martinize2 program. Martini topology and structure files are generated based on the information from the provided PDB file.
2. The coarse-grained protein structure is placed in a cubic box of side length of 9 nm using the Gromacs editconf tool.
3. Protein is solvated using the Python script ***insane.py***. Sodium and chloride ions are added to neutralize the system and to reach the physiological salt concentration of 0.15 M. Note that flag "-salt" applied to the ***insane.py*** script indicates the salt concentration in molar units.
4. Energy minimization is performed. Parameters of the energy minimization procedure are specified in the Gromacs input file ***minimization.mdp***.
5. The force-field parameters describing non-bonded interactions between the protein and water beads are re-scaled by a factor λ using the bash script ***rescaling.sh***. The values of parameter λ are set here to 0.96, 0.97 ... 1.10.

Option 1. Scaling of solute-solvent interactions

The goal of performing a series of MD simulations with different λ-values is to determine an optimal value of parameter λ which yields the best agreement between the simulation and experiment[34]. The range of values of parameter λ can be changed in the bash script ***rescaling.sh***. Note that λ = 1 corresponds to unscaled solute-solvent interactions, i.e. to the original Martini 3 force field. Such scaling is performed as in the abovementioned point 5 of the ***setup_scripts/setup_martini_AB.sh*** script.

Option 2. Simulations of Aβ42 with a disulfide bond

To set up Martini simulations of the Aβ42 Leu17Cys/Leu34Cys mutant, denoted here by Aβ42$_{\text{disul}}$, execute the bash script ***ABeta_CC/setup_scripts/setup_martini_AB.sh***. This script performs operations analogous to those described in points 1 to 5 above with two modifications: (i) The PDB file ***amyloidB_withSS.pdb*** with a Aβ42$_{\text{disul}}$ structure is used as input. This structure contains a disulfide bond between Cys17 and Cys34. (ii) The martinize2 program is called with the flag "-cys auto". With this flag, martinize2 automatically detects and incorporates disulfide bonds into the topology file as constraints on the distance between the cysteine side-chain SC1 beads. Here, the distance between the SC1 beads of Cys17 and Cys34 is constrained at 0.24 nm using the LINCS algorithm[44]. Importantly, accurately including disulfide bonds in the input structure may require a high-resolution initial structure or preprocessing, such as all-atom energy minimization.

Option 3. Simulations of a protein with a well-defined structure

To set up Martini simulations of a protein with a well-defined structure, it is necessary to use an elastic network[45], i.e. a set of harmonic restraints (or so-called “rubber bands”) that ensure that the secondary and tertiary structure of the protein is preserved in the course of the simulation. To this end, the martinize2 program should be called with additional flags, e.g. "-elastic -ef 700.0 -el 0.5 -eu 0.9 -ea 0 -ep 0 -scfix". This imposes harmonic restraints between backbone beads within the lower and upper distance cut-off of 0.5 and

0.9 nm, respectively. The spring constant of the harmonic restraints is set to 700 kJ mol^{-1} nm^{-2}. In addition, the flag "-scfix" adds extra torsional potentials between backbone beads and sidechain beads[46]. These potentials are included in the force field to prevent unphysical flipping of side chains in β-strands.

It is important to note that the elastic network is created on the basis of a reference secondary structure that needs to be indicated to the marinize2 program, which can be done by adding the flag "-dssp mkdssp". With this flag, the DSSP program (https://github.com/cmbi/dssp) is used to determine the secondary structure elements based on the input structure. DSSP recognizes secondary structure based on the positions of both heavy atoms and hydrogens, therefore, it is important to use a high-resolution initial structure or perform preprocessing, such as all-atom energy minimization, to generate accurate secondary-structure restraints.

Option 4. Simulations of a multidomain protein with intrinsically disordered linkers

To set up Martini simulations of mix-folded proteins, in which folded protein domains are linked by intrinsically disordered regions, we recommend to follow the method and procedures introduced by the Lindorff-Larsen group[34].

Equilibration

Execute the bash script ***equilibration/run_equilibration.sh*** to perform equilibration simulations. The same range of λ-values as in the system setup protocol above (i.e. λ = 0.96, 0.97 ... 1.10) is used here. All the parameters of the equilibration simulations are specified in the Gromacs input file ***equilibration.mdp***. Importantly, the integration time step is set to 2 fs to allow for slow relaxation of the system. The equilibration time is set to 10 ns, which should be enough for typical protein systems to reach the specified temperature and density.

In the first step, the script ***run_equilibration.sh*** generates a binary tpr file that contains information about all the simulation parameters (including force field parameters), about the system topology, and about the starting coordinates and velocities of all the beads forming the simulation system. Here, the starting velocities are generated on the basis of the Maxwell-Boltzmann distribution with the specified temperature. In the second step, the MD simulation is started using the tpr file as input.

The simulation box size after the equilibration can be checked at the end of the ***eqb1.00.gro*** file. Here, the simulation box is a cube of the side length of 9.23 nm.

Production simulations

Enter folder ***production*** with 15 Slurm scripts ***qsub_096.sh*** … ***qsub_110.sh*** that can be used to submit simulations for execution on a computer cluster. Each of the Slurm scripts starts one simulation with a specified λ-value, ranging from λ=0.96 to λ=1.1. All the simulation parameters are given in the Gromacs input file ***dynamic.mdp***. Here, the integration time step is set to 20 fs and the total simulation time is 20 µs. The temperature and pressure are kept constant at T = 300 K and p = 1 bar, respectively, using the velocity-rescaling thermostat and the Parrinello-Rahman barostat. Nonbonded interactions are treated with the Verlet cutoff scheme. The cutoff for van der Waals interactions is set to 1.1 nm. Coulomb interactions are treated using the reaction-field method with a cutoff of 1.1 nm and dielectric constant of 15. Frames are saved every 1 ns.

Similarly as in the equilibration procedure, the Slurm scripts firstly generate binary tpr files containing all the information needed to start the simulations (i.e. system topology, MD simulation parameters, and starting coordinates and velocities of the beads). Then the MD simulations are started using the tpr files as input. Each of the production simulations takes about two days on a 12-core compute node.

The following files are generated in the course of the simulations:

1. ***run_pbc_lambda*.gro*** files containing the Cartesian coordinates of the beads at the final step of the simulation.
2. ***run_pbc_lambda*.xtc*** trajectory files.

3. ***run_pbc_lambda*.edr*** binary files that contain time-series non-trajectory data such as energy breakdowns, temperature and pressure of the simulation system.
4. ***run_pbc_lambda*.log*** files containing information about the overall performance of the simulation. These files also include energy breakdowns as well as the instantaneous values of the pressure and temperature of the simulation system.
5. ***run_pbc_lambda*.cpt*** binary files required for restarting the simulation. The complete information about the system state is stored in these checkpoint files.

Analysis

1. Execute the bash script ***production/script_pbc.sh*** which performs the following tasks: Firstly, the Gromacs ***trjconv*** tool with the flags "-pbc mol -center" makes the Aβ42 peptide centered inside the simulation box. The resulting trajectory files are named ***run_pbc_lambda*.xtc*** and moved to directories ***AB_**** for all the simulated values of parameter λ. Secondly, the radius of gyration of the Aβ42 molecule, R_g, is calculated for all of the simulation structures and recorded in output files ***rg_run_pbc_lambda*.xvg***, which are moved to folder ***/analysis_results/RG***. It should be noted that the radius of gyration is calculated here using the coordinates of all the beads forming Aβ42.
2. Execute the bash script ***production/validation.sh*** which calculates the average radius of gyration for each of the trajectories with λ = 0.96, …, 1.1. The statistical errors are estimated using block averaging. Namely, a given trajectory is divided into 10 blocks of 2 µs each, and the average value of R_g is calculated for each of the blocks. The standard deviation of these black-averaged R_g values estimates the statistical error of the average value of R_g in the whole trajectory.
3. Execute the bash script ***production/dmax.sh*** which calculates the maximum dimension of the Aβ42 molecule, D_{max}, for each of the simulation structures in each of the PBC-treated trajectories. D_{max} is defined as the maximum distance between all pairs of solute beads. The D_{max} values are recorded in files ***dmax*.xvg*** and moved to folder ***dmax***.
4. Execute the bash script ***production/contactdata.sh*** which calculates contact frequency maps based on the PBC-treated trajectories. The contact frequency maps are recorded in files ***contact*.xvg*** and moved to folder ***contactmap***. The only parameter in the contact map calculation is the cutoff distance below which a contact is assumed to exist between a pair of beads. Here, the default value of 0.7 nm is used (flag cutoff=0.7 in the Python script ***contactdata.py***). Thus, if the distance between any bead of a given amino-acid residue and any bead of another amino-acid residue is smaller than 0.7 nm, then this residue pair is deemed to be in contact.

Backmapping and visualization

1. The peptide conformations sampled during the Martini simulations and recorded in the trajectory files can be conveniently visualized using the VMD program. Execute the command ***vmd peptide.gro run_pbc_lambda1.00.xtc*** to display the trajectory of Aβ42 centered in the simulation box. In the VMD command line type ***pbc box*** to show the simulation box. To display the beads in the van der Waals representation, go to **Graphics → Representations** and from the **Drawing Method** list select **VDW**. By default, the backbone beads are depicted in pink whereas the side chain beads are shown in yellow. The slider in the main window can be used to move between frames. The play button can be used to visualize the whole trajectory.
2. Execute script ***production/backmapping/backmapping.sh*** to convert the coarse-grained structures generated in the Martini simulations to all-atom structures. This script is based on the backmapping algorithm introduced by Wassenaal et al.[47] and performs the following operations:
   a. An all-atom topology file is created using the PDB file ***Ab42_input.pdb*** and the CHARMM36 force field.
   b. A single coarse-grained structure is extracted from the PBC-corrected Martini trajectory file.

c. The coarse-grained structure is mapped back to an all-atom structure using the ***backward.py*** program. This backmapped structure is relaxed in several short cycles of energy minimization and position-restrained MD simulations with the CHARMM36 force field.

d. Points b and c are iterated over all the coarse-grained structures obtained from the Martini simulations, combining the resulting all-atom structures into a single trajectory file.

The backward program consists of several scripts and mapping definition files, including the Python script ***backward.py*** which conducts the actual backmapping; the bash wrapper ***initral-v5.sh*** that performs the energy minimization and position-restraint MD simulations using Gromacs with the CHARMM36 force field; files ****.map*** containing definitions of the mapping between the all-atom and coarse-grained representations; and the Python script ***init .py*** to interpret the map files.

It should be noted that the all-atom structures can be used not only to visualize the simulation trajectory but also to perform analyses analogous to those introduced in the Analysis section above.

3. Use VMD to visualize the all-atom structures of the peptide. Open the trajectory file obtained from the backmapping procedure in the VMD program: **vmd Ab42_input.pdb merged.xtc**. To change the color background go to **Graphics**→**Colors** and next choose **Display** → **Background** from **Categories** and set the color to white. Using the frame slider, choose the frame you want to visualize and save it as an image. To do this, open **File** → **Render** and choose **Start Rendering** command.
4. An alternative is to use Chimera. To open a PDB or GRO file in Chimera, go to **File → Open** and choose the file. To change the background color, go to **Favorites → Preferences** and in the **Category** menu choose **Background**. In option **Background color** set the color by changing the RGB values. To show specific atoms, select a whole chain, **Select → Chain → (no ID)**, and then select **Actions → Atoms/Bonds → show**. To change the representation style, select the whole chain, next click **Actions → Atoms/Bonds → ball & stick** to show atoms of the side chain. To show backbone atoms, hide the ribbon representation **Actions → Ribbon → hide**. Save the view as an image by choosing **File**→ **Save Image** and, after setting the resolution in the **Image Size** option, use the **Save** button.

REPRESENTATIVE RESULTS:

Execution of the example files provided in the Supporting Materials generates a complete procedure for preparing initial files for MD simulation, performing minimization, equilibration, and production runs, and conducting basic result analysis. Figure 1 illustrates the differences between all-atom and coarse-grained simulation boxes in Amber and Gromacs packages. In Amber, a truncated octahedron periodic box is used to save computational time by omitting calculations for the water molecules placed in the corners of the box. In contrast, Gromacs coupled with the Martini 3 force field uses a more expensive but simpler cubic box in terms of symmetry operations and further analysis.

Since the monomeric Aβ42 is intrinsically disordered and lacks a stable structure for reference in RMSD calculations, RMSD plots are not presented here. Nevertheless, RMSD can be easily calculated with minor modifications of the analysis scripts. The relevant properties that can be showcased here include the radius of gyration and the maximum dimension (Figure 2 for the results obtained using the Amber package). The presented results show that in the all-atom simulations with the ff19SB force field, Aβ42 tends to adopt a rather compact state with an average R_g of 1.29 ± 0.14 nm, and it becomes even more compact if the disulfide bond is present (Table 1). In the SIRAH coarse-grained model, Aβ42 exhibits a similar tendency, but the peptide conformations are even more compact, with an average R_g of 1.13 ± 0.11 nm without the disulfide bond present. To verify whether there are any contacts between periodic images (in other words, to check if the simulation box is sufficiently large), one can calculate the minimum-image distance (Figure 3). In our case, it is clear that there are no interactions between periodic images within the cut-off range (0.9 and 1.2 nm for ff19sb and SIRAH models, respectively), and the lowest observed values are above 1.6 and 2.6 nm for ff19sb and SIRAH models, respectively. These values are safely above the threshold, indicating that the periodic box was sufficiently large.

Figure 4 shows results of the Martini simulations with λ=1, i.e. with no rescaling of the protein-water interactions. The radius of gyration as a function of time for Aβ42 and Aβ42$_{disul}$ is shown, respectively, in panels A and B in Figure 4. Because of the constraint imposed by the covalent bond between Cys17 and Cys34, Aβ42$_{disul}$ exhibits smaller R_g values than Aβ42. The abrupt fluctuations of R_g in time indicate that the simulation system is well equilibrated on the time scale of several microseconds. The maximum dimension as a function of time for Aβ42 and Aβ42$_{disul}$ is shown, respectively, in panels C and D in Figure 4. It should be noted that D_{max} values do not exceed 7 nm, which is smaller than the simulation box side length of about 9 nm. Thus, the simulation box is large enough to ensure that the peptide does not interact with its periodic image during the Martini simulations.

Figure 5 shows the average radius of gyration as a function of the water-protein interaction rescaling parameter λ used in the Martini simulations. As one would expect, the average R_g increases monotonically with λ, both in the case of Aβ42 and Aβ42$_{disul}$. Indeed, increasing the protein-water interactions should result in expanded conformations of IDPs[34, 48] . Importantly, by comparing the average R_g from the simulations with the R_g from experimentals, one can determine an optimal value of parameter λ which yields the best agreement between the simulation and experiment[34].

The contact maps of Aβ42 and Aβ42$_{disul}$ obtained from the Martini trajectory with λ=1 and no backmapping are shown, respectively, in panels A and B in Figure 6. The color scale indicates the natural logarithm of contact frequency. Consequently, the points in red, yellow and blue represent frequent, transient and rare contacts, respectively. The contacts between amino acid residues that are close sequentially, namely contacts (i,i+1), (i,i+2), (i,i+3) and (i,i+4), are not shown. The presence of the disulfide bond between Cys17 and Cys34 is clearly reflected in the map shown in panel B. In general, contact map analysis can yield valuable insights into intramolecular interactions stabilizing protein conformations.

The result of mapping coarse-grained simulation structures back to the all-atom representation is illustrated in Figure 7. Specifically, the final structure obtained from the Martini simulation of Aβ42$_{disul}$ with λ=1 is shown here in the van der Waals representation (panel A), in the stick-and-ball representation (panel B) and the ribbon representation (panel C). The all-atom structures obtained from the backmapping calculations can be used not only for visualization purposes but also for analysis and comparison with experimental data[34]. Figure 8 compares the most extended and most compact structures obtained from the all-atom and coarse-grained simulations. Structures with and without the disulfide bond are depicted.

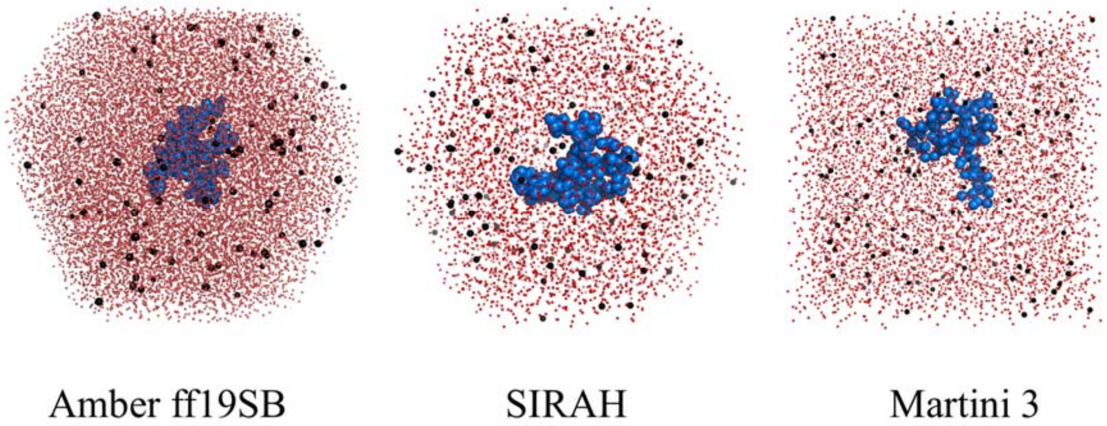

Figure 1: Visualization of the entire simulation boxes using both all-atom and coarse-grained methods. Atoms or interaction centers are represented as spheres in red (water), black (ions) and blue (Aβ42 peptide). To save computational time, the MD simulations in the Amber package employed a truncated octahedron box, while Martini 3 simulations used a standard cubic box.

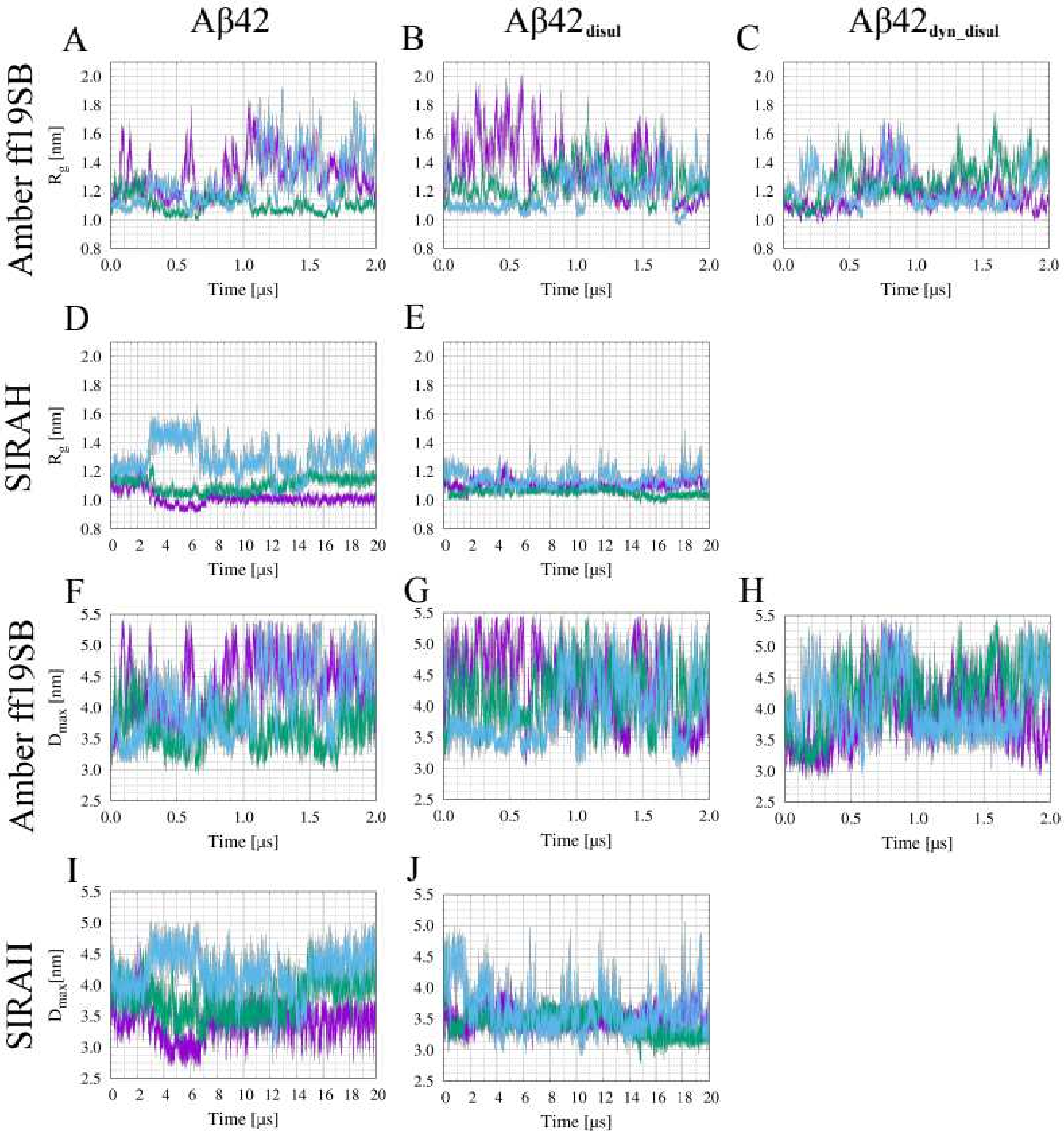

Figure 2: Time evolution of the radius of gyration (A-E) and of the maximum dimension (F-J) obtained from the MD simulations employing the all-atom Amber force field ff19SB (A-C and F-H) and SIRAH coarse-grained force field (DE and IJ) without the disulfide bond (A, D, F, I), with the static disulfide bond (B, E, G, J) and with the dynamic treatment of the disulfide bond (C and H).

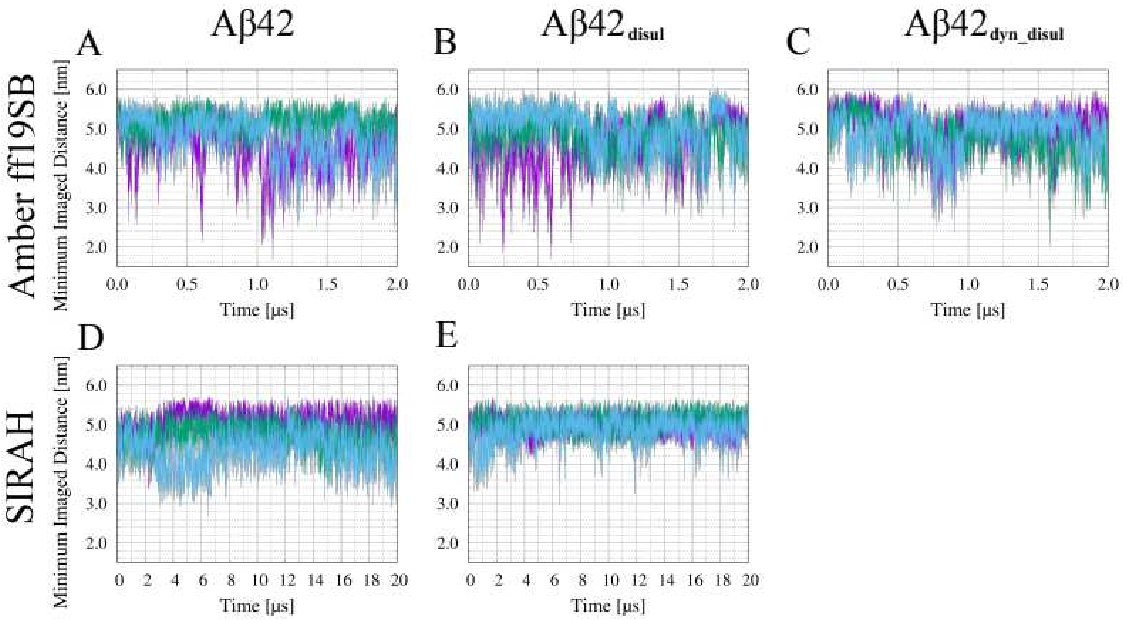

Figure 3: Time evolution of the minimum non-self image distance in the all-atom Amber ff19SB simulations of Aβ42 (A), of Aβ42 with the static disulfide bond (B), and of Aβ42 with the dynamic disulfide bond (C), as well as in the SIRAH coarse-grained simulations of Aβ42 (D) and of Aβ42 with the static disulfide bond (E).

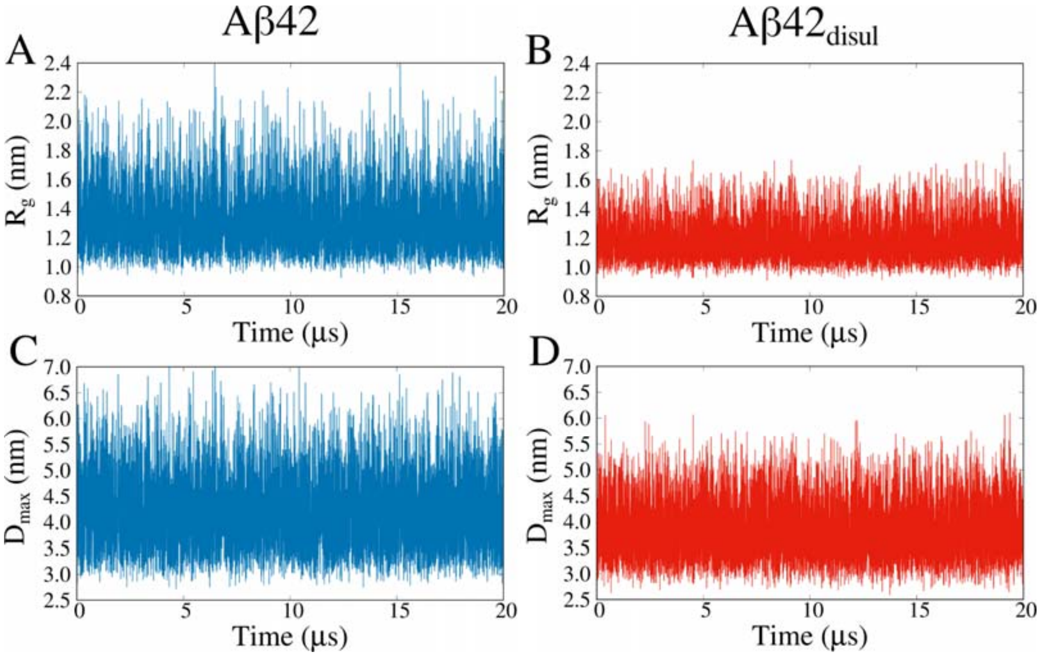

Figure 4: Time evolution of the radius of gyration (A,B) and of the maximum dimension (C,D) obtained from the Martini simulations of Aβ42 (A,C) and Aβ42$_{\mathrm{disul}}$ (B,D) with λ=1. Both Aβ42 and Aβ42$_{\mathrm{disul}}$ exhibit large conformational fluctuations at thermodynamic equilibrium. Because of the constraint imposed by the covalent bond between Cys17 and Cys34, Aβ42$_{\mathrm{disul}}$ exhibits smaller values of R_{g} and D_{max} than Aβ42.

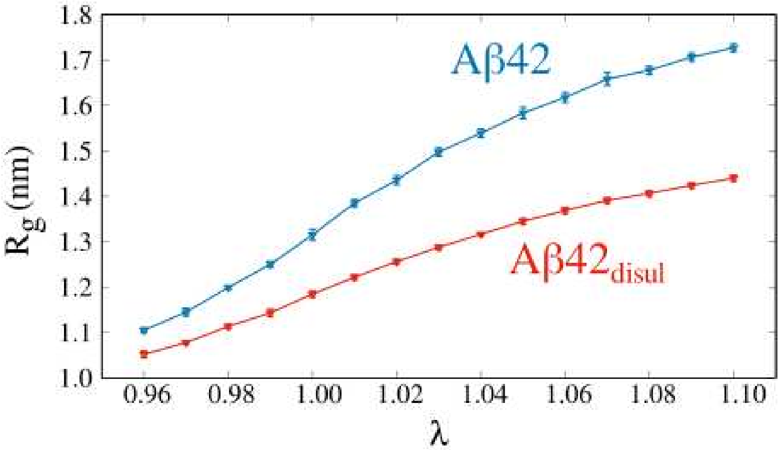

Figure 5: Average radius of gyration as a function of the water-protein interaction rescaling parameter λ obtained from the Martini simulations of Aβ42 (blue) and Aβ42$_{disul}$ (red). The error bars indicate the standard deviation calculated over a single trajectory using block averaging.

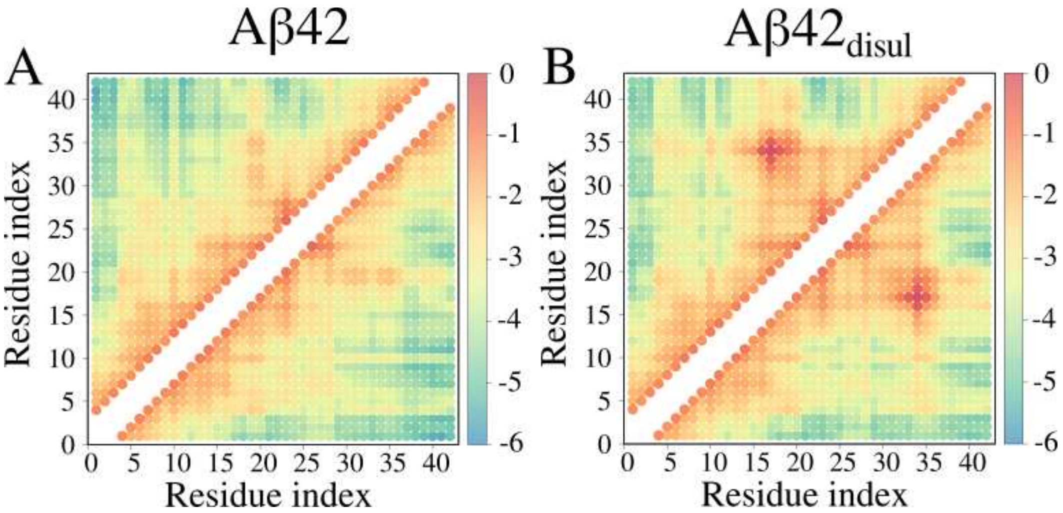

Figure 6: Contact maps obtained from coarse-grained structures of Aβ42 (A) and Aβ42$_{disul}$ (B) generated in the Martini simulations with λ=1 (no scaling of solute-solvent interactions). The color scale indicates the natural logarithm of contact frequency. The points in red, yellow and blue represent frequent, transient and rare contacts, respectively.

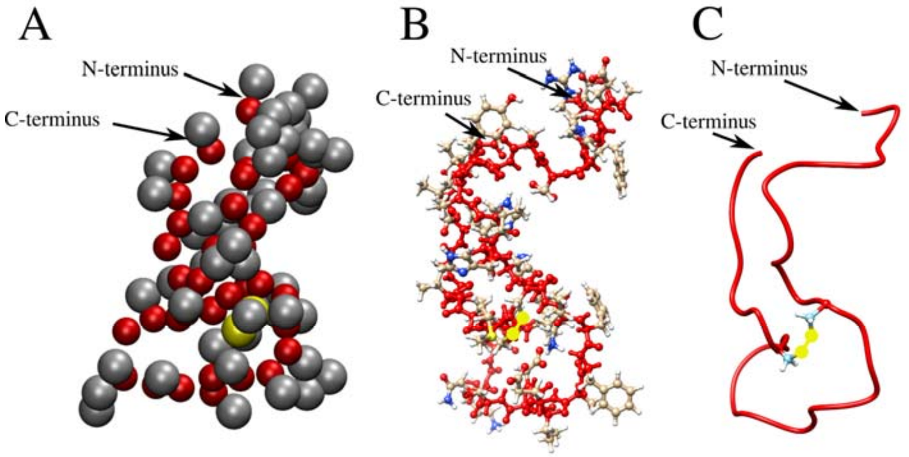

Figure 7: The final structure obtained from the Martini simulation of Aβ42$_{disul}$ is shown in three different representations. (A) The coarse-grained structure, where the backbone and sidechain beads are colored in red and gray, respectively. The SC1 beads of Cys17 and Cys34 are highlighted in yellow. (B) The all-atom structure, as obtained from the backmapping calculation, is displayed in a stick-and-ball representation. The backbone atoms are shown in red. The sidechain carbon, nitrogen, and oxygen atoms are shown in gray, blue, and red, respectively. The sulfur atoms forming the covalent disulfide bridge are highlighted in yellow. (C) The all-atom structure is presented in a ribbon representation: the main chain is shown in red, while sidechains are not displayed, except for Cys17 and Cys34, which are shown in the stick representation.

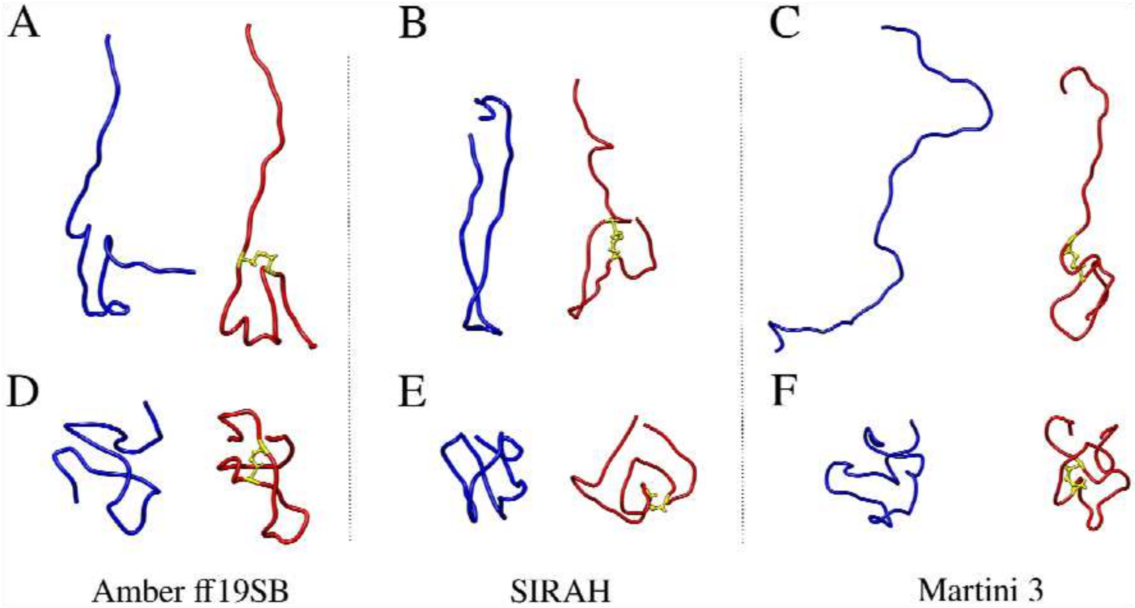

Figure 8: Ribbon representation of the structures from the MD simulations with the all-atom Amber ff19SB (A, D), coarse-grained SIRAH (B, E) and coarse-grained Martini 3 (C, F) force fields with maximum (A, B, C) and minimum (D, E, F) values of the radius of gyration in blue and red, respectively.

DISCUSSION:

In this work, we demonstrated how all-atom and coarse-grained force fields can be utilized to study a model IDP - monomeric Aβ42. To ensure the high reliability of the results, state-of-the-art methods were employed. Specifically, we used the all-atom Amber ff19sb force field for proteins, coupled with the four-point OPC water model[49] and a co-optimized ions model[23] to mimic physiological salt concentration, approximately 0.15M NaCl[50]. Additionally, we employed the SIRAH coarse-grained force field within the Amber package and the Martini 3 coarse-grained model and force field within the Gromacs package. In the latter approach, solute-solvent interactions were scaled up and down, and the results were compared to the radius of gyration values to ensure the correct compactness of the simulation structures. The use of a coarse-grained representation significantly accelerated the simulations by approximately 1-2 orders of magnitude, with no significant impact on the results. However, it is worth noting that the use of coarse-grained models can complicate the analysis, as it often requires reconstruction to all-atom models, and some atomistic details are lost due to the simplification of protein representation. Still, coarse-grained models seem to be a method of choice when it comes to extensive simulations of large IDP complexes [51] and biomolecular condensates [52, 53].

Overall, the results obtained indicate that all the methods produced largely unstructured conformations, as expected for monomeric Aβ42. However, this outcome is not always achieved, especially when older methods are employed [13, 54]. Due to the much larger computational cost associated with all-atom simulations, only three trajectories, each of 2 microseconds, were executed, which may not provide the highest level of robustness in the results. In contrast, with the coarse-grained SIRAH and MARTINI methods, simulations were 10 times longer, and convergence was reached more quickly due to the reduction in the number of degrees of freedom, resulting in a lower total computational cost. This approach of employing two entirely different computational methods, such as all-atom and coarse-grained force fields, is recommended for obtaining highly reliable results, especially in the absence of or with limited experimental data.

Earlier computational studies suggest that R_g values for Aβ42 are in range of 0.8 to 1.2nm[55–57], while the experimental lowest found R_g values are about 1nm, which should correspond to the most compact conformations of the monomers. However, it should be noted that experimentally observed hydrodynamic radius strongly depends on the conditions and is found to be in the range of 0.8 to 1.5nm[57], further confirming the molecular flexibility of Aβ42. More recent studies still present very different R_g values, ranging from 1.0 [58] to 1.6 nm[59] obtained from SANS experiments. In the context of this evidence, all of the calculated values in this work fit into experimental and theoretical R_g ranges.

Point mutations are commonly introduced to proteins and peptides to either decrease [60] or increase [9] their aggregation properties, depending on the study's objectives. In the case of Aβ42, a series of MD simulations were conducted on the L17C/L34C mutant with a disulfide bond acting as a cross-link between the amino-acid residues that come into contact in the fibrillar form. The designed mutant is known to efficiently form fibrils with both the wild-type and mutated Aβ42 [9], emphasizing some of the fibril-like contacts, including Q15-V36, L17-L34, and F19-I32. Interestingly, our all-atom and coarse-grained simulations, even without the disulfide bond, exhibit a significant number of conformations forming latter two contacts, and these conformations were further stabilized when a disulfide bond was added (see Figure 6 and Table 2).

Table 1. Average Rg and Dmax with SD (for Amber and SIRAH force field averaged and SD are calculated over three trajectories, while for Martini it is calculated over a single trajectory with block averaging).

	Amber ff19SB Aβ42	Amber ff19SB Aβ42$_{disul}$	Amber ff19SB Aβ42$_{dyn\text{-}disul}$	SIRAH Aβ42	SIRAH Aβ42$_{disul}$	Martini Aβ42 with λ=1	Martini Aβ42$_{disul}$ with λ=1
R_g (nm)	1.29±0.14	1.27±0.01	1.22±0.07	1.13±0.11	1.10±0.04	1.32±0.01	1.19±0.01
D_{max} (nm)	4.22±0.42	4.16±0.06	4.09±0.21	3.83±0.31	3.48±0.09	4.20±0.04	3.83±0.03

Table 2. Frequency (in %) of contacts with SD in the simulations. A contact between atoms in a pair is counted as '1' in a given snapshot if the CB atoms in all-atom representation or centers of interactions are within a 0.7 nm range.

	Amber ff19SB Aβ42	Amber ff19SB Aβ42$_{disul}$	Amber ff19SB Aβ42$_{dyn\text{-}disul}$	SIRAH Aβ42	SIRAH Aβ42$_{disul}$	Martini Aβ42 with λ=1	Martini Aβ42$_{disul}$ with λ=1
Q15-V36	12.54±6.40	9.09±9.94	1.55±1.67	3.45±2.65	0.00±0.00	1.63±0.35	7.11±0.63
L17-L34 (C17-C34)	59.10±29.75	100.00±0.00	100.00±0.00	43.35±7.25	100.00±0.00	5.39±0.75	100.00±0.00
F19-I32	82.34±10.27	58.20±41.84	70.05±36.89	25.72±22.65	26.88±36.38	12.02±0.92	24.72±1.09

The use of a coarse-grained SIRAH model allows us to obtain approximately a 6-fold decrease in computational time needed to calculate 1,000,000 steps compared to the all-atom ff19SB force field (Table 3). Taking into account a 10 times longer timestep value, this provides us with a 60-fold speed-up when using a low-end GPU, while the total number of interaction centers decreased by about 13 times. These values are in agreement with those obtained by SIRAH developers for a much larger system[36]. However, it should be noted that in our case, SIRAH simulations on a GPU did not scale at all, even with the use of a much more powerful GPU, while for the all-atom system, a 4-fold speed-up was observed, which is most likely caused by the small size of the studied system. An 80-fold speed-up is observed in the Amber all-atom force field when using a state-of-the-art GPU instead of a 20-core CPU node, whereas this difference is only about 10-fold if SIRAH is used in the Amber package. It should be noted that use of Martini 3 model in the Gromacs package provides about a 15-fold speed-up compared to SIRAH in the Amber package when a similar CPU is used, which most likely comes from a much different coarse-grained model used (especially for water), different cutoff methods and ranges. Moreover, the performance of Martini in Gromacs scales well even when a high number of CPU cores is used (e.g. 3 times speed-up when 64-cores are used instead of 12).

Table 3. Computational time for various all-atom and coarse-grained approaches for monomeric Aβ42 with and without disulfide bond treatment showed with multiple configurations to demonstrate that these simulations can be run on a modern PC or a supercomputer center.

Force field	**Disulfide bond**	**Number of interaction centers of:**		**Timestep and cutoff values**	**Real time for 1,000,000 steps**	**Hardware**	
		peptide	total			CPU/GPU model	Cores no & frequency
All-atom Amber ff19SB	none	627	60718	2 fs 0.9 nm	580 min 0 s	2*Intel(R) Xeon(R) E5-2670 v2	20 cores 2500 MHz
					27 min 20 s	NVIDIA GeForce RTX 3060	3584 cores 1320 MHz
					12 min 30 s	NVIDIA GeForce RTX 3080	8704 cores 1800 MHz
					9 min 40 s	NVIDIA GeForce RTX 4070Ti	7680 cores 2640 MHz
	static	609	62396	2 fs 0.9 nm	10 min 20 s	NVIDIA GeForce RTX 4070Ti	7680 cores 2640 MHz
					6 min 50 s	NVIDIA GeForce RTX 4090	16384 cores 2520 MHz

	dynamic	611	62378	2 fs 0.9 nm	17 min 20 s	NVIDIA GeForce RTX 3070Ti	6144 cores 1830MHz
					11 min 0 s	NVIDIA GeForce RTX 4070Ti	7680 cores 2640 MHz
Coarse-grained SIRAH	none	195	4774	20 fs 1.2 nm	50 min 20 s	2*Intel(R) Xeon(R) E5-2670 v2	20 cores 2500 MHz
					4 min 40 s	NVIDIA GeForce RTX 2060S	2176 cores 1810 MHz
					4 min 35 s	NVIDIA GeForce RTX 4090	16384 cores 2520 MHz
	static	195	4926	20 fs 1.2 nm	4 min 40 s	NVIDIA GeForce RTX 2060S	2176 cores 1810 MHz
Coarse-grained MARTINI	none	96	6722	20 fs 1.1 nm	1 min 07 s	AMD EPYC 7763	64 cores 2450 MHz
					4 min 20 s	Intel(R) Core(TM) i7-5820K	6 cores 330 MHz
					3 min 25 s	Intel(R) Xeon(R) E5-2670 v3	12 cores 2300 MHz
					5 min 37 s	Intel(R) Xeon(R) E5-2640	6 cores 2500 MHz
	static	96	6770	20 fs 1.1 nm	5 min 5 s	Intel(R) Core(TM) i7-5820K	6 cores 330 MHz
					5 min 41 s	Intel(R) Xeon(R) E5-2640	6 cores 2500 MHz
					5 min 33 s	Intel(R) Xeon(R) X5650	6 cores 2670 MHz

In general, the provided protocols can be easily employed for routine simulations of most biomacromolecules, with the flexibility to change the force field, water model, and other components of the system. This adaptability can be managed even by less experienced users. While it is recommended to conduct classical MD simulations on supercomputers, access to which is often limited, the use of GPU computations and coarse-grained representations allows these simulations to be successfully run on modern desktop computers, which are usually equipped with up to 32-core CPUs. This democratizes access to scientific research by eliminating the need for substantial budgets, thereby promoting inclusivity in scientific studies.

ACKNOWLEDGMENTS:

This work has received financial support from the National Science Centre, Poland, *via* grants SONATA No 2019/35/D/ST4/03156 (to PK) and OPUS No 2020/39/B/NZ1/00377 (to BR). PK and PS gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016624. BR thanks the Centre of Informatics – Tricity Academic Supercomputer and networK (CI TASK) in Gdansk, Poland, for access to computer resources. PR, BR and MMA acknowledge Polish high-performance computing infrastructure PLGrid for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through PLL/2023/04/016485.

REFERENCES:

1. Whitford, D. *Proteins: Structure and Function*. Wiley. (2005).
2. Anfinsen, C.B. Principles that govern the folding of protein chains. *Science*. **181** (4096), 223–230 (1973).
3. Dunker, A.K., Brown, C.J., Lawson, J.D., Iakoucheva, L.M., Obradović, Z. Intrinsic disorder and protein function. *Biochemistry*. **41** (21), 6573–6582 (2002).
4. Sgourakis, N.G., Yan, Y., McCallum, S.A., Wang, C., Garcia, A.E. The Alzheimer's peptides Abeta40 and 42 adopt distinct conformations in water: a combined MD / NMR study. *Journal of molecular biology*. **368** (5), 1448–1457 (2007).
5. Uversky, V.N. Intrinsically disordered proteins and their environment: effects of strong denaturants, temperature, pH, counter ions, membranes, binding partners, osmolytes, and macromolecular crowding. *The protein journal*. **28** (7-8), 305–325 (2009).
6. Levine, Z.A., Larini, L., LaPointe, N.E., Feinstein, S.C., Shea, J.-E. Regulation and aggregation of intrinsically disordered peptides. *Proceedings of the National Academy of Sciences of the United States of America*. **112** (9), 2758–2763 (2015).
7. Sengupta, U., Nilson, A.N., Kayed, R. The Role of Amyloid-β Oligomers in Toxicity, Propagation, and Immunotherapy. *EBioMedicine*. **6**, 42–49 (2016).
8. Breijyeh, Z., Karaman, R. Comprehensive Review on Alzheimer's Disease: Causes and Treatment. *Molecules* . **25** (24), at <https://www.ncbi.nlm.nih.gov/pubmed/33302541> (2020).
9. Shivaprasad, S., Wetzel, R. An intersheet packing interaction in A beta fibrils mapped by disulfide cross-linking. *Biochemistry*. **43** (49), 15310–15317 (2004).
10. Wiedemann, C., Kumar, A., Lang, A., Ohlenschläger, O. Cysteines and Disulfide Bonds as Structure-Forming Units: Insights From Different Domains of Life and the Potential for Characterization by NMR. *Frontiers in chemistry*. **8**, 280 (2020).
11. Bulaj, G. Formation of disulfide bonds in proteins and peptides. *Biotechnology advances*. **23** (1), 87–92 (2005).
12. Smardz, P., Sieradzan, A.K., Krupa, P. Mechanical Stability of Ribonuclease A Heavily Depends on the Redox Environment. *The journal of physical chemistry. B*. **126** (33), 6240–6249 (2022).
13. Krupa, P., Quoc Huy, P.D., Li, M.S. Properties of monomeric Aβ42 probed by different sampling methods and force fields: Role of energy components. *The Journal of chemical physics*. **151** (5), doi: 10.1063/1.5093184 (2019).
14. Mu, J., Liu, H., Zhang, J., Luo, R., Chen, H.-F. Recent Force Field Strategies for Intrinsically Disordered Proteins. *Journal of chemical information and modeling*. **61** (3), 1037–1047 (2021).
15. Klein, F., Barrera, E.E., Pantano, S. Assessing SIRAH's Capability to Simulate Intrinsically Disordered Proteins and Peptides. *Journal of chemical theory and computation*. **17** (2), 599–604 (2021).
16. Hornak, V., Abel, R., Okur, A., Strockbine, B., Roitberg, A., Simmerling, C. Comparison of multiple Amber force fields and development of improved protein backbone parameters. *Proteins*. **65** (3), 712–725 (2006).
17. Maier, J.A., Martinez, C., Kasavajhala, K., Wickstrom, L., Hauser, K.E., Simmerling, C. ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. *Journal of chemical theory and computation*. **11** (8), 3696–3713 (2015).
18. Mackerell, A.D., Jr, Feig, M., Brooks, C.L., 3rd Extending the treatment of backbone energetics in protein force fields: limitations of gas-phase quantum mechanics in reproducing protein conformational distributions in molecular dynamics simulations. *Journal of computational chemistry*. **25** (11), 1400–1415 (2004).
19. Huang, J., MacKerell, A.D., Jr CHARMM36 all-atom additive protein force field: validation based on comparison to NMR data. *Journal of computational chemistry*. **34** (25), 2135–2145 (2013).
20. Tian, C. *et al.* ff19SB: Amino-Acid-Specific Protein Backbone Parameters Trained against Quantum Mechanics Energy Surfaces in Solution. *Journal of chemical theory and computation*. **16** (1), 528–552 (2020).
21. Huang, J. *et al.* CHARMM36m: an improved force field for folded and intrinsically disordered proteins. *Nature methods*. **14** (1), 71–73 (2017).
22. Piana, S., Donchev, A.G., Robustelli, P., Shaw, D.E. Water dispersion interactions strongly influence simulated structural properties of disordered protein states. *The journal of physical chemistry. B*. **119** (16), 5113–5123 (2015).
23. Sengupta, A., Li, Z., Song, L.F., Li, P., Merz, K.M., Jr Parameterization of Monovalent Ions for the OPC3,

OPC, TIP3P-FB, and TIP4P-FB Water Models. *Journal of chemical information and modeling*. **61** (2), 869–880 (2021).

24. Somavarapu, A.K., Kepp, K.P. The Dependence of Amyloid-β Dynamics on Protein Force Fields and Water Models. *Chemphyschem: a European journal of chemical physics and physical chemistry*. **16** (15), 3278–3289 (2015).

25. Sugita, Y., Okamoto, Y. Replica-exchange molecular dynamics method for protein folding. *Chemical physics letters*. **314** (1), 141–151 (1999).

26. Qi, R., Wei, G., Ma, B., Nussinov, R. Replica Exchange Molecular Dynamics: A Practical Application Protocol with Solutions to Common Problems and a Peptide Aggregation and Self-Assembly Example. *Peptide Self-Assembly: Methods and Protocols*. 101–119 (2018).

27. Lee, T.-S. *et al.* GPU-Accelerated Molecular Dynamics and Free Energy Methods in Amber18: Performance Enhancements and New Features. *Journal of chemical information and modeling*. **58** (10), 2043–2050 (2018).

28. Kmiecik, S., Gront, D., Kolinski, M., Wieteska, L., Dawid, A.E., Kolinski, A. Coarse-Grained Protein Models and Their Applications. *Chemical reviews*. **116** (14), 7898–7936 (2016).

29. Latham, A.P., Zhang, B. Unifying coarse-grained force fields for folded and disordered proteins. *Current opinion in structural biology*. **72**, 63–70 (2022).

30. Liwo, A. *et al.* A unified coarse-grained model of biological macromolecules based on mean-field multipole-multipole interactions. *Journal of molecular modeling*. **20** (8), 2306 (2014).

31. Liwo, A. *et al.* Chapter Two - Scale-consistent approach to the derivation of coarse-grained force fields for simulating structure, dynamics, and thermodynamics of biopolymers. *Progress in Molecular Biology and Translational Science*. **170**, 73–122 (2020).

32. Wu, H., Wolynes, P.G., Papoian, G.A. AWSEM-IDP: A coarse-grained force field for intrinsically disordered proteins. *The journal of physical chemistry. B*. **122** (49), 11115–11125 (2018).

33. Souza, P.C.T. *et al.* Martini 3: a general purpose force field for coarse-grained molecular dynamics. *Nature methods*. **18** (4), 382–388 (2021).

34. Thomasen, F.E., Pesce, F., Roesgaard, M.A., Tesei, G., Lindorff-Larsen, K. Improving Martini 3 for Disordered and Multidomain Proteins. *Journal of chemical theory and computation*. **18** (4), 2033–2041 (2022).

35. Case, D.A. *et al.* AMBER; University of California, San Francisco, 2023.

36. Klein, F. *et al.* The SIRAH force field: A suite for simulations of complex biological systems at the coarse-grained and multiscale levels. *Journal of structural biology*. **215** (3), 107985 (2023).

37. Bekker, H. *et al.* GROMACS - A PARALLEL COMPUTER FOR MOLECULAR-DYNAMICS SIMULATIONS. *4th International Conference on Computational Physics (PC 92)*. 252–256 (1993).

38. Krupa, P., Sieradzan, A.K., Mozolewska, M.A., Li, H., Liwo, A., Scheraga, H.A. Dynamics of Disulfide-Bond Disruption and Formation in the Thermal Unfolding of Ribonuclease A. *Journal of chemical theory and computation*. **13** (11), 5721–5730 (2017).

39. Jurrus, E. *et al.* Improvements to the APBS biomolecular solvation software suite. *Protein science: a publication of the Protein Society*. **27** (1), 112–128 (2018).

40. Jo, S., Kim, T., Iyer, V.G., Im, W. CHARMM-GUI: a web-based graphical user interface for CHARMM. *Journal of computational chemistry*. **29** (11), 1859–1865 (2008).

41. Vanquelef, E. *et al.* R.E.D. Server: a web service for deriving RESP and ESP charges and building force field libraries for new molecules and molecular fragments. *Nucleic acids research*. **39** (Web Server issue), W511–7 (2011).

42. Yang, J., Zhang, Y. I-TASSER server: new development for protein structure and function predictions. *Nucleic acids research*. **43** (W1), W174–81 (2015).

43. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature*. **596** (7873), 583–589 (2021).

44. Hess, B., Bekker, H., Berendsen, H.J.C., Fraaije, J.G.E.M. LINCS: A linear constraint solver for molecular simulations. *Journal of computational chemistry*. **18** (12), 1463–1472 (1997).

45. Periole, X., Cavalli, M., Marrink, S.-J., Ceruso, M.A. Combining an Elastic Network With a Coarse-Grained Molecular Force Field: Structure, Dynamics, and Intermolecular Recognition. *Journal of chemical theory and computation*. **5** (9), 2531–2543 (2009).

46. Herzog, F.A., Braun, L., Schoen, I., Vogel, V. Improved Side Chain Dynamics in MARTINI Simulations of Protein-Lipid Interfaces. *Journal of chemical theory and computation*. **12** (5), 2446–2458 (2016).

47. Wassenaar, T.A., Pluhackova, K., Böckmann, R.A., Marrink, S.J., Tieleman, D.P. Going Backward: A Flexible Geometric Approach to Reverse Transformation from Coarse Grained to Atomistic Models. *Journal*

of chemical theory and computation. **10** (2), 676–690 (2014).
48. Best, R.B., Zheng, W., Mittal, J. Balanced Protein-Water Interactions Improve Properties of Disordered Proteins and Non-Specific Protein Association. *Journal of chemical theory and computation*. **10** (11), 5113–5124 (2014).
49. Izadi, S., Anandakrishnan, R., Onufriev, A.V. Building Water Models: A Different Approach. *Journal of Physical Chemistry Letters*. **5** (21), 3863–3871 (2014).
50. Terry, C.A., Fernández, M.-J., Gude, L., Lorente, A., Grant, K.B. Physiologically relevant concentrations of NaCl and KCl increase DNA photocleavage by an N-substituted 9-aminomethylanthracene dye. *Biochemistry*. **50** (47), 10375–10389 (2011).
51. Różycki, B., Boura, E. Conformational ensemble of the full-length SARS-CoV-2 nucleocapsid (N) protein based on molecular simulations and SAXS data. *Biophysical chemistry*. **288**, 106843 (2022).
52. Dignon, G.L., Zheng, W., Kim, Y.C., Best, R.B., Mittal, J. Sequence determinants of protein phase behavior from a coarse-grained model. *PLoS computational biology*. **14** (1), e1005941 (2018).
53. Anila, M.M., Ghosh, R., Różycki, B. Membrane curvature sensing by model biomolecular condensates. *Soft matter*. **19** (20), 3723–3732 (2023).
54. Wu, K.Y., Doan, D., Medrano, M., Chang, C.-E.A. Modeling structural interconversion in Alzheimers' amyloid beta peptide with classical and intrinsically disordered protein force fields. *Journal of biomolecular structure & dynamics*. **40** (20), 10005–10022 (2022).
55. Massi, F., Peng, J.W., Lee, J.P., Straub, J.E. Simulation study of the structure and dynamics of the Alzheimer's amyloid peptide congener in solution. *Biophysical journal*. **80** (1), 31–44 (2001).
56. Li, X., Mehler, E.L. Simulation of molecular crowding effects on an Alzheimer's beta-amyloid peptide. *Cell biochemistry and biophysics*. **46** (2), 123–141 (2006).
57. Nag, S. *et al.* Nature of the amyloid-beta monomer and the monomer-oligomer equilibrium. *The Journal of biological chemistry*. **286** (16), 13827–13833 (2011).
58. Festa, G. *et al.* Aggregation States of Aβ1-40, Aβ1-42 and Aβp3-42 Amyloid Beta Peptides: A SANS Study. *International journal of molecular sciences*. **20** (17).
59. Zhang-Haagen, B., Biehl, R., Nagel-Steger, L., Radulescu, A., Richter, D., Willbold, D. Monomeric Amyloid Beta Peptide in Hexafluoroisopropanol Detected by Small Angle Neutron Scattering. *PloS one*. **11** (2), e0150267 (2016).
60. Maisuradze, G.G. *et al.* Preventing fibril formation of a protein by selective mutation. *Proceedings of the National Academy of Sciences of the United States of America*. **112** (44), 13549–13554 (2015).