DGX Spark GROMACS MD Simulation Setup & Usage Guide

1. Environment Architecture

Environment	Name	Location	Purpose	Command
Pre-processing	**gaff-gen**	Host Terminal	Ligand Opt, Format Conversion	`micromamba activate gaff-gen`
Simulation	**GROMACS**	Docker Container	System Setup, MD Run	`docker exec -it gmx_session bash`

2. Installation Guide

2.1. Setup Pre-processing Environment (gaff-gen)

Run in **DGX Host Terminal**:

```
# Create environment
micromamba create -p /home/geneon_dgx/micromamba/envs/gaff-gen python=3.10 -c cor

# Install required packages
micromamba install -p /home/geneon_dgx/micromamba/envs/gaff-gen -c conda-forge rc
```

2.2. Setup Simulation Environment (Docker)

Run in **DGX Host Terminal**:

```
# Pull Image
docker pull nvcr.io/hpc/gromacs:2023.2

# Run Container (Mount Data Volume)
export PROJECT_ROOT="/srv/dgx_ws/storage/CADD_chemflow_g/results/AF_P10275_F1_moc
docker run --gpus all --name gmx_session -v ${PROJECT_ROOT}:/project -it -d nvcr.
```

3. How to Access & Use

👉 Scenario A: Modifying Files or Preparing Ligands

Use **gaff-gen** environment on Host.

```
# 1. Activate Environment
micromamba activate gaff-gen
```

```
# 2. Run Python/Obabel commands
python optimize_ligand.py 1V4
obabel -ipdb ... -omol2 ...

# 3. Deactivate
micromamba deactivate
```

👉 Scenario B: Running MD Simulations (gmx)

Use **GROMACS** container.

```
# 1. Enter Container
docker exec -it gmx_session bash

# 2. Set Environment Variables (Required every time)
export GMX_BIN="/usr/local/gromacs/arm_neon_asimd/bin/gmx"
export GMX_GPU_ENABLE=false

# 3. Run GROMACS Commands
cd /project/md_simul/gromacs/ZZA_run
${GMX_BIN} grompp -f ...
```

4. Execution Script (run_md_cpu.sh)

Save this script in your run directory (e.g., `/project/md_simul/gromacs/ZZA_run/run_md_cpu.sh`) inside the container.

```
#!/bin/bash
export GMX_BIN="/usr/local/gromacs/arm_neon_asimd/bin/gmx"
MDP_PATH="../../scripts"
THREADS="-ntmpi 4 -ntomp 5"
MD_OPTS_CPU="-nb cpu -pme cpu -bonded cpu -update cpu"
set -e

echo "=== MD Pipeline Start (CPU Mode) ==="
date

# 1. Minimization
echo "[Step 1] Energy Minimization..."
${GMX_BIN} mdrun -s run_minim_final.tpr -deffnm minim -v ${THREADS} ${MD_OPTS_CPU

# 2. NVT
echo "[Step 2] NVT Equilibration..."
${GMX_BIN} grompp -f ${MDP_PATH}/nvt.mdp -c minimized.gro -r minimized.gro -p top
${GMX_BIN} mdrun -s nvt.tpr -deffnm nvt -v ${THREADS} ${MD_OPTS_CPU} -c nvt.gro

# 3. NPT
echo "[Step 3] NPT Equilibration..."
${GMX_BIN} grompp -f ${MDP_PATH}/npt_eq.mdp -c nvt.gro -r nvt.gro -t nvt.cpt -p t
${GMX_BIN} mdrun -s npt.tpr -deffnm npt -v ${THREADS} ${MD_OPTS_CPU} -c final_eq.

# 4. Production
echo "[Step 4] Production MD (50ns)..."
${GMX_BIN} grompp -f ${MDP_PATH}/prod.mdp -c final_eq.gro -r final_eq.gro -t npt.
```

```
        ${GMX_BIN} mdrun -s run_prod_50ns.tpr -deffnm gmx_50ns_cpu -v ${THREADS} ${MD_OPT

        echo "✓✓✓ MD Simulation Completed!"
        date
```