Contact-based molecular dynamics of structured and disordered proteins in a coarse-grained model: fixed contacts, switchable contacts and those described by pseudo-improper-dihedral angles

Łukasz Mioduszewski[a], Mateusz Chwastyk[a], Marek Cieplak[a,*]

[a]*Institute of Physics, Polish Academy of Sciences, Al. Lotników 32/46, 02-668 Warsaw, Poland*

Abstract

We present a coarse-grained C_α-based protein model that can be used to simulate structured, intrinsically disordered and partially disordered proteins. We use a Go-like potential for the structured parts and two different variants of a transferable potential for the disordered parts. The first variant uses dynamic inter-residue contacts that form and disappear quasi-adiabatically during the simulation. By using specific geometric criteria we distinguish sidechain-sidechain, sidechain-backbone and backbone-backbone contacts. The second variant is a non-radial multi-body pseudo-improper-dihedral (PID) potential that does not include time-dependent terms but requires more computational resources. Our model can simulate in reasonable time thousands of residues on millisecond time scales.

Keywords: coarse grained model; intrinsically disordered proteins; molecular dynamics; implicit solvent; contact map;

PROGRAM SUMMARY

Program Title: cg.f
Licensing provisions: MIT
Programming language: FORTRAN
Nature of problem(approx. 50-250 words):

---

*Corresponding author.
E-mail address:* mc@ifpan.edu.pl

Simulations of one or more protein chains, structured, intrinsically disordered or partially disordered. Calculating their equilibrium and kinetic properties in processes including but not limited to: folding, aggregation, conformational changes, formation of complexes, aggregation, response to deformation. All those processes need long simulation times or many trajectories to properly sample the system.

Solution method(approx. 50-250 words):

A molecular dynamics implicit-solvent coarse-grained model where each pseudoatom represents one amino acid residue. Residues are harmonically connected to form a chain. The system evolves according to Langevin dynamics. The backbone stiffness potential involves bond angle and dihedral angle terms (or a chirality term in the structured case). Residues interact via modified Lennard-Jones or Debye-Hueckel potentials.

The potential is different for structured and disordered parts of a protein. A Go model contact map is used for the structured parts, where an interaction between two residues is attractive if effective spheres associated with their heavy atoms overlap in the native structure [1,2,3]. Non-attractive contacts use only the repulsive part of the Lennard-Jones potential. The potential for the disordered parts has two versions: in the quasi-adiabatic variant, the contacts are formed dynamically based on the geometry of the chain and are quasi-adiabatically turned on and off [4]. In the slower, but more accurate pseudo-improper-dihedral variant the Lennard-Jones term is multiplied by a pseudo-improper-dihedral term that gives similar geometric restrictions with a time-independentHamiltonian [5].

Proteins can be pulled by their termini to mimic stretching by an Atomic Force Microscope. Multiple protein chains can be put in a simulation box with periodic boundary conditions or with walls that may be repulsive or attractive for the residues. The walls may move to mimic pulling or shearing deformations. The program accepts PDB files or protein sequences with optional user-defined contact maps. It saves the results in an output file (summary), a PDB file (structure) and a map file (contact map in a given snapshot).

Additional comments including Restrictions and Unusual features (approx. 50-250 words):
The program supports openMP (effective up to 8 cores, use 4 cores for optimal speed-up).

References

[1] M. Sikora, J. I. Sułkowska and M. Cieplak, Mechanical strength of 17 134 model proteins and cysteine slipknots, PLoS Computational Biology **5** (2009)

e1000547.

[2] J. I. Sułkowska and M. Cieplak, Selection of optimal variants of go-like models of proteins through studies of stretching, Biophys. J. **95** (2008) 317491.

[3] K. Wołek, Á. Gómez-Sicilia and M. Cieplak, Determination of contact maps in proteins: a combination of structural and chemical approaches, J. Chem. Phys. **143** (2015) 243105.

[4] Ł. Mioduszewski and M. Cieplak, Disordered peptide chains in an α-c-based coarse-grained model, Phys. Chem. Chem. Phys. **20**, (2018) 19057-19070.

[5] Ł. Mioduszewski, B. Różycki and M. Cieplak, Pseudo-improper-dihedral model for intrinsically disordered proteins, J. Chem. Theory Comput. accepted manuscript (2020).

1. Usage

The program is a single FORTRAN file (`cg.f`). After compilation, it requires exactly one argument: the name of the inputfile. Names of all the other files that may be needed must be written in the inputfile. Possible input and output files are shown on Fig. 1. All files used in the program are in the plain text format. Each line of the inputfile has two words separated by a single space: the name of a variable and the value it should have. All variables have a predefined default value, so only the non-default variables need to be included in the inputfile. More technical details (like the list of the variables) are summarized in the `README.txt` file. This program is distributed as a `.tar` archive and as a Code Ocean capsule (doi [available after publishing]).

2. Equation of motion

Each pseudoatom is centered on the position of the C_α atom of the amino acid it represents. We use Langevin dynamics with the following equation of motion for the i-th residue in a chain:

$$m\frac{d^2\vec{r_i}}{d^2t} = \vec{F_i} - \gamma\frac{d\vec{r_i}}{dt} + \vec{\Gamma_i} \qquad (1)$$

where m is the average amino acid mass (the variable `lmass` allows to turn on different amino acid masses), $\vec{r_i}$ is the position of the residue, $\vec{F_i}$ is the force

```
Map files prot.map
(optional contact maps
in a custom format)
```

```
Sequence file input.txt
(protein sequence
in a custom format)
```

```
Parameters param.txt
(potential parameters
for disordered proteins)
```

OR

```
PDB file input.pdb
(protein structure
in PDB format)
```

```
Map file input.map
(optional contact map
in a different format)
```

```
Inputfile inputfile
(all non-default variables)
    klenstr 6
    file result
For disordered proteins:
  seqfile input.txt
  paramfile param.txt
        OR
For ordered proteins:
  pdbfile input.pdb
      lcpot F
      lpdb T
    Optionally:
  mapfile input.map
```

```
Compile the program:
  gfortran cg.f -o cg
and run it (with the name of the
inputfile as the only argument):
     ./cg inputfile
```

```
PDB file result.pdb
(protein structures in
PDB format, saved
every ksave timesteps)
```

```
Output file result.out
(various statistics, saved
every kwrite timesteps)
```

```
Restart file
result_timestamp.rst
(needed for restart, saved
every krst timesteps)
```

```
Map file result.map
(optional custom contact
map, saved every
ksave timesteps)
```
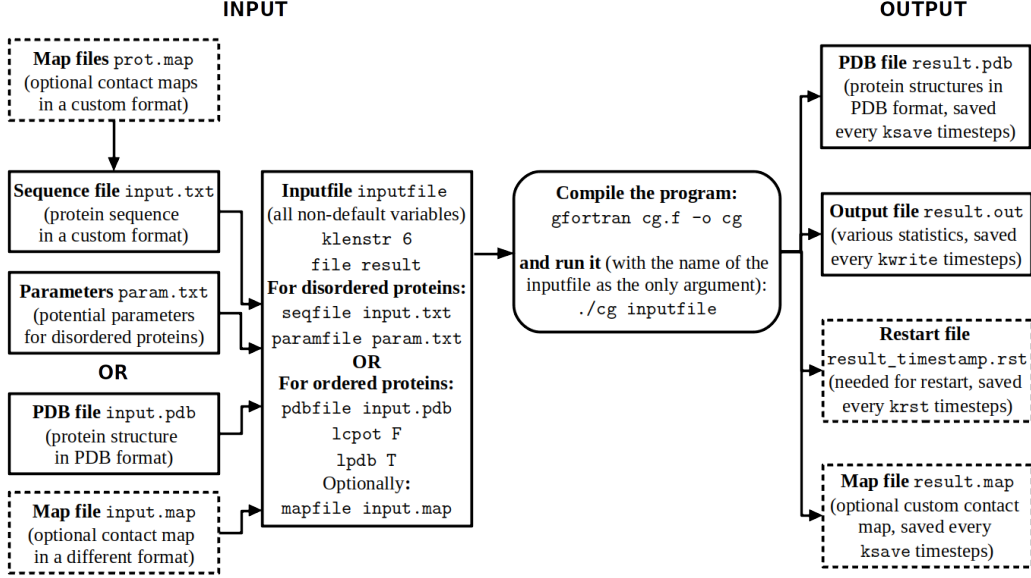
Figure 1: A scheme visualizing the usage of the program.

resulting from the potential, $\gamma = 2m/\tau$ is the damping coefficient and $\vec{\Gamma}_i$ is the thermal white noise with the variance $\sigma_T^2 = 2\gamma k_B T$. The implicit solvent is implemented via this noise and the damping. The default damping coefficient $\gamma = 2m/\tau$ is chosen to represent an overdamped case with diffusional (not ballistic) dynamics. More realistic values of γ should be about 25 times larger [1, 2], but adopting them would lead to a much longer conformational dynamic.

The time unit $\tau \approx 1$ ns was verified for a different model with the same equations of motion [3], so the value of τ is only orientational. The equations of motion are integrated by the fifth order predictor-corrector algorithm [4, 5]. One τ corresponds to 200 integration steps.

The energy is expressed in units of ϵ, where $\epsilon \approx 1.5$ kcal/mol [6, 7]. The internal unit of energy in the code is also ϵ. The distances in the input and output of the program are in the units of Å. However, the distances used in the integrating module are dimensionless: they are divided by 5 Å; hence the conversion unit `unit` of 5. (so a distance ,,10 Å" in the input is processed as ,,2" during the integration of the equations of motion).

The residues in one chain are tethered harmonically with the spring constant $k = 100$ Å$^{-2} \cdot \epsilon$ and the equilibrium distance between them is taken

4

as 3.8 Å. If the simulation is based on a PDB file, equilibrium distances are taken from that file and can be different from 3.8 Å. Due to limitations in the chain numbering in the PDB files, the program allows only 153 unique chain labels in a single file. However, there are ways to use more than one file [8, 9]. Setting the variable `lwritexyz` to true causes the program to use a much simpler XYZ format, which does not distinguish between chains (and prints only the x,y,z positions of each residue).

3. Backbone stiffness

Local interactions between amino acids separated by at most three residues in the protein sequence are known to stabilize folded proteins and improve their folding kinetics [10], and are even more important in determining the conformational ensembles of disordered proteins [11]. Those local interactions require using special backbone stiffness potentials (although in this program residues i and $j \geq i + 3$ can also attract each other in non-local interactions). Below we describe those potentials. Unstructured parts of a chain can use only bond angle and dihedral angle terms.

3.1. Chirality potential for structured parts

Chirality potential is defined [12] as $V^{CH} = \frac{1}{2}e_{chi}\Sigma_{i=2}^{i=N-2}(C_i - C_i^n)^2$ where N is the number of residues and the chirality of the residue i is $C_i = (\vec{v}_{i-1} \times \vec{v}_i) \cdot \vec{v}_{i+1}/d_0^3$ where $\vec{v}_i = \vec{r}_{i+1} - \vec{r}_i$ and d_0 is equal to the length of $\vec{v}_i$ in the native structure (usually very close to 3.8 Å). C_i^n is the chirality of residue i in the native structure. This dependence on the native structure restricts usage of the chirality potential only to structured proteins (the variable `lpdb` must be set to T). This potential can be turned on by setting the variable `lchiral` to T. The amplitude e_{chi} of the potential can be tuned using the variable `echi` (the default value is 1, in the internal units of energy $\epsilon \approx 110$ pN·Å). It is computationally faster than the backbone stiffness potential based on the bond and dihedral angle terms defined below, but it is important to note that using chirality potential (with the default `echi`) lowers the effective room temperature of the system to 0.25 (in the program units, which correspond to ϵ/k_B). The room temperature for structured proteins corresponds to the minimum of the median folding time and other properties of the system, defined in [12].

5

3.2. Bond and dihedral angle potentials

The bond angle between residues $i - 1, i, i + 1$ is defined as θ_i, and the dihedral angle between residues $i - 2, i - 1, i, i + 1$ is ϕ_i. Backbone stiffness potentials based on those angles are turned on by the variable `langle`.

3.2.1. Structured parts

The bond angle potential for the structured parts is harmonic with the minimum in the bond angle θ_n present in the native structure: $V_\theta = k_\theta(\theta - \theta_n)$, where the value of k_θ is defined by the variable `CBA` (its default value is 30, in ϵ/rad^2 units).

The bond angle term is usually used together with the dihedral term, which can be turned on by setting the variable `ldih` T. The dihedral term has two forms. If the variable `ldisimp` is set to T, the harmonic form is used: $V_p hi^{simp} = \frac{1}{2}k_\phi(\phi - \phi_n)$, where ϕ_n is the dihedral angle in the native structure and k_ϕ can be set in the variable `CDH` (with the default value of 3.33, in ϵ/rad^2 units). The room temperature for this set of parameters has not been determined.

When `ldisimp` is false, a more sophisticated dihedral potential is used: $V_\phi = K_1[1 - \cos(\phi - \phi_n)] + K_3[1 - \cos(3(\phi - \phi_n))]$, where the K_1 and K_3 parameters are specified by variables `CDA` and `CDB`, both with a default value 0.66 (in ϵ/rad^2 units), respectively. This set of values (together with the bond angle potential) corresponds to room temperature 0.7, at which the kinetics of folding of structured proteins is optimal [12].

3.2.2. Unstructured parts

For the backbone stiffness of the unstructured parts we use a 3-letter alphabet distinguishing Gly, Pro and X residues, where X is one of the 18 other amino acids. For the bond angle we check only what the second and third residues are in the three consecutive residues forming the angle, which gives 9 possible choices. Including the first residue would give 27 possibilities, but the symmetry between the first and the third residue in a bond angle can be broken (because proteins are chiral). The bond angle potential has the form of a sixth degree polynomial. Its coefficients for all 9 cases are listed in the parameter file specified by the variable `paramfile`.

The dihedral angle requires a consideration of four consecutive residues but we distinguish only the two central ones, which again gives 9 possible combinations. The dihedral potential is governed by a formula $a \sin(\phi) +$

$b \cos(\phi) + c \sin^2(\phi) + d \cos^2(\phi) + e \sin(\phi) \cos(\phi)$, where the values of the coefficients a, b, c, d, e for all 9 cases are also listed in the `paramfile`.

The coefficients for both the bond angle and the dihedral potentials were obtained [13] by fitting the formulas to a potential resulting from the inverse Boltzmann method applied to a random coil database. This potential was shown to correctly describe properties of proteins in denaturating conditions [14]. Thus our backbone stiffness has no preference towards any secondary structure, but α-helices and β-sheets can still be formed by the backbone-backbone contacts (see section 4.2).

Ghavami et al [14] distinguished only the middle residue and whether it precedes proline, so only 6 from the 9 combinations in the `paramfile` are different.

The backbone stiffness is important in setting the effective room temperature for the model [12], so in this case we checked what temperature corresponds best to the experimental end-to-end distances for the polyprolines of various lengths [15]. We chose polyproline because its conformation is mostly determined by its backbone stiffness, so the parameters unrelated to the backbone stiffness could be optimized in the correct temperature. Temperatures between 0.35 and 0.4 ϵ/k_B turned out to be optimal, which is consistent with our energy unit $\epsilon \approx 1.5$ kcal/mol. Then 0.38 $\epsilon/k_B \approx 300$ K.

3.2.3. User-defined potential

The backbone stiffness potential may be also specified in a table where each row corresponds to a different angle and each column to a different Gly/Pro/X combination (in the same manner as in the previous section). The file `stiffnesstabularized.txt` contains the Boltzmann inversion potentials from Ghavami et al [14].

4. Non-local potential

The non-local potential is different for structured and (partially) disordered proteins. Possible variants of the potential together with the variables needed to switch between them are schematically shown on Fig. 2.

4.1. Structured parts

If the variables `lpdb` and `lallatom` are set to true and a PDB file is provided, the program automatically constructs from the PDB file a contact map based on the overlap of heavy atoms [16] after assigning to each of them
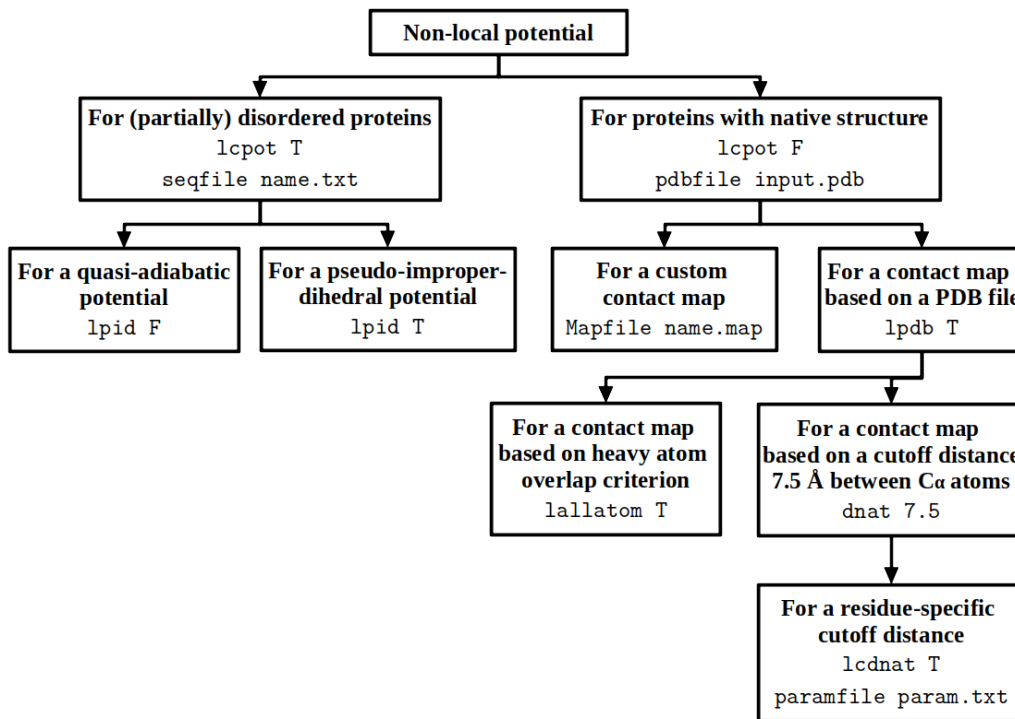
Non-local potential

For (partially) disordered proteins
`lcpot T`
`seqfile name.txt`

For proteins with native structure
`lcpot F`
`pdbfile input.pdb`

For a quasi-adiabatic potential
`lpid F`

For a pseudo-improper-dihedral potential
`lpid T`

For a custom contact map
`Mapfile name.map`

For a contact map based on a PDB file
`lpdb T`

For a contact map based on heavy atom overlap criterion
`lallatom T`

For a contact map based on a cutoff distance 7.5 Å between Cα atoms
`dnat 7.5`

For a residue-specific cutoff distance
`lcdnat T`
`paramfile param.txt`

Figure 2: A scheme showing variants of the non-local potential.

a radius that takes into account different positions of atoms in the amino acid (e. g. it distinguishes the radii of C_α and C_β atoms) [17]. In the case of an incomplete PDB file a simpler contact map based on C_α-C_α distances may be used (by setting `lallatom` false).

An analysis of many Go model versions showed that a simple variant where each attractive contact is described by a Lennard-Jones (L-J) potential with the depth 1 ϵ gives the best agreement with the experiments in which proteins were stretched by the Atomic Force Microscope (AFM) tips [18].

It is important to note that the overlap of heavy atoms is just one possible criterion for defining a contact. More sophisticated criteria may involve the chemical identity of the overlapping heavy atoms [19]. Constructing such contact maps is not possible in the program presented here, but we provide a webserver for computing contact maps using a rCSU method: `info.ifpan.edu.pl/~rcsu/`

The simple L-J potential describing an attractive interaction between

residues i and j has the form $\epsilon\left[\left(\frac{r_{minij}}{r}\right)^{12} - 2\left(\frac{r_{minij}}{r}\right)^{6}\right] = 4\epsilon\left[\left(\frac{\sigma_{ij}}{r}\right)^{12} - \left(\frac{\sigma_{ij}}{r}\right)^{6}\right]$,
where $r_{minij} = 2^{1/6}\sigma_{ij}$ is the minimum of the potential (for the structured proteins it is equal to the C_α-C_α distance in the native conformation). The residues that are not in a contact repel each other with a truncated and shifted form of the L-J potential: $V_r(r \leq r_o) = \epsilon\left[\left(\frac{r_{min}}{r}\right)^{12} - 2\left(\frac{r_{min}}{r}\right)^{6} + 1\right]$,
where $r_o = 4$ Å ensures the excluded volume (by default it is set to 5 Å in the variable `cut`). Contacts between the ith and $i+2$nd residues are always described by the repulsive version of the L-J potential (also in potentials for disordered proteins). Please set `cut` to 4 Å when using only the structure-based potential, in order to get results consistent with the literature [6, 12, 18].

4.2. Unstructured parts

The backbone stiffness potential for disordered parts does not favor any secondary structure, so in order to make structures like α-helices or β-sheets possible, we allow for attractive contacts between the ith and $i+3$rd residues in the chains, as these contacts correspond to hydrogen bonds between backbone atoms in the all-atom representation [20]. Forming backbone-backbone contacts results in helix-like and sheet-like structures that are similar to α-helices and β-sheets (see Fig. 3). In this approximate way, our model incorporates secondary structure elements that may be disrupted and reformed, as backbone-backbone contacts keep being broken and reestablished. However the nature of $i, i+4$ contacts is different: the quasi-adiabatic potential works best with $i, i+4$ contacts turned on (the variable `lii4` set to true), and the pseudo-improper-dihedral potential without them (`lii4` false). A more detailed comparison of the model variants is available in [21].

The logical variable `lcpot` controls if our custom potential for disordered proteins is turned on. With `lcpot` false only residue pairs listed in the contact map attract each other. Turning `lcpot` true allows for a custom potential for all the residues, except those present in user-defined contact maps. So if you provide a PDB file and tell the program to construct a Go model contact map for it (by setting `lpdb` true), but at the same time you set `lcpot` true, the pairs of residues listed in the contact map will interact via the ,,simple" L-J potential described above, but those not in the contact map will be subject to a potential customized for intrinsically disordered proteins. There are two versions of this potential: quasi-adiabatic and pseudo-improper-dihedral. Both of them are described below.
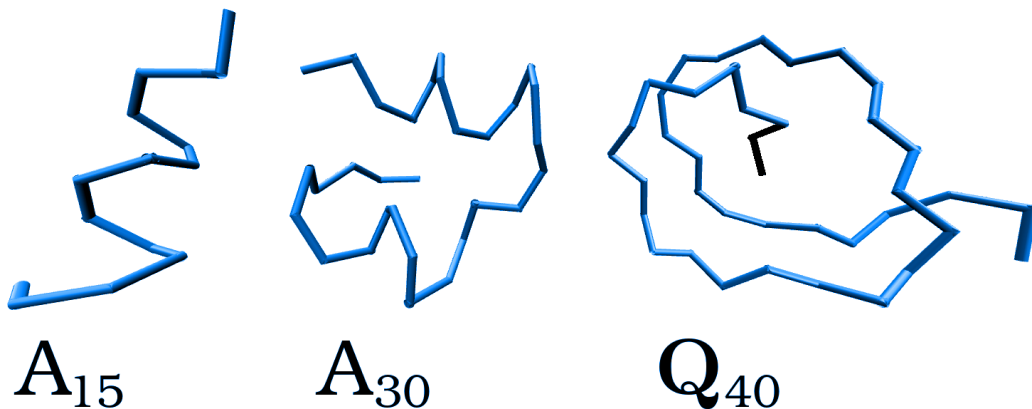
4.2.1. Quasi-adiabatic potential

Figure 3: Three examples of conformations generated by the quasi-adiabatic potential [13].

The default, quasi-adiabatic version is faster, and uses contacts described by the L-J potential with the depth 1 ϵ (just as the Go potential for the structured parts) and a minimum r_{min} (defined below), but here the contacts are formed and broken quasi-dynamically, based on the geometry of the chain. As real amino acids are made from a sidechain and a backbone, in our model we distinguish sidechain-sidechain (ss), backbone-sidechain (bs) and backbone-backbone (bb) contacts (each has different criteria for formation, but only one type of contact can be made between residues at a given time).

After conditions for forming the contact are met, formation of the contact happens quasi-adiabatically: the depth of the L-J potential well increases linearly form 0 to ϵ within 10 τ, which is long enough for the system to thermalize, but short enough to not disrupt dynamics of the system [13]. Breaking the contact also happens quasi-adiabatically, with the same time scale.

There are 3 criteria for forming a contact between residues i and j:

- **Distance**: each type of contact has specific r_{min} associated with it (based on a maximum of a statistical distribution from the PDB [13]). It is 5 Å for bb, 6.8 Å for bs, and a value specific for a given pair of amino acids for an ss contact. Those numbers are specified in the `paramfile`. To make a contact of a given type, the distance $r_{i,j}$ between

the residues must be smaller than $r_{min} \cdot (1+tolerance)$ to make a contact (the default value of the variable `tolerance` is 0).

- **Direction**: the approximate directions of a backbone hydrogen bond or a sidechain C_β atom may be derived based on the position of 3 consecutive C_α atoms [22, 23, 13]. Let's assume that v_i is the vector connecting the ith and $i+1$st beads (centered on the supposed positions of the C_α atoms). Then we can construct two auxiliary vectors:

$$\mathbf{n}_i = \frac{\mathbf{v}_i - \mathbf{v}_{i-1}}{\mathbf{v}_i - \mathbf{v}_{i-1}} \qquad\qquad \mathbf{h}_i = \frac{\mathbf{v}_i \times \mathbf{v}_{i-1}}{|\mathbf{v}_i \times \mathbf{v}_{i-1}|} \qquad (2)$$

They represent the normal ($\mathbf{n}_i$) and binormal ($\mathbf{h}$) vectors of the Frenet frame defined by the $\mathbf{v}_i, \mathbf{v}_{i-1}$ vectors, and can be associated with the direction of the sidechain ($-\mathbf{n}_i$) and with the direction of a hydrogen bond made by an atom from the backbone ($\pm\mathbf{h}_i$). A survey of the contacts from the PDB [22, 23] led to the following criteria for a contact of each type:

- bb: $|cos(\mathbf{h}_i, \mathbf{r}_{i,j})| > 0.92$, $|cos(\mathbf{h}_j, \mathbf{r}_{i,j})| > 0.92$ (the threshold angle is 23°), $|cos(\mathbf{h}_i, \mathbf{h}_j)| > 0.75$ (the treshold angle is 41°)
- bs: $cos(\mathbf{n}_i, \mathbf{r}_{i,j}) < 0.5$, $|cos(\mathbf{h}_j, \mathbf{r}_{j,i})| > 0.92$ (where the ith residue donates its sidechain and the jth residue its backbone)
- ss: $cos(\mathbf{n}_i, \mathbf{r}_{i,j}) < 0.5$, $cos(\mathbf{n}_j, \mathbf{r}_{j,i}) < 0.5$ (the threshold angle is 60°)

- **Coordination number**: the solvent in the program is implicit, so we cannot simulate how polar residues make contacts with water molecules. Using the one-bead-per-residue model also makes the protein less dense than it really is. We try to atone for that by allowing each amino acid to form only a limited number of contacts. Each residue can make only 2 contacts with their backbone (1 for proline) and s ss contacts with their sidechain. Out of these contacts only n_H can be made with hydrophobic residues and n_P with polar residues. s, n_H and n_P depend on the amino acid. Those numbers and the division into polar and hydrophobic residues are available in the `paramfile` and here in the table 1. Those values are based on the PDB statistics [13]. bs contacts are considered to be „polar" (thus $n_H + n_P$ may be bigger than s).

11

name	Gly	Pro	Gln	Cys	Ala	Ser	Val	Thr	Ile	Leu
type	-	-	P	P	H	P	H	P	H	H
s	0	0	2	3	3	2	4	2	5	5
n_H	0	0	0	2	1	0	4	0	4	4
n_P	0	0	2	2	1	2	1	2	2	2
name	Asn	Asp	Lys	Glu	Met	His	Phe	Arg	Tyr	Trp
type	P	P_-	P_+	P_-	H	P	H	P_+	H	H
s	2	2	2	2	4	2	6	2	4	5
n_H	0	0	0	0	1	0	4	0	2	4
n_P	2	2	2	2	1	2	2	2	2	3

Table 1: Coordination numbers of residues. ,,type'' indicates whether a given amino acid is considered polar (P) or hydrophobic (H). The subscripts + or – indicate the charge. s, n_P and n_H are defined in the text.
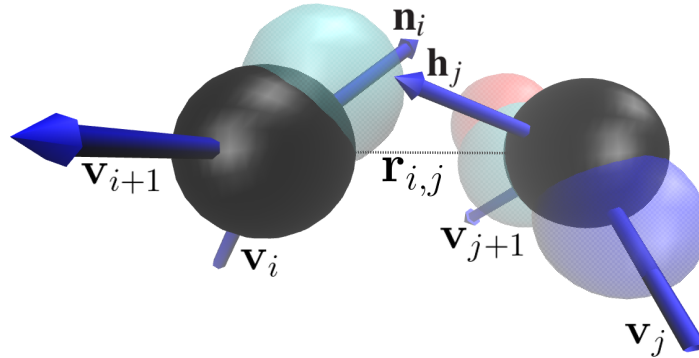
Figure 4: An example of a bs contact between Ala (on the left) and Met. Only sidechain heavy atoms of Ala and only backbone heavy atoms of Met are shown, all in the CPK coloring scheme. The C_α atoms are shown in black. The arrows indicate vectors $\mathbf{v}$, $\mathbf{h}$ and $-\mathbf{n}$. The vector $r_{i,j}$ connecting the C_α atoms is drawn as a dotted line. The value of r_{min} for bs contacts is 6.8 Å.

An example of a bs contact is shown on Fig. 4.

The contact between residues i and j is considered broken when $r_{i,j} >$ `cntfct` $\cdot 2^{-1/6} \cdot r_{min}$, where the default value of the variable `cntfct` is and the $2^{-1/6}$ factor comes from the inflection point of the L-J potential.

Notice that despite complex geometrical criteria required for forming a contact in the quasi-adiabatic potential, the contact itself is described by a spherically-symmetric potential.

It is important to note that the `cntfct` variable is also used in another context, to describe the breaking of a contact in the structure-based Go model. In this case the condition $r_{i,j} > \mathtt{cntfct} \cdot 2^{-1/6} \cdot r_{min}$ does not change anything in the dynamics of the model. It is just used to count how many contacts are present. Using this criterion in this way is arbitrary, but we recommend using `cntfct=1.5` for counting contacts in the Go model, to get the same results as in the literature [6, 12, 18].

4.2.2. Pseudo-improper-dihedral potential

The second custom potential (turned on by setting the variable `lpid` true) is slower, but its hamiltionian is time-independent and it better agrees with experiment [21]. It distinguishes only backbone-backbone (bb) and sidechain-sidechain (ss) contacts. A contact between residues i and j is described by a 6-body potential that involves residues $i-1, i, i+1$ and $j-1, j, j+1$. It is a product of a Lennard-Jones potential and two imroper dihedral potentials, each of them involving only 4 residues. A dihedral angle potential always restricts the angle between two planes, each set by three residues. Such a potential is already used in the model to describe the backbone stiffness (then it acts on 4 consecutive residues). In its „improper" version, this potential acts on three consecutive beads and a sidechain bead connected to the middle one. The „pseudo-improper" version acts on residues that are not even covalently connected. When residues i and j interact, the first „pseudo-improper-dihedral" (PID) angle is defined by the planes spanned by residues $i-1, i, i+1$ and $i-1, j, i+1$. The second PID angle is between the planes $j-1, j, j+1$ and $j-1, i, j+1$.
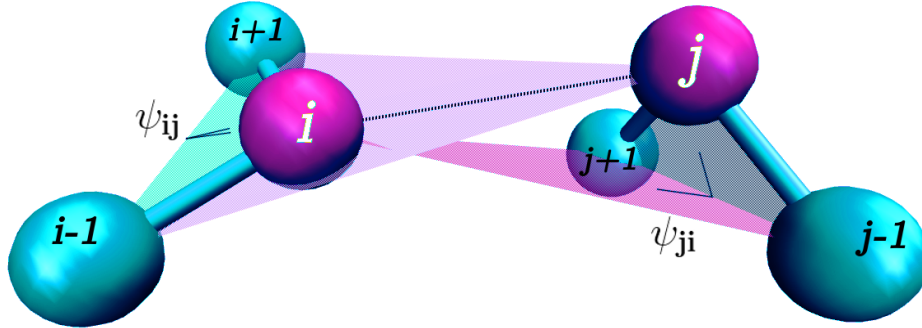

Figure 5: The idea of the PID angles. The interaction between residues i and j involves angles Ψ_{ij} (defined by $i-1, i, i+1$ and j beads) and Ψ_{ji} (defined by $j-1, j, j+1$ and i beads).

Those planes are shown in Fig. 5. The potential for each pair of residues is

described by one distance $r_{i,j}$ and two pseudo-improper-dihedral (PID) angles $\Psi_{i,j}$ and $\Psi_{j,i}$. The potential is a product of three terms: $V(\psi_{i,j}, \psi_{j,i}, r_{i,j}) = \lambda_{i,j}(\psi_{i,j})\lambda_{j,i}(\psi_{j,i})\phi(r_{i,j})$, where $\phi(r_{i,j})$ is a L-J potential. The PID angle distributions show clear peaks for the bb and ss cases. We used the Boltzmann inversion to discover that $\lambda(\psi)$ is indeed a Gaussian-like function [21]. Because the peaks are narrow, instead of a Gaussian function we implemented two functions that vanish away from the peak (i.e. they have a compact support). The first of them is a fragment of a cosine function (used if the variable `lcospid` is true), shifted to the peak value ψ_0 and rescaled by the α parameter that is inversely proportional to the width of the peak:

$$\lambda(\psi) = \begin{cases} 0.5 \cdot \cos\left[\alpha(\psi - \psi_0)\right] + 0.5 & \text{when } -\pi < \alpha(\psi - \psi_0) < \pi \\ 0 & \text{otherwise} \end{cases}$$

The second equation gives a very similar shape of $\lambda(\psi)$, but is made from algebraic sigmoid functions, so it is faster (here $\upsilon = \alpha(\psi - \psi_0)$):

$$0.5 \cdot \cos\upsilon + 0.5 \approx \frac{(\upsilon/\pi)^2 - 2 \cdot |\upsilon/\pi| + 1}{2 \cdot (\upsilon/\pi)^2 - 2 \cdot |\upsilon/\pi| + 1} \tag{3}$$

The distributions of the PID angles for the bb and ss cases are different so $V = V^{ss} + V^{bb}$. For the ss contacts, $V^{ss}(\psi_A, \psi_B, r) = \lambda^{ss}(\psi_A)\lambda^{ss}(\psi_B)\phi^{ss}(r)$. It turns out that the supports of λ^{bb} and λ^{ss} do not overlap. This makes the program faster (it computes only one λ at a time).

The value of r_{min} for the ss cases depends on which pair of amino acids is interacting. All values of r_{min} are listed in the `paramfile` (in the same place as for the quasi-adiabatic potential).

The bb PID angle distribution has two peaks (ψ_0^{bb+} and ψ_0^{bb-}), corresponding to the anti-parallel and parallel β strands, so the bb potential has two corresponding terms:

$$\lambda_{bb}(\psi) = \begin{cases} 0.5 \cdot \cos\left[\alpha^{bb+}(\psi - \psi_0^{bb+})\right] + 0.5 & \text{when } -\pi < \alpha^{bb+}(\psi - \psi_0^{bb+}) < \pi \\ 0.5 \cdot \cos\left[\alpha^{bb-}(\psi - \psi_0^{bb-})\right] + 0.5 & \text{when } -\pi < \alpha^{bb-}(\psi - \psi_0^{bb-}) < \pi \\ 0 & \text{otherwise} \end{cases}$$

These two terms correspond to a left- and right- handed α-helix in the case of $i, i+3$ interactions, so only one of them is used to mimic right-handedness of most of the α-helices in proteins.

14

The repulsive part of the LJ potential should always be present for small distances, to prevent the residues from passing through one another (the excluded volume effect). So even if λ is set to 0, the repulsive potential should not vanish. Therefore:

$$V^{bb}(\psi_A, \psi_B, r) = \begin{cases} \lambda^{bb}(\psi_A)\lambda^{bb}(\psi_B)\phi^{bb}(r) & \text{when } r > r_{min}^{bb} \\ \phi^{bb}(r) + (1 - \lambda^{bb}(\psi_A)\lambda^{bb}(\psi_B))\epsilon^{bb} & \text{otherwise} \end{cases}$$

For the bb case, r_{min} is stored in the variables `rbb1` and `rbb2`(corresponding to the $^+$ and $^-$ cases). The default values are $r_{min}^+ = 5.6$ Å and $r_{min}^- = 6.2$ Å. Other default parameters are: $\alpha^{bb+} = 6.4$, $\alpha^{bb-} = 6.0$, $\alpha^{ss} = 1.2$, $\psi_0^{ss} = -0.23$ rad, $\psi_0^{bb+} = 1.05$ rad, $\psi_0^{bb-} = -1.44$ rad. The fit to the statistical potential is available in ref. [21].

The depth of the L-J potential ϕ does not have to be uniform and always equal to 1 ϵ. For bb contacts, ϵ^{bb} can be set in the variable `epsbb`. By default $\epsilon^{bb} = 1$ ϵ. If the variable `lmj` is set to true, a Miyazawa-Jernigan matrix can be supplied by the `paramfile`. We include the `paramfile`s with the classic Miyazawa-Jernigan matrix based on the PDB statistics from 1996 [24] (`parametersMJ96.txt`), the MDCG matrix derived from all-atom simulations [25] (`parametersMDCG.txt`) and the MDCG matrix multiplied by 0.05 (`parametersMD05.txt`). The best agreement with the experiment for the PID model is accomplished for the MDCG matrix multiplied by a factor between 0 and 0.1 [21]. IDPs are strongly hydrated, which may be the cause of such small values. We include multiplier 0.05, because 0.1 turned out to lead to excessive aggregation of many chains (in ref. [21], the PID model is parameterized only for single chains). In example 2 (attached with the program) we use α-synuclein, a strongly hydrated IDP that was extensively studied [26, 27]. The parameters used in example 2 should be a good starting point for recreating its properties (like the number of contacts), but further parameterization is probably needed.

Long sidechains have significant flexibility, resulting in broad distributions of C_α-C_α distances for ss contacts [13]. To recreate this we introduced a modified form of the L-J potential, shown below (it is turned on by the `lsink` variable):

$$\phi^{ss}(r) = \begin{cases} \epsilon^{ss}\left[\left(\frac{r_{min}^{ss}}{r}\right)^{12} - 2\left(\frac{r_{min}^{ss}}{r}\right)^{6}\right] & \text{when } r > r_{min}^{ss} \\ -\epsilon^{ss} & \text{when } r_{min}^{ss} > r > r_{min}^{bb} \\ \epsilon^{ss}\left[\left(\frac{r_{min}^{bb}}{r}\right)^{12} - 2\left(\frac{r_{min}^{bb}}{r}\right)^{6}\right] & \text{when } r_{min}^{bb} > r \end{cases}$$

4.2.3. Electrostatics

The electrostatic interactions for the quasi-adiabatic and pseudo-improper-dihedral potentials are the same (unless the variable `lepid` is set to true, which is an untested feature). They are described by the Debye-Hueckel (D-H) screened electrostatic potential $V_{D-H}(r_{i,j}) = \frac{q_i q_j \exp(-r_{i,j}/s)}{4\pi\varepsilon\varepsilon_0 r_{i,j}}$, where the screening length s is defined in the variable `screend` and the charges q_i, q_j are 0, +1 or -1, based on the charge defined in the `paramfile` (in our units the elementary charge $e = 1$).

The electric permittivity ε is constant if the variable `lecperm` is true, but if `lecperm` is false, a distance-dependent $\varepsilon = 4$ Å$/r_{i,j}$ is used [28]. In both cases ε (or the coefficient before $r_{i,j}^{-1}$) is included in the amplitude of the D-H interactions, `coul`. Default values of this amplitude are 85 ϵ·Å·Å (for the distance-dependent $\varepsilon = 4$ Å$/r_{i,j}$) and 2.63 ϵ·Å (for the constant $\varepsilon = 80$ case).

The electrostatic interactions between the oppositely-charged residues can be treated differently by setting the variable `lrepcoul` true. Then the attractive interactions between those residues are governed by the custom potential (quasi-adiabatic or pseudo-improper-dihedral). The repulsive interactions between same-charged residues are still treated by the D-H potential.

4.3. Disulfide bonds

For structured proteins, cysteines are treated as every other amino acid, with the exception of those specified in the SSBOND records in the PDB file - those are connected harmonically with the spring constant `H1` and the equilibrium distance 6 Å. If the variable `lsslj` is true, even the cysteines in the SSBOND records are treated as other amino acids.

The options for the disordered case (when `lcpot` is true) work only for the quasi-adiabatic version of the model. The default option is the same harmonic potential as above, but it is quasi-adiabatically turned on when the criteria of an ss contact formation are fulfilled, together with two additional

conditions: cysteines are not part of any other disulfide bond, and the sum of the numbers of their neighbours must be smaller than `neimaxdisul` (9 by default). The number of neighbours is computed as the number of residues within a radius `rnei` (default 7.5 Å) from a given cysteine. This harmonic potential can be quasi-adiabatically turned off if $(r_{ij} - 6$ Å$)^2 > 0.1$ Å^2 and the total number of neighbours is smaller than `neimaxdisul`. Thus, the disulfide bond formation in disordered proteins can be turned off by specifying `neimaxdisul`$= -1$. Those neighbour calculations are supposed to mimic the effect of solvent accessibility, which is crucial for the disulfide bond oxidation and reduction [29].

If the variable `lsselj` is true, a special L-J potential with the depth `dislj` (4 ϵ by default) is used. When this special type of contact is quasi-adiabatically turned on between a pair of cysteines, they can no longer make such pairs with other cysteines. This contact obeys the same criteria for formation and breaking as any other ss contact.

5. Simulation protocols, geometries and boundary conditions

All the variables that describe time in the sections below are given in τ units. Every type of simulation can be repeated multiple times with different random seed during one execution of the program. Each repeat is called a trajectory. The variable `ntraj` determines the number of repeats. Out of all residues, the first `nen1` of them can be „frozen" - they will not move during the simulation (which may greatly reduce computation time).

5.1. Simulation protocols

5.1.1. Equilibrium simulations

With the default options the proteins are put in an infinite space and there are no external forces acting upon them (except for friction and the thermostat). Time of the whole simulation is `mstep`. All the other protocols are preceded by the equilibration period. Boundary conditions may then be different, but the duration of the equilibration is always specified by the variable `kteql`. However the simulation time never exceeds `mstep`.

5.1.2. Stretching by AFM tips

If the `lvelo` variable is true, the first and last residues are caught by harmonic springs with the spring constant `HH1` and pulled apart in the direction determined by the vector connecting both residues. To perform stretching

17

simulations with spring constant resembling real AFM tips, please set HH1 to
0.06 $\epsilon/\text{Å}^2$ in the inputfile [6].

The springs start to act after the equilibration and this vector is also
determined at the time $t =$ kteql. Then, the springs are moved apart from
each other along this vector with the velocity velo (in $\text{Å}/\tau$ units). The
variable lforce has a similar effect, but the pulling occurs with a constant
force (specified in the variable coef, in $\epsilon/\text{Å}$ units) acting on the terminal
residues, instead of the constant speed.

5.1.3. Folding simulations

When the variables lconftm or lmedian are true, the simulation starts
from a straight line or a self-avoiding random walk (which is controlled by
the variable lsawconftm). Then, for lconftm, the average time of contact
formation is computed for each native contact. The average is taken over all
trajectories.

For lmedian, the program computes the median folding time. The folding
time is defined as the time where all the native contacts are established [12].
If more than half of the trajectories did not end in the folded state, the
simulation ends (because then it is impossible to compute the median folding
time). Trajectories during folding simulations may end after mstep timesteps,
or after the native conformation is reached (if the variable lnatend is set to
true).

5.1.4. Unfolding simulations

When the variable lunfold is true, the simulation starts from the native
structure and the average time of contact breaking is computed for each na-
tive contact. The average is taken over all trajectories. Those simulations
are usually performed in the temperature temp higher than the room tem-
perature. If the variable lthermo is true, thermodynamic properties such
as the heat capacity $C_V = \frac{<E^2>-<E>^2}{T^2}$ are computed. A set of different
temperatures can be simulated by specifying different tstart and tend.

5.1.5. Deformations of the simulation box

When the variable lwall is true, the simulation does not occur in infinite
space but in a box. Currently all the simulation protocols supported in this
case require a specified density (in residue/Å^3 units), so the variable ldens
must be set to true. In the case of proteins from a PDB file, the box sizes
in the CRYST1 record are initially used. For disordered proteins, the initial

density is `sdens` and the simulation box is a cube. The chains (straight lines or self-avoiding walks, as set in `lsawconftm`) are constructed randomly in the box. If a chain goes beyond the box, the box size is increased to fit it (unless there are periodic boundary conditions in the initial state, which is controlled by the `lcpb` variable). It means that the actual density during the start of the dynamics may be even lower than `sdens`. Then, the box shrinks uniformly in all the 3 dimensions (the wall speed is `densvelo`) until it reaches the target density `tdens`. Then the box size does not change for a time equal to `ktrest`.

If the variable `loscillate` is true, the box starts to deform again (after the resting period specified in `ktrest`). There are two possible modes of oscillating the box walls: pulling and shearing (determined by the `lshear` variable). In the pulling mode, the distance between the two walls perpendicular to the Z axis changes periodically. In the shearing mode, the same two walls move periodically along the X axis.

The oscillation starts by achieving the maximal amplitude of oscillation. If the variable `lampstrict` is true, the amplitude `ampstrict` is the wall displacement (in the Z direction for pulling, in the X direction for shearing) in Å. Otherwise, the amplitude `amp` is determined relative to the initial distance between the two moving walls. After the walls are moved to their maximum displacement, there is another resting period (with a duration time `ktrest`) for equilibration. Then the displacement s changes periodically: $s(t) = s_0(1 - A\cos 2\pi(t/period))$, where the period of the oscillations is the variable `period` and the amplitude A is either `amp`$\cdot s_0$ or `ampstrict`. The number of full oscillations is specified by `kbperiod`. If the variable `lconstvol` is true, the wall distance in the X and Y directions is also changed to maintain the constant volume of the box. After oscillations (or after equilibration if `loscillate` is false), the system is equilibrated again (with the resting time `ktrest`) and then the walls in the Z direction extend to $2s_0$ (if the variable `lpullfin` is true, otherwise the box does not deform anymore).

5.2. Boundary conditions

As mentioned before, by default the proteins are in an infinite space. Setting `lwall` true makes the simulation box finite. Then, by default the two walls perpendicular to the Z axis are ,,solid". Setting `lpbc` true results in periodic boundary conditions only in the X and Y dimensions. To make them in all the three dimensions, the variable `lnowal` must be also set to true.

The variable `lwalls` is used to get ,,solid" walls in the X and Y directions. Here ,,solid" means repulsive, as decribed by the potential $V_{wall} = \frac{-\epsilon}{9d^9}$, where d is the distance from a residue to the wall [30].

The walls in the Z direction may have different ways of interacting with the residues. However, initially they always interact via V_{wall} and there is a specific time when other special types of interaction may be turned on for the two walls along the Z axis. This specific time is controlled by the variable `kconnecttime`. Here are its possible values:

`kconnecttime`	description
1	just after reaching target density `tdens`
3	after waiting for `ktrest` after reaching `tdens`
5	just after reaching the maximal amplitude of oscillations
7	just after oscillations
9	never

Table 2: Meaning of the kconnecttime variable

Below are listed all the tested ways to introduce interactions between the residues and the two walls that perpendicular to the Z direction.

5.2.1. FCC walls

If the variable `lfcc` is true, each of the walls is made from two layers of beads arranged in a face-centered-cubic lattice cut through the [111] crystallographic direction. The lattice constant is `af`. The beads are immovable (except for the movement of the whole wall) and interact via the L-J potential with the same amplitude as for disulfide bonds, `dislj`, and the same minimum, `walmindist`. The interactions are quasi-adiabatically turned on when the residue comes closer than `walmindist` (the minimum of the L-J potential) to one of the beads and is broken in the same way when the distance $r_{i,\text{bead}} > \texttt{cntfct} \cdot 2^{-1/6} \cdot \texttt{walmindist}$.

5.2.2. Flat attractive walls

If `ljwal` is true, the residues interact with the wall via the same quasi-adiabatic L-J potential as above. The only difference is that when the residue comes closer than `walmindist` to the wall, it starts to be attracted by an artificial bead with the same X and Y coordinates as the residue's coordinates when it crosses the `walmindist` distance. This artificial bead does not attract other residues and disappears when $r_{i,\text{bead}} > \texttt{cntfct} \cdot 2^{-1/6} \cdot \texttt{walmindist}$. This

20

can give a significant computational speed-up, as those artificial beads are not checked when updating the Verlet list.

If the variables `ljwal` and `lfcc` are false, all the residues closer than `walmindist` to the wall are harmonically attached to it, with the spring constant `HH1` and the equilibrium distance equal to the distance in the moment of turning on the interaction. The default value of `HH1` is 30 $\epsilon/\text{Å}^2$, which corresponds to strong, covalent bonding.

6. Advantages and disadvantages of the program

The quasi-adiabatic and pseudo-improper-dihedral potentials are specifically designed for simulating intrinsically disordered proteins. The first potential generalizes the idea of the contact interactions taken from the structure-based Go model [6] (which is also included in the program). This enables the simulation of partially disordered proteins, like the structured domains connected by flexible linkers (e.g. cellulosome proteins [31]), proteins with disordered tracts (e.g. huntingtin [32]) and large protein complexes with different amounts of complexity (e.g. gluten [33]). Our program was used to simulate the last case, gluten, which (according to our knowledge) was never simulated before [34]. The program was also used to study aggregation of polyglutamine, which is a fully disordered protein [35]. These two applications (both involving simulations of thousands of residues for more than a milisecond) prove the suitability of the quasi-adiabatic potential for large protein systems. One disadvantage of the program is that (in the present form) it is not yet prepared to study folding-upon-binding mechanisms [36].

The main strength of the program is allowing the study of large protein systems (in relatively long timescales), while retaining their protein nature. For the largest systems (like gluten) the faster quasi-adiabatic potential is best-suited [13]. Smaller systems can be studied with better accuracy by the slower pseudo-improper-dihedral potential [21]. The program may be also used to study fully structured proteins [1, 3, 6, 8, 9, 10, 12].

There are many other packages that may be used for implicit solvent, coarse-grained protein molecular dynamics [37, 38, 39, 40, 41, 42, 43, 44], but none of them implements the quasi-adiabatic and pseudo-improper-dihedral potentials presented here. Many of them allow user-defined force fields [37, 38, 42, 40, 44] and/or are open-source [39, 40, 41], however, none of them allows for an easy addition of dynamic contacts, switched on and off based on criteria of distance, direction and coordination number (see section

4.2.1). We discuss the advantages of using those criteria in a one-bead-per-residue model elsewhere [13]. The novel pseudo-improper-dihedral potential (described in section 4.2.2 and in [21]) also cannot be easily added to the existing packages (with the possible exception of [40] and [41]). The main disadvantage of using our software is that in the current state it is not as scalable as the alternatives (it cannot be effectively run on GPUs and it does not support MPI, only openMP).

Acknowledgments

*Electronic address: mc@ifpan.edu.pl

References

[1] M. Cieplak and T.X. Hoang, Universality classes in folding times of proteins. Biophys J. **84**(1) (2003) 475-488.

[2] T. Veitshans, D. Klimov and D. Thirumalai, Protein folding kinetics: time scales, pathways and energy landscapes in terms of sequence-dependent properties, Folding Des. **2** (1997) 1-22.

[3] P. Szymczak and M. Cieplak, Stretching of proteins in a uniform flow. J. Chem. Phys. **125**(16) (2006) 164903.

[4] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, New York, 1987.

[5] J. M. Haile, *A Primer on the Computer Simulation od Atomic Fluids by Molecular Dynamics* Clemson University, Clemson, NC (1980).

[6] M. Sikora, J. I. Sułkowska and M. Cieplak, Mechanical strength of 17 134 model proteins and cysteine slipknots. PLoS Comp. Biol. **5** (2009) e1000547.

[7] A. B. Poma, M. Chwastyk and M. Cieplak, Polysaccharide-protein complexes in a coarse-grained model. J. Phys. Chem. B. **119** (2015) 12028-12041.

[8] M. Cieplak and M. O. Robbins, Nanoindentation of virus capsids in a molecular model. J. Chem. Phys. **132** (2010) 015101.

[9] K. Wołek and M. Cieplak, Self-assembly of model proteins into virus capsids. J. Phys. Cond. Matter **47** (2017) 474003.

[10] T. X. Hoang and M. Cieplak, Molecular dynamics of folding of secondary structures in Go-like models of proteins, J. Chem. Phys. **112** (2000) 6851-6862.

[11] L. M. Pietrek, L.S. Stelzl and G. Hummer, Hierarchical ensembles of intrinsically disordered proteins at atomic resolution in molecular dynamics simulations, J. Chem. Theory Comput. **16**(1) (2020) 725-737.

[12] K. Wołek and M. Cieplak, Criteria for folding in structure-based models of proteins. J. Chem. Phys. **144** (2016) 185102.

[13] Ł. Mioduszewski and M. Cieplak, Disordered peptide chains in an α-c-based coarse-grained model. Phys. Chem. Chem. Phys. **20** (2018) 19057-19070.

[14] A. Ghavami, E. V. D. Giessen and P. R. Onck, Coarse-grained potentials for local interactions in unfolded proteins. J. Chem. Theory Comput. **9** (2013) 432–440.

[15] B. Schuler, E. A. Lipman, P. J. Steinbach, M. Kumke, and W. A. Eaton, Polyproline and the ,,spectroscopic ruler" revisited with single-molecule fluorescence. Proc. Natl. Acad. Sci. USA **102** (2005) 2754–2759.

[16] G. Settanni, T. X. Hoang, C. Micheletti and A. Maritan (2002) Folding pathways of prion and doppel. Biophys. J. **83** (2002) 3533-3541.

[17] J. Tsai, R. Taylor, C. Chothia and M. Gerstein, The packing density in proteins: Standard radii and volumes. J. Mol. Biol. **290** (1999) 253-266.

[18] J. I. Sułkowska and M. Cieplak, Mechanical stretching of proteins – A theoretical survey of the Protein Data Bank, topical review, J. Phys.: Cond. Mat. **19** (2007) 283201.

[19] K. Wołek, Á. Gómez-Sicilia and M. Cieplak, Determination of contact maps in proteins: a combination of structural and chemical approaches, J. Chem. Phys. **143** (2015) 243105.

[20] A. Kolinski, Protein modeling and structure prediction with a reduced representation. Acta Biochim. Pol. **51** (2004) 349-371.

[21] Ł. Mioduszewski, B. Różycki and M. Cieplak, Pseudo-improper-dihedral model for intrinsically disordered proteins, J. Chem. Theory Comput. **16**(7) (2020) 4726–4733.

[22] M. Enciso and A. Rey, A refined hydrogen bond potential for flexible protein models. J. Chem. Phys. **132** (2010) 235102.

[23] N. B. Hung, D.-M. Le and T. X. Hoang, Sequence dependent aggregation of peptides and fibril formation. J. Chem. Phys. **147** (2017) 105102.

[24] S. Miyazawa and R. L. Jernigan, Residue – Residue Potentials with a Favorable Contact Pair Term and an Unfavorable High Packing Density Term, for Simulation and Threading, J. Mol. Biol. **256** (1996) 623-644.

[25] M. R. Betancourt and S. J. Omovie, Pairwise energies for polypeptide coarse-grained models derived from atomic force fields, J. Chem. Phys. **130** (2009) 195103.

[26] P. Robustelli, S. Piana and D. E. Shaw, Developing a molecular dynamics force field for both folded and disordered protein states, Proc. Natl. Acad. Sci. USA **115**(21) (2018) 4758-4766.

[27] M. Chwastyk, M. Cieplak, Conformational Biases of α-Synuclein and Formation of Transient Knots. J. Phys. Chem. B. **124**(1) (2020) 11-19.

[28] V. Tozzini, J. Trylska, C.E. Chang and J.A. McCammon, Flap opening dynamics in hiv-1 protease explored with a coarse-grained model, J. Struct. Biol. **157** (2007) 606-615.

[29] M. Qin, W. Wang and D. Thirumalai, Protein Folding Guides Disulfide Bond Formation. Proc. Natl. Acad. Sci. USA **112** (2015) 11241-11246.

[30] M. Cieplak and M. O. Robbins, Nanoindentation of 35 virus capsids in a molecular model: Relating mechanical properties to structure, PLoS ONE **8** (2013) e63640.

[31] B. Różycki, P.-A. Cazade, S. O'Mahony, D. Thompson, M. Cieplak, The length but not the sequence of peptide linker modules exerts the primary influence on the conformations of protein domains in cellulosome multi-enzyme complexes. Phys. Chem. Chem. Phys. **19** (2017) 21414-21425.

[32] F. Giorgini, Polyglutamine provides huntingtin flexibility, Proc. Natl. Acad. Sci. USA **110**(36) (2013) 14516-14517.

[33] H. Wieser. Chemistry of gluten proteins. Food Microbiol. 2007, **24**(2), 115–119.

[34] Ł. Mioduszewski and M. Cieplak, Viscoelastic properties of wheat gluten in a molecular dynamics study, doi: 10.1101/2020.07.29.226928 (2020, in review).

[35] Ł. Mioduszewski and M. Cieplak, Protein droplets in systems of disordered homopeptides and the amyloid glass phase, Phys. Chem. Chem. Phys. **22** (2020) 15592–15599.

[36] P. Robustelli, S. Piana and D. E. Shaw, Mechanism of Coupled Folding-upon-Binding of an Intrinsically Disordered Protein, J. Am. Chem. Soc. **142**(25) (2020) 11092–11101.

[37] M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess and E. Lindahl, GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers, SoftwareX, **1–2** (2015) 19-25.

[38] S. Plimpton, Fast Parallel Algorithms for Short-Range Molecular Dynamics, J. Comp. Phys. **117** (1995) 1-19.

[39] A. Liwo, S. Oldziej, C. Czaplewski, D. S. Kleinerman, P. Blood and H. A. Scheraga, Implementation of molecular dynamics and its extensions with the coarse-grained UNRES force field on massively parallel systems; towards millisecond-scale simulations of protein structure, dynamics, and thermodynamics, J. Chem. Theory Comput. **6** (2010) 890-909.

[40] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, R. P. Wiewiora, B. R. Brooks, V. S. Pande, OpenMM 7: Rapid Development of High Performance Algorithms for Molecular Dynamics, PLoS Comput. Biol. **13** (2017) e1005659.

[41] H. Kenzaki, N. Koga, N. Hori, R. Kanada, W. F. Li, K. Okazaki, X. Q. Yao and S. Takada, CafeMol: A Coarse-Grained Biomolecular Simulator for Simulating Proteins at Work, J. Chem. Theory Comput. **7**(6) (2011) 1979-1989.

[42] J. A. Anderson, J. Glaser, and S. C. Glotzer, HOOMD-blue: A Python package for high-performance molecular dynamics and hard particle Monte Carlo simulations, Comput. Mater. Sci. **173** (2020) 109363.

[43] I. A. Solov'yov, A. V. Yakubovich, P. V. Nikolaev, I. Volkovets, and A. V. Solov'yov, Meso Bio Nano Explorer - a universal program for multiscale computer simulations of complex molecular structure and dynamics, J. Comput. Chem. **33** (2012) 2412-2439.

[44] K. J. Bowers, E. Chow, H. Xu, R. O. Dror, M. P. Eastwood, B. A. Gregersen, J. L. Klepeis, I. Kolossvary, M. A. Moraes, F. D. Sacerdoti, J. K. Salmon, Y. Shan and D. E. Shaw, Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters, Proc. ACM/IEEE Conf. Supercomput. **SC06** (2006).