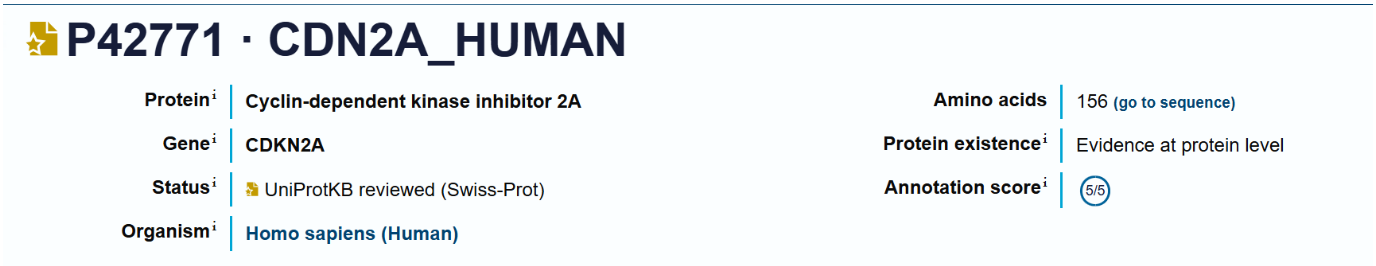
p16_gromacs_pipeline

Tutorial 1



P42771 · CDN2A_HUMAN

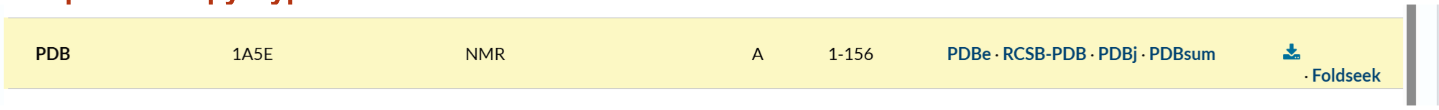
Protein i	**Cyclin-dependent kinase inhibitor 2A**	**Amino acids**	156 **(go to sequence)**
Gene i	**CDKN2A**	**Protein existence** i	Evidence at protein level
Status i	UniProtKB reviewed (Swiss-Prot)	**Annotation score** i	5/5
Organism i	**Homo sapiens (Human)**		

https://www.uniprot.org/uniprotkb/P42771/entry

Protein: p16^INK4a
Gene: CDKN2A
Chromosome: 9p21
Type: tumor suppressor
Function: inhibits CDK4 and CDK6
Effect: stops the cell cycle in G1
Family: INK4
Significance in oncology: often mutated or lost in cancer

p16^INK4a is a tumor suppressor protein that slows down the cell cycle.
It blocks CDK4 and CDK6, thereby preventing the cell from transitioning from G1 to the S phase.
It functions as a natural “brake” on cell division.
When it is lost or mutated, cells begin to divide uncontrollably → often leading to cancer.

Выбранная структура:

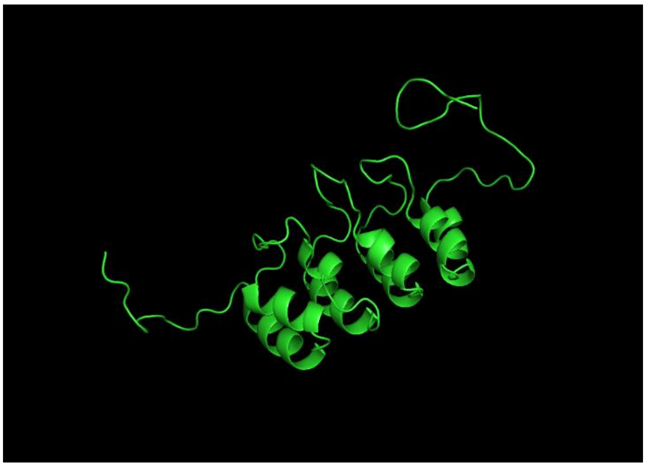


PDB	1A5E	NMR	A	1-156	PDBe · RCSB-PDB · PDBj · PDBsum	· Foldseek

*NMR = nuclear magnetic resonance; a method for determining the structure of a protein in solution

Rename pdb1a5e.ent → p16.pdb

** In An NMR PDB File There Are Usually 10–30 Models (In pdb1a5e.ent There Are 18 Structures Inside), GROMACS By Default Takes ONLY The First Model (MODEL 1).*



Standard GROMACS Pipeline:

- force field: **AMBER99SB-ILDN** *("gold standard" roughly 2010–2020, but still works excellently)*
- water: **TIP3P**

1) Topology Generation (pdb2gmx)

```
gmx pdb2gmx -f p16.pdb -o p16_processed.gro -p topol.top -water tip3p -ignh
```

-f p16.pdb — *input structure*
-o p16_processed.gro — *output structure after processing*
-p topol.top — *generate topology*
-water tip3p — *use TIP3P water*
-ignh — *remove all hydrogens from the PDB and rebuild them correctly*

When prompted for force field in the menu select: **6** (AMBER99SB-ILDN)

Output files:
p16_processed.gro — *final structure*
topol.top — *topology*
posre.itp — *position restraints*

Creation of the Box (editconf)

```
gmx editconf -f p16_processed.gro -o p16_box.gro -c -d 1.0 -bt cubic
```

-f p16_processed.gro — *input structure*
-o p16_box.gro — *structure in the box*
-c — *center the protein*
-d 1.0 — *1.0 nm distance to the box wall*
-bt cubic — *cubic box (simple and clear)*

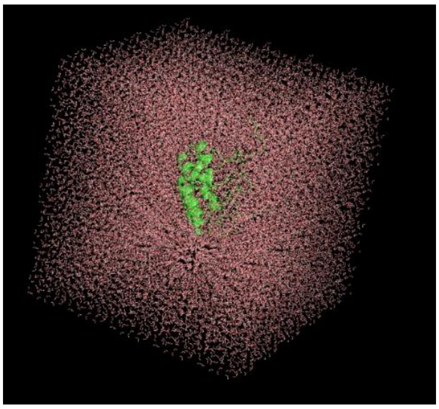
Output file:
p16_box.gro — *structure in a cubic box*

2) Добавление воды (solvate)

```
gmx solvate -cp p16_box.gro -cs spc216.gro -o p16_solv.gro -p topol.top
```

-cp p16_box.gro — *structure in the box*
-cs spc216.gro — *water coordinates (these are SPC parameters, but GROMACS will still apply TIP3P, since tip3p.itp is specified in the topology)*
-o p16_solv.gro — *output system with water*
-p topol.top — *update topology (number of water molecules)*

Output file:
p16_solv.gro — *system filled with water*
(You can open and inspect it in PyMOL, VMD, or any other convenient program)



4) Preparation for adding ions

4.1 Create ions.tpr (grompp)

First, you need to create a simple file `ions.mdp`.

```
integrator   = steep
nsteps       = 500
cutoff-scheme = Verlet
coulombtype  = PME
rcoulomb     = 1.0
rvdw         = 1.0
```

The file must be located in the same folder from which you run the command:

Now the command:

```
 gmx grompp -f ions.mdp -c p16_solv.gro -p topol.top -o ions.tpr -maxwarn 1
```

-f ions.mdp — *minimal MDP file for ionization preparation*
-c p16_solv.gro — *system after solvation*
-p topol.top — *topology (already includes SOL water)*
-o ions.tpr — *output tpr file for genion*
-maxwarn 1 — *allows one warning (about charge) so grompp doesn't stop*

Output file:
ions.tpr — *prepared input file for adding ions (genion). It contains the system with water, but still without ions.*

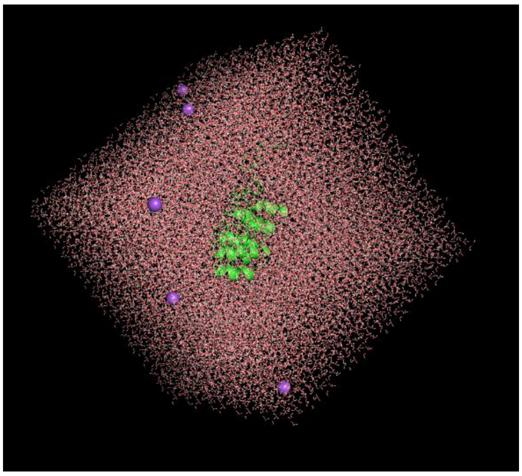
4.2 Add Ions (genion)

```
 gmx genion -s ions.tpr -o p16_solv_ions.gro -p topol.top -neutral -pname NA -
nname CL
```

-s ions.tpr — *input system*
-o p16_solv_ions.gro — *system after replacing water with ions*
-p topol.top — *update topology*
-neutral — *make the system electrically neutral*
-pname NA — *Na^+ cations*
-nname CL — *Cl^- anions⁻*

During execution of the genion command: `"Select a group: SOL"` → Select SOL (the number is usually around 13–15).

After running genion, the following final file will appear:
p16_solv_ions.gro — *system in which part of the water molecules has been replaced with ions (Na^+/Cl^-). This is already a complete system with ions.*



5) Energy Minimization (EM)

Minimization is needed to remove bad contacts after solvation and ion addition.

5.1 Create the em.mdp file

Create the `em.mdp` file in the working directory:

```
integrator  = steep
emtol       = 1000.0
nsteps      = 50000
cutoff-scheme = Verlet
nstlist     = 20
coulombtype = PME
rcoulomb    = 1.0
rvdw        = 1.0
constraints = h-bonds
```

5.2 Prepare the .tpr (grompp)

```
 gmx grompp -f em.mdp -c p16_solv_ions.gro -p topol.top -o em.tpr
```

-f em.mdp — *parameters for minimization*
-c p16_solv_ions.gro — *system with water and ions*
-p topol.top — *topology*
-o em.tpr — *prepared tpr file for mdrun*

Output file: **em.tpr**

5.3 Run Minimization (mdrun)

```
 gmx mdrun -deffnm em
```

-deffnm em — *set the base name for all output files*

Output files:
em.gro — *structure after minimization*
em.edr — *energy file*
em.log — *minimization log*
em.trr — *minimization trajectory*

6) NVT — Temperature Equilibration

NVT fixes the temperature (usually 300 K), but not yet the pressure.
It is a gentle heating phase of the system after EM.

6.1 Create the `nvt.mdp` file

```
integrator    = md
nsteps        = 50000        ; 100 ps при dt=0.002
dt            = 0.002
; Temperature
tcoupl        = V-rescale
tc-grps       = Protein Non-Protein
tau_t         = 0.1     0.1
ref_t         = 300     300
; Pressure
pcoupl        = no
cutoff-scheme = Verlet
nstlist       = 20
coulombtype   = PME
rcoulomb      = 1.0
rvdw          = 1.0
constraints   = h-bonds
; Velocity generation
gen-vel       = yes
gen-temp      = 300
gen-seed      = -1
```

6.2 Prepare the tpr

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

-f nvt.mdp — *parameters*
-c em.gro — *structure after minimization*
-r em.gro — *hard reference positions (restraints)*
-p topol.top — *topology*
-o nvt.tpr — *output tpr*

Output file: **nvt.tpr**

6.3 Run NVT

```
 gmx mdrun -deffnm nvt -v
```

-deffnm nvt — *set the base name for all NVT output files*
-v — *shows real-time progress: steps, energy, ETA*

Output files:
nvt.gro — *structure after NVT*
nvt.edr — *energy file*
nvt.log — *log file*
nvt.cpt — *checkpoint file*

How to understand that everything is fine?
Command to check the temperature:

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

Select **`Temperature`**.
The temperature should stabilize around **300 K**.

7) NPT — Pressure Equilibration

7.1 Create the npt.mdp *file*

(usually almost the same as nvt.mdp, but with a barostat):

```
integrator      = md
nsteps          = 50000        ; 100 ps
dt              = 0.002
; Temperature
tcoupl          = V-rescale
tc-grps         = Protein Non-Protein
tau_t           = 0.1     0.1
ref_t           = 300     300
; Pressure
pcoupl          = Parrinello-Rahman
pcoupltype      = isotropic
tau_p           = 2.0
ref_p           = 1.0
compressibility = 4.5e-5
cutoff-scheme   = Verlet
nstlist         = 20
coulombtype     = PME
rcoulomb        = 1.0
rvdw            = 1.0
constraints     = h-bonds
continuation    = yes
```

7.2 Prepare the tpr file:

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

-f npt.mdp — *file with integration and barostat parameters for the NPT stage*
-c nvt.gro — *structure after NVT equilibration (input geometry)*
-r nvt.gro — *reference positions for position restraints (keeps the protein fixed)*
-p topol.top — *system topology (protein + water + ions)*
-o npt.tpr — *final input .tpr for running NPT*

Output file: **npt.tpr**

7.3 Run NPT:

```
 gmx mdrun -deffnm npt -v
```

-deffnm npt — *base name for all output files*
-v — *show real-time progress (energy, step, ETA)*

Output files:
npt.gro — *structure after the NPT stage*
npt.edr — *energy file (pressure, temperature, volume)*
npt.log — *calculation log*
npt.trr — *trajectory of coordinates/velocities*
npt.cpt — *checkpoint file for continuation*

8) Productive Run (MD)

8.1 Create the MDP file for production (200 ns)

Create `md_200ns.mdp`

```
; 200 ns production MD
integrator              = md
dt                      = 0.002        ; 2 fs
nsteps                  = 100000000    ; 200 ns (200000 ps)
continuation            = yes
constraints             = h-bonds
constraint_algorithm    = lincs
lincs_iter              = 1
lincs_order             = 4
; output
nstxout                 = 10000
nstvout                 = 10000
nstenergy               = 1000
nstlog                  = 1000
nstxout-compressed      = 1000
compressed-x-grps       = System
; Neighbor lists and cutoffs
cutoff-scheme           = Verlet
nstlist                  = 10
rlist                   = 1.0
rcoulomb                = 1.0
rvdw                    = 1.0
coulombtype             = PME
pme-order               = 4
fourierspacing          = 0.12
; Thermostat
tcoupl                  = V-rescale
tc-grps                 = Protein   Water_and_ions
tau_t                   = 0.1       0.1
ref_t                   = 300       300
; Barostat
pcoupl                  = Parrinello-Rahman
pcoupltype              = isotropic
tau_p                   = 2.0
ref_p                   = 1.0
compressibility         = 4.5e-5
; Do not generate velocities, take them from NPT
gen_vel                 = no
```

8.2 Prepare the tpr file for production

```
gmx grompp -f md_200ns.mdp -c npt.gro -t npt.cpt -p topol.top -o md_200ns.tpr
```

-f md_200ns.mdp — *parameters for production MD (200 ns)*
-c npt.gro — *structure after NPT stabilization (input geometry)*
-t npt.cpt — *checkpoint after NPT (take final velocities and barostat state)*
-p topol.top — *topology of the whole system (protein, water, ions)*
-o md_200ns.tpr — *final input file for mdrun*

Output file:
md_200ns.tpr — *ready .tpr file for running long MD*

8.3 Запуск 200 нс симуляции

```
gmx mdrun -deffnm md_200ns -v
```

-deffnm md_200ns — *base name for all long MD output files*
-v — *print real-time simulation progress (energy, ETA)*

Output files:
md_200ns.gro — *final structure after 200 ns*
md_200ns.edr — *energy data over the whole trajectory*
md_200ns.log — *MD log (temperature, pressure, steps)*
md_200ns.trr — *uncompressed trajectory (full coordinates/velocities)*
md_200ns.xtc — *compressed coordinate trajectory (main file for analysis)*
md_200ns.cpt — *checkpoint to continue MD at any time*

*If the **MD stopped/was interrupted**, you **do NOT need** to run grompp again — GROMACS will automatically take everything from the checkpoint:*

Command to continue the simulation:

```
 gmx mdrun -deffnm md_200ns -v -cpi md_200ns.cpt
```

-deffnm md_200ns — *Name of the MD file set (md_200ns.tpr, md_200ns.cpt, md_200ns.edr ...)*
-cpi md_200ns.cpt — *Read the checkpoint and continue exactly from the step where it stopped*

!!! You can also run MD in this **extended form** if you need to explicitly control computational resources:

```
CUDA_VISIBLE_DEVICES=1 gmx mdrun -deffnm md_200ns -nt 32 -pin on -v
```

CUDA_VISIBLE_DEVICES=1 — *specify which GPU to use (numbering starts from 0; here the second GPU is used)*
-nt 32 — *number of CPU threads allocated for the calculation*
-pin on — *pin GROMACS threads to CPU cores (reduces context switching, improves stability and performance)*

9) Centering and PBC Correction

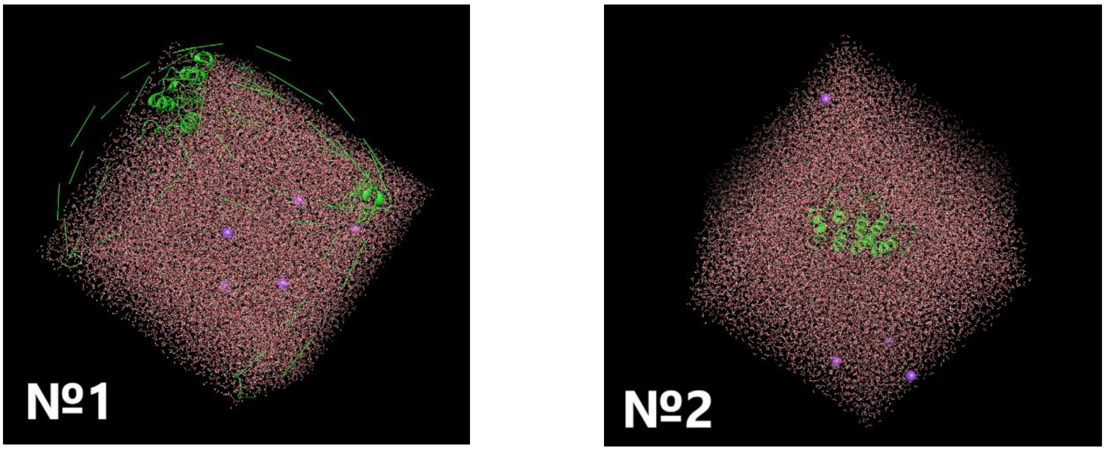
PBC correction is needed so that the protein does not appear "broken" due to periodic boundaries, and centering is required to remove its drift across the box and analyze only real structural changes rather than motion of the entire system.

```
gmx trjconv \
 -s md_200ns.tpr \
 -f md_200ns.xtc \
 -o md_200ns_fit.xtc \
 -pbc mol \
 -center \
 -ur compact
```

Group selection::
1 → Protein (for centering)
2 → System (as the output group)

-s md_200ns.tpr — *file with topology and system information*
-f md_200ns.xtc — *original MD trajectory*
-o md_200ns_fit.xtc — *output trajectory after processing*
-pbc mol — *restore integrity of molecules broken by periodic boundaries*
-center — *center the system (usually the protein) in the box*
-ur compact — *make the unit cell compact (convenient for visualization)*

Output file:
md_200ns_fit.xtc — *trajectory without breaks and with the protein centered*



Example of the last frame from the non-centered (No. 1) and centered (No. 2) trajectories (PBC correction).)

Command to extract the last frame:

```
gmx trjconv -f md_200ns_fit.xtc -s md_200ns.tpr -o last_fit.pdb -dump 200000
```

-f md_200ns_fit.xtc — *input trajectory (already centered/PBC-corrected)*
-s md_200ns.tpr — *tpr file with topology and simulation time*
-o last_fit.pdb — *output PDB file with the selected frame*
-dump 200000 — *extract the frame at 200000 ps (200 ns), i.e., the last frame of the trajectory*