CHAPTER FOUR

Screening for cysteine-stabilized scaffolds for developing proteolytic-resistant AMPs

Mariana Rocha Maximiano[a,b,†], Samilla Beatriz Rezende[a,†], Thuanny Borba Rios[a,b], Michel Lopes Leite[c], Liana Costa Pereira Vilas Boas[b,d], Nicolau Brito da Cunha[b], Állan da Silva Pires[b], Marlon Henrique Cardoso[a,b], and Octávio Luiz Franco[a,b,d,*]

[a]S-Inova Biotech, Pós-Graduação em Biotecnologia, Universidade Católica Dom Bosco, Campo Grande, Brazil
[b]Centro de Análises Proteômicas e Bioquímicas, Programa de Pós-Graduação em Ciências Genômicas e Biotecnologia, Universidade Católica de Brasília, Brasília, Brazil
[c]Departamento de Biologia Molecular, Instituto de Ciências Biológicas, Universidade de Brasília, Brasília, Brazil
[d]Programa de Pós-Graduação em Patologia Molecular, Universidade de Brasília, Brasília, Brazil
[*]Corresponding author: e-mail address: ocfranco@gmail.com

Contents

Abstract

Antimicrobial peptides (AMP) are present in all organisms and can present several activities and potential applications in human and animal health. Screening these molecules scaffolds represents a key point for discovering and developing novel biotechnological products, including antimicrobial, antiviral and anticancer drugs candidates and

[†] These authors contributed equally.

Methods in Enzymology, Volume 663
ISSN 0076-6879
https://doi.org/10.1016/bs.mie.2021.11.001

insecticidal molecules with potential applications in agriculture. Therefore, considering the amount of biological data currently deposited on public databases, computational approaches have been commonly used to predicted and identify novel cysteine-rich peptides scaffolds with known or unknown biological properties. Here, we describe a step-by-step *in silico* screening for cysteine-rich peptides employing molecular modeling (with a core focus on comparative modeling) and atomistic molecular dynamics simulations. Moreover, we also present the concept of additional tools aiming at the computer-aided screening of new Cs-AMPs based drug candidates. After the computational screening and peptide chemical synthesis, we also provide the reader with a step-by-step *in vitro* activity evaluation of these candidates, including antibacterial, antifungal, and antiviral assays.

1. Introduction

Antimicrobial peptide (AMP) are molecules that present up to 10 kDa and has been found in all organisms (Huan, Kong, Mou, & Yi, 2020; Nijnik & Hancock, 2009). Cysteine-stabilized peptides (Cs-AMPs) correspond to a group that presents conserved cysteine motifs (Table 1) and a broad variation in sequence and numerous structural profiles (β-sheets, α-helical, or a mixture of these two secondary structures) (Porto, Pires, & Franco, 2017). These peptides can present various activities (de Oliveira Dias & Franco, 2015; Srivastava et al., 2021), acting as promising model molecules for the generation of future peptide based-drug candidates with and potential applications in medicine, veterinary medicine, and agriculture, including the development of novel antimicrobial and antiviral properties

Table 1 Conserved cysteine motifs present in antimicrobial peptides families.

Peptide family	Conserved cysteine motifs	Reference
Lipid transporter proteins (LTPs)	$Cx_{(7–9)}Cx_{(12–14)}Cx_{(8–19)}Cx_{(1)}Cx_{(19–23)}Cx_{(13–15)}C$	Shelenkov, Slavokhotova, & Odintsova, 2020
Snakins	$CX_{(3)}CX_{(3)}CX_{(8)}CX_{(3)}CX_{(2)}CCX_{(2)}CX_{(1)}CX_{(11)}CX_{(1)}CX_{(12)}C$	
Heveins	$CX_{(3,8)}CX_{(4)}CCX_{(5)}\ CX_{(6)}CX_{(3,5)}CX_{(1,3)}C$	
Thionin	$CCX_{(11)}CX_{(9,15)}CX_{(5)}CX_{(6,11)}C$	
Cyclotides	$CX_{(3)}CX_{(4,5)}CX_{(4,6)}CX_{(1)}CX_{(4,5)}C$	
Cis-defensins	$CX_{(10)}CX_{(5)}CX_{(3)}CX_{(9)}CX_{(6)}CX_{(1)}\ CX_{(3)}C$	

(Boas, Campos, Berlanda, de Carvalho Neves, & Franco, 2019; Fleitas Martínez, Cardoso, Ribeiro, & Franco, 2019; Maximiano & Franco, 2021).

In this context, computational approaches are increasingly used to study peptide/protein functions from their primary sequence to their tridimensional (3D) structural arrangement, which can be investigated through *in vitro* and *in silico* techniques. *In silico* methodologies have assisted researchers to predict peptide/protein atomic coordinates in the absence of experimentally elucidated structures, which are commonly characterized by X-ray crystallography, solution nuclear magnetic resonance (NMR) or solid-state NMR, and cryo-electron microscopy (Cardoso, Oshiro, Rezende, Candido, & Franco, 2018).

Thus, comparative molecular modeling comprises a useful computational tool for 3D peptides' structure prediction (Fiser & Šali, 2003). Specifically, comparative molecular modeling uses one or more peptide/proteins of known structure (templates) to predict the 3D structure of a given peptide/protein sequence (target) based on sequence alignments (Webb & Sali, 2016). Currently, there are numerous web servers for automating comparative peptide/protein modeling (Table 2).

Table 2 Web servers are used to predict the 3D structures for peptides and proteins.
Web servers—comparative modeling

Name	Link	References
3D-JIGSAW server	https://bmm.crick.ac.uk/~svc-bmm-djigsaw/help_crick.htmL	Bates, Kelley, MacCallum, and Sternberg (2001)
HHpred	https://toolkit.tuebingen.mpg.de/tools/hhpred	Söding, Biegert, and Lupas (2005)
IntFOLD	http://www.reading.ac.uk/bioinf/IntFOLD/IntFOLD6_form.html	McGuffin et al. (2019)
I-TASSER	https://zhanggroup.org/I-TASSER/	Roy, Kucukural, and Zhang (2010)
RaptorX	http://raptorx.uchicago.edu/	Xu, Mcpartlon, and Li (2021)
Robetta	http://robetta.bakerlab.org/	Song et al. (2013)
SWISS-MODEL	https://www.expasy.org/resources/swiss-model	Schwede, Kopp, Guex, and Peitsch (2003)

In this scenario, this work highlights the strategies to screen for Cs-AMPs structure, aiming at developing peptide-based drug candidates, with potential biotechnological applications in several areas, as medicine, veterinary medicine, and agriculture.

2. Materials and equipment

2.1 *In silico*

Minimal recommended configuration:

- Core I5 4GB RAM 500GB HD. Linux Ubuntu 20.04 LTS
- GROMACS 5.1.4
- MODELLER 10.1

2.2 *In vitro*

2.2.1 Antimicrobial (antibacterial and antifungal) assay

- Mueller–Hinton broth;
- Mueller–Hinton agar;
- Potato dextrose agar;
- RPMI 1640 medium;
- Spectrophotometer/microplate reader;
- Sterile 96-well microplates;
- Microcentrifuge tubes, 1.5 mL;
- Falcon tubes;
- Shaker;
- Sterile Petri plates.

2.2.2 Antiviral assay

- CO_2 incubator at 37 °C for cell growth
- Flask for culturing cells (e.g., glass flat bottom 60 mL);
- Inverted microscope
- 6, 24 and 96 wells plates
- DMEM—Dulbecco's Modified Eagle Medium
- Penicillin/Streptomycin solution for cell culture
- Fetal Bovine Serum for cell culture
- Sodium bicarbonate powder
- Carboxymethyl cellulose (CMC);
- Formaldehyde
- Crystal violet powder

3. Step-by-step method details

3.1 *In silico*

This topic presents how regular expressions (RegEx) can be applied for Cs-AMPs screening in public databases and repositories, including proteomes, genomes, and transcriptomes. Additionally, we present the concept of molecular modeling and how comparative modeling, with a core focus on MODELLER, has been used to predict the 3D structure of known and unknown Cs-AMPs sequences. Finally, we highlight how atomistic molecular dynamics simulations have profoundly contributed to calculating the trajectory of Cs-AMPs in molecular systems mimicking a desired biological condition.

3.1.1 Searching for Cs-AMPs in public databases

3.1.1.1 Preparing the computational environment and databases

(a) Perl or Python is necessary to perform the searches proposed here. An updated repository for installation on the Linux environment is available in the link: (https://www.cpan.org/src/5.0/perl-5.28.1.tar.gz, for Perl; and https://www.python.org/downloads/ for Python)

(b) A target database can be chosen based on specific preferences. For demonstrative purposes, here we will use the *Arabidopsis thaliana* proteome. The database is available on the Uniprot Consortium (https://www.uniprot.org/proteomes/UP000006548)

(c) There are several Cs-AMPs from plants (e.g., defensins, lipid transfer proteins (LPTs) and snakins), and the pipeline could be performed for most of them. However, this methodology can not be performed for families with a poor amount of sequences described such as β-barrelin. A RegEx that represents the peptide family is required to screen for Cs-AMP sequences within a target proteome/genome/transcriptome. Many of these families already have one or more RegEx described in the literature (Costa et al., 2020; Silverstein et al., 2007; Tomczak et al., 2012; Zhu, 2008). For this pipeline, the CSαβ-defensin RegEx: $CX_{2-18}CX_3CX_{2-10}$[GAPSIDERYW]XCX_{4-17}CXC (each amino acid is represented by one-letter code; "X" represents any proteinogenic amino acid, and brackets indicate positions that allows one of those amino acids between brackets), described by Zhu (2008), will be used as an example. A commented Python script is available for data mining (Supplementary File 1 in the online version

at https://doi.org/10.1016/bs.mie.2021.11.001). The packages *re* (Van Rossum, 2020) and Bio (Cock et al., 2009) are necessary for the proper performance of the script

3.1.2 Comparative modeling of Cs-AMPs by using MODELLER

3.1.2.1 Installing MODELLER

(a) MODELLER should be installed to proceed with the analyses. The software is available for Mac operating system (OS), Windows and Unix/Linux (https://salilab.org/modeller/download_installation.html).

(b) The files required to complete the basic protocol can be found on http://salilab.org/modeller/tutorial/basic-example.tar.gz (Unix/Linux) or http://salilab.org/modeller/tutorial/basic-example.zip (Windows).

(c) Here, the most recent version of MODELLER (v10.1) was used. The algorithms applied were tested for MODELLER v9.17 and superior.

3.1.2.2 Searching for suitable template structures

(a) Here, we will use as an example the following target Cs-AMP sequence:
>PDB ID: 1ti5
R T **C** M I R R E G W G R **C** L I D T T **C** A H S **C** K N K G Y I G G N **C** K G M T K T **C** Y **C** L V N **C**

Where cysteine residues are in bold type

(b) The template structures can be searched using different algorithms, including BLASTp (https://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM= blastp) and HHpred (https://toolkit.tuebingen.mpg.de/tools/hhpred). BLAST (Basic Local Alignment Tool) uses local sequence alignments to find proteins with high sequence similarity. By contrast, HHpred applies a secondary structure alignment to find similar proteins. In both cases, a target sequence is necessary as input

(c) In this process, alignment statistics such as sequences' identity, e-value, and sequence coverage are essential to select a template structure. Therefore, the identity values should be ~30% or higher for closely related Cs-AMP sequences. Moreover, the e-value defines the statistical measure of the alignment, representing the chances of random alignments

(d) All these parameters detect similarities between the target sequence and database template structures

(e) Once the template is selected, the .pdb file (the atomic coordinates of a structurally determined peptide) can be obtained from Protein Data Bank (https://www.rcsb.org/) (Fig. 1).

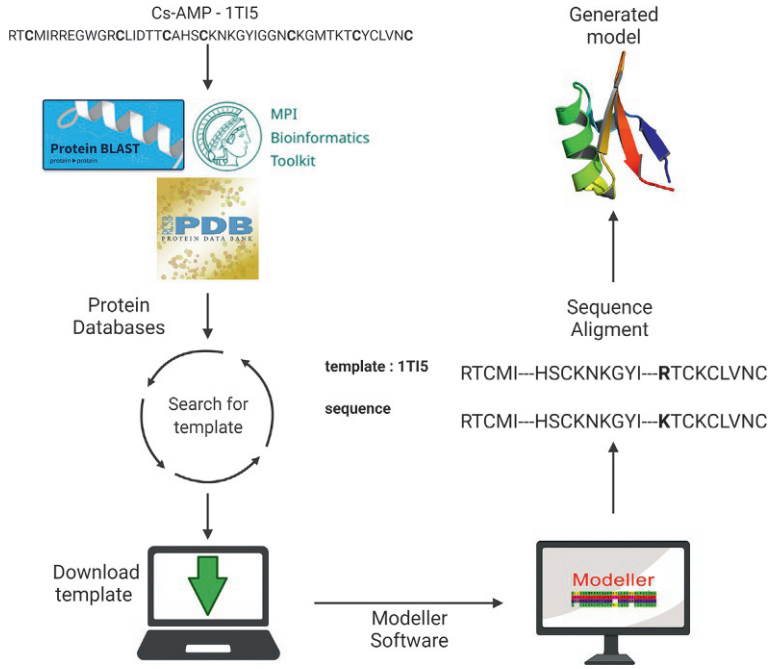

Fig. 1 Scheme demonstrating the steps to download .pdb files for a given peptide/protein structure from databases (Protein Data Bank Server, Blastp and HHpred). These experimentally determined structures are then used for comparative modeling. After downloading the template .pdb structure using a four-letter code (e.g., 1ti5), it is submitted to software MODELLER, based on sequence alignment simulations the model would be generated. Created with the support of Biorender (BioRender.com).

(f) In addition, MODELLER also provides a script for template searching. The script is named build_profile.py (https://salilab.org/modeller/tutorial/basic.html).

3.1.2.3 Checking necessary scripts to perform sequence alignment and generating 3D theoretical models

(a) TvLDH.ali—this PIR format file is necessary for MODELLER to proceed and write alignments. In this script, the target sequence is inserted in FASTA format. The first line consists of (>P1;code), followed by the sequence's identifier. After populating the sequence target, insert (*), which marks its end (Fig. 2)

(b) Align2d.py—this file is a script used to align the template (.pdb) sequence with the target sequence described in the TvLDH.ali script

```
>P1;TvLDH
sequence:target sequence:::::::0.00: 0.00
RTCMIRREGWGRCLIDTTCAHSCKNKGYIGGNCKGMTKTCYCLVNC*
```

Fig. 2 Representation of the TvLDH.ali script, which contains the target sequence in PIR format.

(c) model-single.py—this script generates the theoretical models based on the template atomic coordinates and spatial orientation. However, for Cs-AMPs, if the cysteines are not aligned in align2d.py output, another script should be used to specify each disulfide bond, as explained below

(d) model-disulfide.py—this script is a punctual script to generate models that contain disulfide bonds not covered by the template. The appropriate disulfide bond restraints are generated for the output model/models

3.1.2.4 Aligning TvLDH.ali with the selected template

(a) Insert the target sequence according to the instructions above in the TvLDH.ali file

(b) With the template (e.g., 1ti5.pdb) structure in the same folder, open the file align2d.py. This script will align the target sequence in the TvLDH.ali file with the template structure described in the .pdb file

(c) Once the align2d.py script is opened, the first line will indicate the template structure (FILE) 1ti5.pdb. It is necessary to change the parameters in (ALIGN_CODES) to "1ti5", followed by (ALN.APPEND), where the variable (FILE) receives "TvLDH.ali", which contains the target sequence. Next, the sixth line will execute the command ALIGN2D to perform the alignment (Fig. 3).

```
from modeller import *

env = environ()
aln = alignment(env)
mdl = model(env, file='1ti5', model_segment=('FIRST:A','LAST:A'))
aln.append_model(mdl, align_codes='1ti5', atom_files='1ti5.pdb')
aln.append(file='TvLDH.ali', align_codes='TvLDH')
aln.align2d()
aln.write(file='TvLDH-1ti5.ali', alignment_format='PIR')
aln.write(file='TvLDH-1ti5.pap', alignment_format='PAP')
```

Fig. 3 Script align2d.py is used to align the target sequence with its selected template.

3.1.2.5 Command lines to execute the alignment

(a) Execute through the command line in Linux (Ubuntu), using the following command: mod10.1 (the number 10.1 depends on the installed version).

(b) To do the alignment, use the command mod10.1 align2d.py

Two output files will be produced: a PIR (TvLDH-1ti5.ali) and a PAP (TvLDH-1ti5.pap) format file. MODELLER uses the PIR format for the model-building stage, whereas the PAP alignment format is more intuitive to check the alignment results. Identical amino acid residues are underscored with an asterisk (Fig. 4A and B).

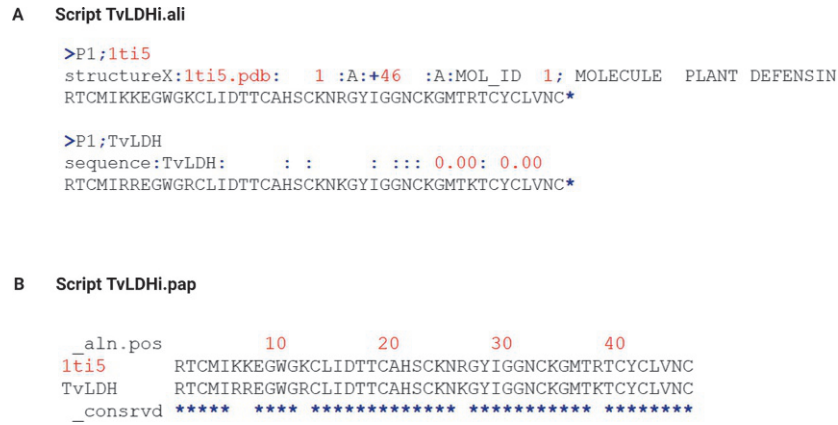

Fig. 4 (A) TvLDH-1ti5.ali, illustrating the alignment between the template and target sequences. (B) TvLDH-1ti5.pap, highlighting by "*" the conserved residues between the two sequences. Created with the support of Biorender (BioRender.com).

3.1.2.6 Generating theoretical models using model-single.py

(a) After the alignment has been executed by the script align2d.py, open the file model-single.py, which calculates 3D models for the target sequence based on the template structure. The first line of this script sets the environment. The following lines comprise parameter values for the "model" routine. The variable (ALNFILE) sets the name of the alignment file "TvLDH-1ti5.al," the parameter (KNOWNS) defines the known template structure and (SEQUENCE) needs to be assigned as TvLDH, which has the target sequence. At last, the parameters (A.STARTING_MODEL) and (A.ENDING_MODEL) define the number of models that will be calculated. The last line refers to the command to run the "model" routine (Fig. 5A).

A **Script model-single.py**

```
from modeller import *
from modeller.automodel import *

env = environ()
a = automodel(env, alnfile='TvLDH-1ti5.ali',
              knowns='1ti5', sequence='TvLDH',
              assess_methods=(assess.DOPE, assess.GA341))
a.starting_model = 1
a.ending_model = 100
a.make()
```

B **Script model-disulfide.py**

```
# Comparative modeling by the automodel class
from modeller import *              # Load standard Modeller classes
from modeller.automodel import *    # Load the automodel class

# Redefine the special_patches routine to include the additional disulfides
# (this routine is empty by default):
class MyModel(automodel):
    def special_patches(self, aln):
        # A disulfide between residues 3 and 46:
        self.patch(residue_type='DISU', residues=(self.residues['3'],
                                                  self.residues['46']))
        # A disulfide between residues 12 and 32:
        self.patch(residue_type='DISU', residues=(self.residues['12'],
                                                  self.residues['32']))

log.verbose()
env = environ()


env.io.atom_files_directory = ['.', '../atom_files']

a = MyModel(env,
            alnfile='TvLDH-1ti5.ali',
            knowns='1ti5', sequence = 'TvLDH',
            assess_methods=(assess.DOPE))
a.starting_model= 1
a.ending_model  = 100

a.make()
```

Fig. 5 Representation of scripts used to run molecular modeling simulations (A) representing model-single.py script used to generate 3D theoretical models for the target sequence based on the template structure. (B) model-disulfide.py script, used to generate models that contain disulfide bonds in their structure. Created with the support of Biorender (BioRender.com).

(b) When modeling a sequence with disulfide bonds, it may be necessary to use a modified version of model-single.py, named model-disulfide.py (https://salilab.org/modeller/manual/node24.html#SECTION:model-disulfide). The restraints to disulfide bonds are added in the script. To generate CHARMM topology it is used the file DISU in the parameter (RESIDUE_TYPE), the residues with the disulfide bonds are inserted in (SELF.RESIDUES[“]) parameter. The instruction should be repeated according to the number of disulfide bonds with each specific pair of cysteines (i.e., cys3-cys46 and cys12-cys32) (Fig. 5B).

3.1.2.7 Using model-disulfide.py used to generate 3D theoretical

(a) Command lines to generate 3D theoretical models with unpaired disulfide bonds

- **i.** To generate models without disulfide bonds restrains, run mod10.1 model-single.py
- **ii.** To generate models containing disulfide bonds without coverage in template structure, run mod10.1 model-disulfide.py
- **iii.** Choose the lowest free-energy generated model based on the DOPE score. The DOPE (Discrete Optimized Protein Energy) is added into MODELLER, based on a refined reference state that coincides with non-interacting atoms in a uniform sphere with a sample native structure radius dependent. It was tested with several non-redundant sets of crystallographic structures. Therefore, it is among the best parameters to calculate peptides/proteins free-energy (Shen & Sali, 2006).

(b) The report of the output files can be found in "model-single.log" and "model-disulfide.log," depending on the scripts used. Helpful information about generated models, including report warnings, errors, and input restraints, can be found in .log files. The model can be visualized using the PyMOL software (https://pymol.org/2/).

3.1.2.8 Structural statistics for a given 3D theoretical model

MODELLER evaluates final models using the scoring function DOPE (Shen & Sali, 2006), developed independently based on a conformation analysis of well-folded peptides and proteins. After the selection of the lowest free-energy model, an additional evaluation step is recommended. For that, many servers can be used. General examples include the ProSA-web server (https://prosa.services.came.sbg.ac.at/prosa.php) and QMEAN (https://swissmodel.expasy.org/qmean/) to evaluate the 3D theoretical model fold in comparison with other peptides of similar size and structurally determined by X-ray crystallography or NMR. Additionally, PROCHECK (https://saves.mbi.ucla.edu/) can be used to check the stereochemistry quality of models. High-quality models are expected to present an NMR or X-ray similar z-score on ProSA, a modular z-score below to 2 on QMEAN and, at least, 90% of the residues in most favored and additionally allowed regions on the Ramachandran Plot.

3.1.3 Molecular dynamics

3.1.3.1 Installing GROMACS

(a) The first step is to download and install GROMACS software at (http://www.gromacs.org/Downloads), distributed for Mac OS, Windows and Unix/Linux system but works natively on a Unix-type system (such as Linux, or Mac OS X).

(b) The installation can be done according to the GROMACS version that the user chose. There are two ways to build GROMACS. We will use a user-friendly installation. For this, the configuration described below is necessary:

(i) Get the latest version of C and C++ compilers;

(ii) Check if the computer has CMake installed (it varies according to GROMACS version);

(iii) Download and unpack the latest version of GROMACS;

(iv) Select a separate build directory and change to it;

(v) Run the command cmake with a path to the source as an argument;

(vi) Run commands make, make check and make install (Fig. 6).

(c) Here, an MD simulation demonstration will be performed using GROMACS version 5.0.4 on Ubuntu 16.04 LTS. For more information about system configuration and performance, please see: (https://manual.gromacs.org/current/user-guide/mdrun-performance.html).

```
tar xfz gromacs-5.0.7.tar.gz
cd gromacs-5.0.7
mkdir build
cd build
cmake .. -DGMX_BUILD_OWN_FFTW=ON -DREGRESSIONTEST_DOWNLOAD=ON
make
make check
sudo make install
source /usr/local/gromacs/bin/GMXRC
```

Fig. 6 Commands to run when executing the quick and dirty installation of GROMACS.

3.1.3.2 Programing the MD simulation

PDB structure

(a) Go to Protein Data Bank and download (https://www.rcsb.org/) a target structure or use your theoretical molecular model generated by comparative modeling. Here cysteine-rich defensin VrD1 (pdb: 1tl5) was used as an example

Checking scripts

(a) Check the necessary scripts that will be used throughout the MD parametrization. The scripts are:
ions.mdp—to generate ions counting to parametrize the system;
minim.mdp—system energy minimization;
nvt.mdp—temperature parametrization;
npt.mdp—pressure parametrization;
md.mdp—to generate the file that will execute md.tpr.

(b) Open the terminal command on Linux/Ubuntu, then source GROMACS using the following command line:
source/usr/local/gromacs/bin/GMXRC.

3.1.3.3 Start programming your MD simulation

(a) First, it is necessary to generate GROMACS coordinate files (.gro) and topology using as input the .pdb file downloaded from PDB or obtained from molecular modeling. The pdb2gmx tool generates a GROMACS atomic coordinate (.gro) from the input .pdb file. To map a disulfide bond, use the flag: -ss (they will be automatically identified). The other parameters refer to the solvent type (e.g., water: flag -water spce), net-charge (neutral; protonation: flag: -inter), N- and C-terminus modifications (flag: -ter), and polar hydrogen addition or removal (flag: -ignh). For more information, access (https://manual.gromacs.org/archive/5.0.4/programs/gmx-pdb2gmx.html).

(b) Type the following command:
gmx pdb2gmx -f 1ti5.pdb -o 1ti5.gro -ignh -inter -ter -water spce -ss

(c) Next, a list of force fields will pop up (Fig. 7). Select the force field that suits best the kind of macromolecules and system you want to simulate. The GROMOS96 43a force field is commonly used for linear and cyclic peptides and, therefore, it was selected for the methodology described here. Select option 9 for GROMOS96 43a

(d) Next, it will be necessary to protonate the peptide/protein. Register how many residues will be protonated. This data will be used in the following steps

(e) The following step includes the simulation box generation. The command (flag: -editconf) converts generic structure from .gro, (flag: -o) to .pdb. To set the geometric center of the system, use (flag: -c), to center the system in the box, use (flag: -d) followed by the measure of the

```
Select the Force Field:
From '/usr/local/gromacs/share/gromacs/top':
 1: AMBER03 protein, nucleic AMBER94 (Duan et al., J. Comp. Chem. 24, 1999-2012, 2003)
 2: AMBER94 force field (Cornell et al., JACS 117, 5179-5197, 1995)
 3: AMBER96 protein, nucleic AMBER94 (Kollman et al., Acc. Chem. Res. 29, 461-469, 1996)
 4: AMBER99 protein, nucleic AMBER94 (Wang et al., J. Comp. Chem. 21, 1049-1074, 2000)
 5: AMBER99SB protein, nucleic AMBER94 (Hornak et al., Proteins 65, 712-725, 2006)
 6: AMBER99SB-ILDN protein, nucleic AMBER94 (Lindorff-Larsen et al., Proteins 78, 1950-58, 2010)
 7: AMBERGS force field (Garcia & Sanbonmatsu, PNAS 99, 2782-2787, 2002)
 8: CHARMM27 all-atom force field (CHARM22 plus CMAP for proteins)
 9: GROMOS96 43a1 force field
10: GROMOS96 43a2 force field (improved alkane dihedrals)
11: GROMOS96 45a3 force field (Schuler JCC 2001 22 1205)
12: GROMOS96 53a5 force field (JCC 2004 vol 25 pag 1656)
13: GROMOS96 53a6 force field (JCC 2004 vol 25 pag 1656)
14: GROMOS96 54a7 force field (Eur. Biophys. J. (2011), 40,, 843-856, DOI: 10.1007/s00249-011-0700-9)
15: OPLS-AA/L all-atom force field (2001 aminoacid dihedrals)
```

Fig. 7 List of force filed that can be chosen depending on the user's needs.

distance from solute and the box (e.g., 1.0). To define the box shape (e.g., cubic, dodecahedral, octahedral, among others), use the command (flag: -bt). For more information, access (https://manual.gromacs.org/archive/5.0.4/programs/gmx-editconf.html).

(f) The final command should be something as described below:

gmx editiconf -f 1ti5.gro -o box.gro -c -d 1.0 -bt cubic

(g) After preparing the simulation box, proceed to the system's solvation. For this, use the flag: -solvate, followed by the flag: -cp to specify the output files from the previous step (editconf). The flag: -cs is used to specify the solvent, which will be the Simple Point Charge water (SPC) using the flag: -spc216.gro. Nevertheless, other co-solvents can also be added to the system, including 2,2,2-trifluoroethanol and methanol

(h) The final command should be something as described below:

gmx solvate -cp box.gro -cs spc216.gro -o solv.gro -p topol.top

(i) The topology file (topol.top) will be updated constantly from this step, as this file contains information about molecule types and the number of molecules

(j) An additional input file with the molecular dynamics parameter file extension (.mdp) will be used to produce a .tpr file with grompp. Grompp assembles specified parameters from the .mdp file with the coordinates and topology information to generate a .tpr file. Type the following command:

gmx grompp -f ions.mdp -c solv.gro -p topol.top -o ions.tpr

(k) With the atomic-level description of the system in the binary file ions.tpr, use the genion command. To add the necessary ions to neutralize the system's net charge by adding the correct number of negative ions (Na+) or positive ions (Cl-), use the flag: -neutral. Moreover, if a given ionic strength is desired, for instance, 0.15 M NaCl, use the flag: -conc 0.15

(l) For more information, please visit (https://manual.gromacs.org/archive/5.0.4/programs/gmx-genion.html).

The final command should be something as described below:

gmx genion -s ions ions.tpr -o solv_ions.gro -p topol.top -neutral or

gmx genion -s ions.tpr -o solv_ions.gro -p topol.top -neutral -conc 0.15

For each ion that is added to the system, one solvent molecule will be removed.

3.1.3.4 MD energy minimization

(a) Energy minimization steps are used to ensure no steric clashes or inappropriate geometry in the system. A process called energy minimization

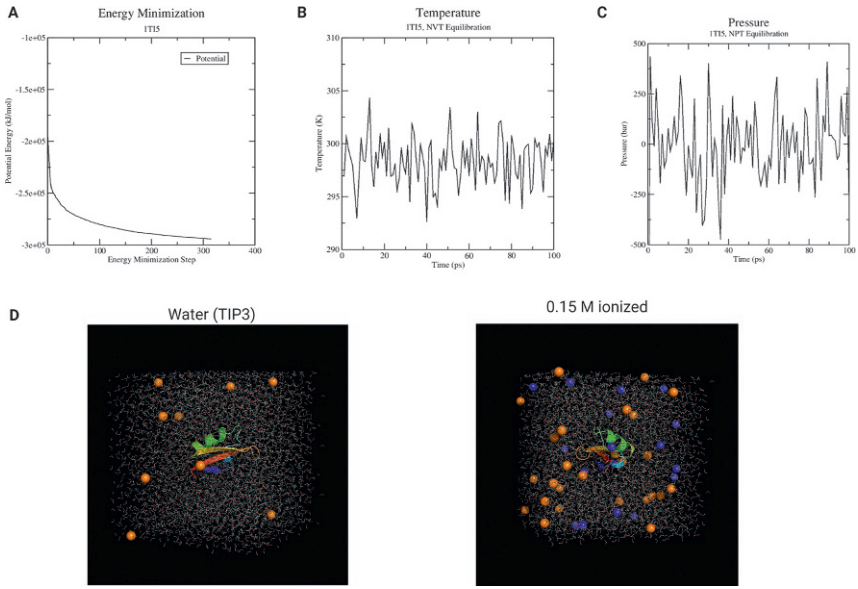

Fig. 8 (A) The plot is represented by an energy potential that results from energy minimization, demonstrating that energy is converging. (B) Plot generated showing the temperature progression of the system according to the number of steps set in the nvt.mdp script. The graph was built using xmgrace. (C) Plot generated using xmgrace shows the system's pressure equilibration, which is usually set at 1 bar. And (D) cubic boxes containing the target cysteine-rich peptide, the TIP3 water molecules and one system only with the necessary ions to neutralize the system's charge (left); whereas the other system (right) shows the same system, however at 0.15 M ionic strength. Both systems had their energies minimized, as described above. Created with the support of Biorender (BioRender.com).

is applied for structure relaxing (Fig. 8A and D). For more information visit: (https://manual.gromacs.org/archive/5.0.4/programs/gmx-grompp.html) and (https://manual.gromacs.org/archive/5.0.4/programs/gmx-mdrun.html).

(b) Run the following command:

gmx grompp -f minim.mdp -c solv_ions.gro -p topol.top -o em.tpr.
gmx mdrun -v -deffnm em

(c) The potential energy graph can be plotted by the following command:

gmx energy -f em.edr -o potential.xvg.

Select the option: Potential.

(d) Then open the plot using xmgrace software. For more information: (https://plasma-gate.weizmann.ac.il/Grace/).

3.1.3.5 MD equilibration

(a) Equilibration is commonly conducted in two steps. The first is conducted under an NVT ensemble (constant Number of particles, Volume, and Temperature). This ensemble can also be referred as "isothermal-isochoric" or "canonical" (Fig. 8B). The system's temperature should reach a plateau at the desired value. For more information, see: (https://manual.gromacs.org/documentation/2018/user-guide/mdp-options.html).

(b) To run temperature equilibration using the nvt.md script type the following commands:

```
gmx grompp -f nvt.mdp -c em.gro -p topol.top -o nvt.tpr.
gmx mdrun -v -deffnm nvt
```

(c) To plot the temperature graph, run the following command:

```
gmx energy -f nvt.edr -o temperature.xvg
```

Select the Temperature option and open the. xvg output file using xmgrace

(d) Pressure equilibration is the second step conducted under an NPT ensemble, as the number of particles, pressure, and temperature are constant. The ensemble is also called the "isothermal-isobaric" ensemble and closely resembles experimental conditions (Fig. 8C). For more information, please visit (https://manual.gromacs.org/documentation/2018/user-guide/mdp-options.html).

(e) To generate NPT files run the following commands:

```
gmx grompp -f npt.mdp -c nvt.gro -t nvt.cpt -p topol.top -o npt.tpr
gmx mdrun -v -deffnm npt
```

(f) To plot the pressure graph, run the following command:

```
gmx energy -f npt.edr -o pressure.xvg
```

Select the Pressure option and open the. Xvg output file using xmgrace.

3.1.3.6 MD simulation

(a) After minimization and equilibration steps, the next step is to release the position restraints and run the MD simulation. It is a similar process to the commands described above for NVT and NPT. For this representative simulation, we will run 10 ns of MD simulation. Therefore, it is necessary to edit script md.mdp and alter the value in nsteps parameter (Fig. 9).

(b) Then execute the following commands:

```
gmx grompp -f md.mdp -c npt.gro -t npt.cpt -p topol.top -o md.tpr.
gmx mdrun -v -deffnm md.tpr
```