

Einführung in Simulationen von Biomolekülen

Sabine Reißer and Tomáš Kubař

Contents

I. Einführung, Visualisierung und Geometrieoptimierungen	2
A. Grundlagen	2
B. Visualisierung mit VMD	3
C. Moleküle erstellen	4
D. Erstellung einer Topologie mit pdb2gmx	5
II. Moleküldynamiksimulationen	6
A. Ein Peptid im Wasser simulieren	6
1. Solvatation	6
2. Equilibrierung	7
3. MD Simulation	8
B. Visualisierung	9
C. Normalmodenanalyse mit Gromacs	9
1. Normalmoden eines Wassermoleküls	9
2. Normalmodenanalyse eines Peptidmoleküls	10
D. Hauptkomponentenanalyse	10
E. Ramachandran Plot und Zeitserien für Diederwinkel	11
III. Extended Sampling Methoden	12
A. Wiederholung: Vorbereitung und Simulation des Dipeptids	12
1. Vorbereitung des Dipeptids	12
2. Simulation des Dipeptids	13
B. Metadynamik eines Dipeptids	13
1. Intro	13
2. Durchführung	14
C. Analyse	14
1. Freie Energie der <i>unbiased</i> Simulation	14
2. Freie Energie der Metadynamik	15
IV. QM/MM Simulation	16
A. Equilibrierung mit MM	16
B. Vorbereitung	16
C. Ausführung	17
D. Auswertung	18
E. Metadynamik	19

All the files for this tutorial will be available at the website:

<http://cbp.cfn.kit.edu/downloads/MD-SS2017/materials-prak/>

I. EINFÜHRUNG, VISUALISIERUNG UND GEOMETRIEOPTIMIERUNGEN

A. Grundlagen

Für dieses Praktikum werden die Linuxrechner im Seminarraum 110, Geb. 30.41 verwendet. Wir arbeiten unter Linux, einem freien Betriebssystem, das weniger verbreitet ist als z.B. Windows, aber für wissenschaftliche Anwendungen breite Verwendung findet. Im Unterschied zur bekannten mausgesteuerten Bedienung von Desktop-PCs ist Linux ein kommando-basiertes System, d.h. dass Nutzer den Rechner hauptsächlich über eine Textkonsole steuern. Dies ist wesentlich effizienter und schneller, bedarf aber einiger Übung. Vielfältige Informationen und Tutorien über Linux/Unix sind im Internet zugänglich, daher wird im folgenden nur ein minimaler Überblick über die üblichsten Befehle gegeben, die im Laufe des Praktikums gebraucht werden.

Zunächst fahren Sie einen der Praktikumsrechner hoch. Dann loggen Sie sich auf dem Rechner ein. Dazu verwenden Sie als Benutzernamen 'student' und als Passwort 'student'. Der Rechner startet eine graphische X-Window Oberfläche. **Bitte beachten Sie:** Der Rechner muss nie heruntergefahren werden, und Sie müssen sich während des Praktikums nie abmelden. Um Textkommandos eingeben zu können starten Sie dann noch eine sog. Shell, ein Fenster verknüpft mit einem Kommandointerpreter. Diese finden Sie im Menu unter Favorites. In der dann startenden Shell können Sie zunächst die grundlegendsten Unixbefehle üben, z.B.:

- `pwd` – Zeigt das aktuelle Verzeichnis an, im Moment das home Verzeichnis, in dem Sie volle Schreib- und Leserechte haben und in dem Sie alle Dateien für das Praktikum ablegen sollten.
- `ls` – zeigt den Inhalt eines Verzeichnisses an. Ihr Homeverzeichnis sollte im Moment leer sein (oder Dateien früherer Nutzer enthalten). Es gibt verschiedene Modi für `ls`, die mit sog. Switches gewählt werden. Z.B. gibt `ls -l` zusätzliche Informationen über Dateien, wie Nutzerrechte aus, oder `ls -ltrh` sortiert die Dateien so, dass die zuletzt erstellte Datei am Ende der Liste steht.
- `man befehl` – Zeigt das Handbuch für einen bestimmten Befehl an. Wenn Sie wissen möchten, welche Optionen (Switches) ein bestimmter Befehl akzeptiert, können Sie dies mit `man` erfahren.
- `mkdir name` – Erstellt Verzeichnisse. Erstellen Sie in ihrem Home-Verzeichnis für die Übungen ein Subverzeichnis namens uebungen.
- `cd name` – Wechselt das Arbeitsverzeichnis. Wechseln Sie mit `cd uebungen` in das gerade erstellte Verzeichnis. Der Name des Zielverzeichnisses kann relativ zum Ausgangsverzeichnis angegeben werden, oder auch absolut, also `cd /home/student/uebungen`. Mit `cd ..` wechseln Sie in das Verzeichnis über dem aktuellen, mit `cd` ohne Name zurück in ihr Home-Verzeichnis. Ein einzelner Punkt `.` steht dabei für das aktuelle Verzeichnis.
- `cp oldname newname-or-location`, `mv oldname newname-or-location`, `rm name` – Dateioperationen um Dateien zu kopieren, verschieben (umbenennen) oder zu löschen. `cp name1 name2` erstellt eine Kopie der Datei `name1` unter `name2`. Dabei können auch unterschiedliche Verzeichnisse angegeben werden. Mit `mv name1 name2` würde eine Datei entsprechend verschoben oder umbenannt werden; `name2` ist entweder das Zielverzeichnis oder der neue Dateiname. Dateien löschen funktioniert mit `rm name`, wobei Vorsicht geboten ist, denn löschen kennt keine Sicherheitsabfrage und kann nicht rückgängig gemacht werden. Komplette Verzeichnisse können mit `rm -r ordnername` gelöscht werden.
- `tar` – Packen, entpacken und archivieren von Dateien. Mit diesem Befehl können Dateien in Archive verpackt und komprimiert werden, z.B. zum einfacheren Versenden oder für Sicherheitskopien.
- `cat`, `more`, `less`, `head`, `tail name` – Betrachten von Textdateien. `cat` gibt die gesamte Datei auf einmal auf den Bildschirm aus, für längere Dateien bieten sich `more` oder `less` an, was seitenweises Scrollen erlaubt. `head` und `tail` geben jeweils die ersten und letzten zehn Zeilen der Datei aus.
- `grep` – Dateien durchsuchen. Mit `grep muster datei` wird die Datei nach Zeilen durchsucht in denen die Zeichenkombination `muster` vorkommt. Beispielsweise gibt

```
grep HA /usr/local/run/gromacs-5.0-dftb-v6a-plumed/share/gromacs/top*/ffbonded.itp
```

alle Zeilen aller Kraftfelddateien für Bindungsparameter an, die sich auf einen bestimmten Wasserstoffatomtyp beziehen.

- **gedit, vim, nano, pico, emacs** – Texteditoren. Um den Inhalt von Textdateien zu editieren gibt es eine Vielzahl von Texteditoren, die zu groß ist um hier auch nur ansatzweise aufgeführt zu werden. Im Rahmen der Übungen genügt es, z.B. den einfachen Editor **gedit** zu verwenden. Wer weiter auf Linuxsystemen arbeiten möchten, sollte sich aber früher oder später mit einem mächtigeren Editor wie **emacs** oder **vim** anfreunden. Betrachten Sie zur Übung die PDB Datei **1HSG.pdb**.
- **alias name='befehl'** – Weist einem Befehl ein Kürzel zu. So könnten Sie z.B. mit **alias ll='ls -ltrh'** eine Kurzform für den detaillierten Dateilistenbefehl definieren.
- Ebenfalls sehr praktisch ist die Möglichkeit, einen mit der linken Maustaste ausgewählten Text anschließend mit der mittleren Maustaste (oder Rad) zu kopieren. Dies geht oft schneller als die von Windows bekannte Sequenz von Strg-C Strg-V, welche allerdings in modernen Linux-Systemen auch funktioniert.
- Mit Strg-R können Sie nach einem Muster in der Befehls Geschichte suchen.
- Mit Tab können Sie Dateinamen oder Befehle automatisch vervollständigen.
- **>** – Das größer-als Zeichen leitet die Ausgabe aus einem Programm (den sog. *standard output* oder **STDOUT**) oder Befehl in eine Datei. Wenn sie den Inhalt des aktuellen Verzeichnis in einer Textdatei abspeichern möchten, tun Sie das mit **ls > was-ist-hier.txt**
- **|** – Der senkrechte Strich leitet die Ausgabe auf die Standardeingabe von einem anderen Programm oder Befehl. Diese sog. Pipe kann man auch mehrere Male auf der Kommandozeile benutzen. So kann man z.B. abzählen, wie oft man sich als Benutzer 'student' auf dem Rechner angemeldet hat: **last | grep "student" | wc -l**

B. Visualisierung mit VMD

Eine der wichtigsten Techniken für molekulare Simulationen ist die anschließende Visualisierung der berechneten Strukturen und Trajektorien. Wir werden dazu das Molekülbetrachtungsprogramm VMD verwenden (das unter <http://www.ks.uiuc.edu/Research/vmd> frei erhältlich ist) Bevor wir eigene Moleküle bauen, wollen wir zunächst eine fertige Röntgenkristallstruktur betrachten. Starten Sie zunächst das Programm aus der Shell heraus mit

```
vmd
```

Es öffnet sich ein Betrachtungs- und ein Kontrollfenster. Öffnen Sie den Dateibrowser unter *File – New Molecule*. Im neu geöffneten Fenster wählen Sie *Determine file type – Web PDB Download* und laden die Röntgenkristallstruktur Datei **1HEL**, oder gerne eine andere, z.B. die von Ihrem Lieblingsprotein. Danach wird die Kristallstruktur im Betrachtungsfenster angezeigt. Testen Sie an diesem Beispiel die verschiedenen Funktionen von VMD:

- Drehen, Verschieben und Skalieren: Mit gedrückter linker oder rechter Maustaste lässt sich das Molekül drehen. Durch Drücken der **t**-Taste gelangen Sie in den Translationsmodus, mit **s** in den Skalierungsmodus und mit **r** zurück in den Drehmodus. Mit **=** wird die Ansicht zurückgesetzt.
- Auswahl von Teilen des Moleküls: Öffnen Sie unter *Graphics – Representations* das Darstellungsfenster. In der Zeile *Selected Atoms* probieren Sie verschiedene Selektionen aus: "water", "backbone", "resid 1 to 10", "resname CYS", etc. Selektieren Sie am Ende nur die beiden Cysteine Nummer 30 und 115.
- Namen und innere Koordinaten: Mit den Tasten **1, 2, 3, 4** wählen Sie aus, Informationen über einzelne Atome, Abstände, Winkel und Dieder zu bekommen. Drücken Sie **1** und wählen Sie eines der Schwefelatome an. In der Konsole werden einige Informationen über dieses Atom angezeigt. Messen Sie weiterhin die Distanz (mit **2**) zwischen den Schwefelatomen, die beiden **S-S-C β** Winkel (Modus **3**) und den **C β -S-S-C β** Diederwinkel (Modus **4**). Sollte der Bildschirm mit Labels überfüllt sein, so kann man diese unter *Graphics – Labels* wieder löschen.
- Verschiedene Darstellungsmöglichkeiten: Setzen Sie die Selektierung zurück auf "all". Wählen Sie nun "Secondary Structure" und "NewCartoon" als Farb- und Zeichenstil. Wählen Sie ein paar weitere Darstellungsmöglichkeiten für das Molekül und generieren Sie eine Ansicht Ihrer Wahl, z.B. eine in der die Disulfidbrücken gut zur Geltung kommen, oder eine in der die Verteilung geladener Reste im Protein ersichtlich wird.

Für eine ausführlichere Beschreibung der (vielfältigen) Darstellungsmöglichkeiten mit VMD experimentieren Sie weiter oder nehmen Sie den User Guide zu Hilfe. VMD eignet sich gut dazu, hochwertige Strukturbilder für Publikationen über das *File – Render* Menü zu erstellen. Wir werden im Verlauf des Praktikums noch von den Möglichkeiten Gebrauch machen, Trajektorien darzustellen und Strukturen zu vergleichen.

C. Moleküle erstellen

Um im folgenden Geometrieoptimierungen oder Moleküldynamiksimulationen zu starten, benötigen Sie im wesentlichen zwei Dinge: Startgeometrien und Kraftfeldparameter. Startgeometrien stammen normalerweise aus experimentellen Daten, z.B. Röntgen- oder NMR-Strukturen, oder aus anderen Modellierungstechniken. Kraftfeldparameter sind normalerweise Bestandteil des verwendeten Modellingpaketes, können prinzipiell aber auch extra erstellt werden. Für die Vorbereitung der Eingabedateien sind im Amberpaket zwei Programme verantwortlich: Leap und Antechamber. Letzteres dient dazu, neue Moleküle zu generieren und Parameter dafür anzupassen, und wird in diesen Übungen nicht verwendet. Wir werden die graphische Version von ersterem (xLeap) verwenden, um ein Peptid zu bauen und als pdb-Datei zu speichern.

xleap

Als nächstes laden Sie eines der Amber Kraftfelder (eine verbesserte Version des amber99-Kraftfelds) und betrachten einige der vordefinierten Einheiten darin. (Sog. Einheiten sind Atome, Moleküle oder Teile von Molekülen, sowie Parameter.)

```
source leaprc.ff14SB
list
desc ALA
desc ALA.1
desc ALA.1.3
edit ALA
edit TIP3PBOX
```

Üben Sie das Betrachten, Markieren und Editieren von Molekülen und Atomen mit dem "edit" Fenster von xLeap. Das Programm stellt verschiedene Möglichkeiten zur Verfügung, solvatisierte und periodische Systeme zu erstellen, momentan beschränken wir uns allerdings auf Systeme im Vakuum.

Für die folgenden Rechnungen erstellen Sie nun das Peptid K2 als neue Einheit `peptid` mit dem Kommando

```
source leaprc.ff14SB
peptid = sequence { NLYS ILE ALA GLY LYS ILE ALA LYS ILE ALA GLY LYS ILE ALA NHE }
```

Im Prinzip hätte man das Molekül auch von Hand in eine leere Einheit zeichnen und dann alle Atomtypen und -ladungen definieren können, aber die vorgefertigten Aminosäurereste aus dem Kraftfeld erleichtern diese Arbeit natürlich enorm.

Als Startstruktur für die Simulationen möchten wir mit einer idealen α -Helix arbeiten, daher muss das Peptid noch entsprechend gefaltet werden. Die Diederwinkel φ und ψ sind entscheidend für die Sekundärstruktur von Proteinen. Im Gegensatz zur relativ starren Peptidgruppe selbst sind mehrere verschiedene Konformationen der beiden Diederwinkel denkbar. φ ist der N-terminale der beiden, der die Rotation entlang der N-C α Bindung beschreibt (Diederwinkel C(vorherige AS)-N-C α -C), ψ beschreibt analog die Rotation entlang C α -C (Diederwinkel N-C α -C-N(nächste AS)):

Wir setzen die Diederwinkel der Peptidbindung auf die Werte für eine ideale α -Helix (schauen Sie die in einem Buch oder im Internet nach):

```
impose peptid { 1 2 3 4 5 6 7 8 9 10 11 12 13 14 } {
{ "N" "CA" "C" "N" Wert-für-psi }
{ "C" "N" "CA" "C" Wert-für-phi }
}
```

Nun sollte K2 eine α -Helix sein. Speichern Sie die Struktur mit

```
savepdb peptid k2.pdb
```

Sie können auch eine Reihe von Befehlen in einer Textdatei abspeichern und diese mit

```
xleap -f Dateiname
```

sequentiell abarbeiten lassen. Ein Beispiel dazu ist die Eingabedatei `leap.in`.

Weitere Möglichkeiten von xLeap entnehmen Sie dem User Guide `Amber-Tools.pdf`.

Betrachten Sie das Peptid nun mit VMD und überprüfen Sie, ob alles so aussieht wie erwartet. Schauen Sie sich auch die Datei selbst mit einem Texteditor (z.B. Nano oder Pico) an. Versuchen Sie, sich folgende Fragen zu beantworten:

- Welche Spalte bedeutet was?
- In welcher Einheit sind die Koordinaten angegeben?
- Ist das Peptid amidiert?

D. Erstellung einer Topologie mit `pdb2gmx`

Die PDB Datei, die wir erstellt haben, enthält nur die Struktur des Moleküls, aber keine Atomladungen, Massen, Bindungsinformationen etc. Für die Zuweisung der Parameter wie auch für alle weiteren Schritte benutzen wir ab jetzt das MD-Simulationsprogramm Gromacs. Gromacs ist wie alle anderen bisher verwendeten Programme *open source* – das heißt, der Quellcode ist frei verfügbar und jeder kann ihn nach seiner Präferenz verändern.

Wir verwenden jetzt das Gromacs tool `pdb2gmx`, um Kraftfeldparameter zu erhalten:

```
gmx pdb2gmx -f k2.pdb -ignh -o k2.gro
```

Wir verwenden das Amber99SB-ILDN Kraftfeld und brauchen zunächst kein Wasser. Die Outputdatei `k2.gro` ist eine Übersetzung der PDB Datei ins Gromacs-Format. Ein Blick hinein zeigt, dass die Formate sehr ähnlich sind.

Interessanter ist die Datei `topol.top`. Sie enthält alle Kraftfeldparameter und die Topologie, also die Zusammensetzung des Systems. Zeilen mit einem Semikolon ; am Anfang sind Kommentare und sind zur Information des Nutzers gedacht.

In dieser Datei werden unter [`moleculetype`] die Molekültypen definiert, in unserem Fall ist das das Peptid K2. Danach werden unter [`atoms`] die Atome aufgelistet, in der Reihenfolge der einzelnen Seitenketten. Für jedes Atom steht nun in Spalte 2 ein Atomtyp, in Zeile 7 eine Ladung und in Zeile 8 eine Masse.

Weiter unten werden unter [`bonds`] die Bindungen zwischen Atomen definiert. Die ersten zwei Zahlen stehen für die Indices der Atome wie unter [`atoms`] definiert. Die Bindungsstärken und -abstände sind im Kraftfeld definiert, was am Anfang der Datei eingebunden wurde:

```
; Include forcefield parameters
#include "amber99sb-ildn.ff/forcefield.itp"
```

Der Pfadname ist hier ausgehend vom aktuellen Verzeichnis oder vom Verzeichnis `/usr/local/run/gromacs-5.0-dftb-v6a-plumed/share/gromacs/top`. Hier finden Sie alle Kraftfelder und die dazugehörigen Parameter. Ganz unten in der Topologiedatei (bei dem Betrachtungsprogramm `less` springt man über `G` zum Dateiende) finden wir nun die Zusammensetzung des Systems mit dem Namen "Protein", was unter [`molecules`] nur ein Molekül mit ebenfalls dem Namen "Protein" enthält.

II. MOLEKÜLDYNAMIKSIMULATIONEN

A. Ein Peptid im Wasser simulieren

In Moleküldynamiksimulationen wird, im Gegensatz zu Geometrieoptimierungen, die zeitliche Entwicklung eines molekularen Systems möglichst realistisch simuliert. Solche Simulationen sind in der tatsächlichen Anwendung sehr zeit- und rechenintensiv. Wir werden daher im Praktikum selbst nur kurze Zeitentwicklungen simulieren. Längere Trajektorien zur Auswertung finden Sie dann jeweils im Praktikumsverzeichnis.

Erstellen Sie nun einen Ordner **part2**. Kopieren Sie die Struktur des Peptids K2 und die Topologie-Datei aus der letzten Übung in dieses Verzeichnis.

Wir packen in dieser Übung das Peptid K2, das Sie im letzten Praktikumsteil erstellt und geometrie-optimiert haben, in eine Wasserbox, heizen diese auf, equilibrieren das System und starten eine kurze Simulation.

Die im Folgenden verwendeten Programme sind Teil des Gromacs Programmpaketes und benötigen verschiedene Eingabedateien, die über Switches zugeordnet werden. In Gromacs werden alle Simulationen inklusive der Geometrieoptimierung mit dem Preprozessor **grompp** vorbereitet und mit dem Programm **mdrun** ausgeführt. Die wichtigsten Switches fuer **grompp** sind beispielsweise:

Switch	Dateiendung	Erklärung
-f	.mdp	Simulationsparameter
-c	.gro	Startstruktur
-p	.top	Topologie
-n	.ndx	Indexdatei
-o	.tpr	Output

Bitte beachten Sie, dass je nach aufgerufenem Programm die Switches andere Dateien oder Eingaben erfordern. Fuer jedes Programm kann ein Hilfetext mit dem Switch **-h** erzeugt werden, der alle Optionen beschreibt.

1. Solvation

Zunächst packen wir das Peptid in eine kubische Box:

```
gmx editconf -f k2 -o k2_newbox -c -d 1.0 -bt cubic
```

Lesen Sie unter **editconf -h** die verwendeten Optionen nach. Schauen Sie sich mit VMD die neue Struktur an. Wenn Sie **pbx box** in die VMD-Konsole eingeben, sehen Sie die Kanten der erstellten Box.

Nun füllen wir die Box mit Wasser. Die verschiedenen Wassermodelle können Sie unter en.wikipedia.org/wiki/Water_model nachschauen; wir benutzen das TIP3P Modell.

```
gmx solvate -cp k2_newbox -p topol -cs spc216 -o k2_sol
```

Lesen sie auch hier unter **gmx solvate -h** die Optionen nach. Die verwendete Wasserbox SPC216 enthält 216 Wassermoleküle und ist mit dem SPC Modell equilibriert (die Modelle SPC und TIP3P sind sehr ähnlich). Das Programm versucht möglichst viele dieser kleinen Boxen in der Peptidbox unterzubringen. Schauen Sie sich die solvatisierte Box wieder mit VMD an.

Wenn Sie sich nun mit **tail topol.top** das Ende der Topologie ausgeben lassen, sehen Sie unter [**molecules**] die Anzahl der hinzugefügten Wassermoleküle. Wir müssen jetzt noch die Moleküldefinition für TIP3P Wasser in die Topologie einfügen. Ergänzen Sie mit einem Texteditor vor [**system**] die drei Zeilen

```
;Include water topology
#include "amber99sb-ildn.ff/tip3p.itp"
#include "amber99sb-ildn.ff/ions.itp"
```

Jetzt müssen Sie das System neutralisieren, weil das Peptid eine positive Ladung trägt. (Frage: Welche Aminosäuren sind dafür zuständig?) Dazu brauchen Sie eine beliebige Inputdatei, z.B. eine leere Datei, die wir mit diesem Befehl erzeugen:

```
echo > dummy.mdp
```

Dann können Sie eine **.tpr** Datei erzeugen, und diese schließlich dem Programm geben, das die Ionen hinzufügt:

```
gmx grompp -f dummy -o dummy -p topol -c k2_sol
gmx genion -s dummy -o k2_ion -p topol -neutral
```

Dieses Programm ersetzt die benötigte Anzahl Wassermoleküle durch Cl^- Ionen. Geben Sie hierzu an, welche der Gruppen Wasser bezeichnet (SOL, wahrscheinlich die 13).

2. Equilibrierung

Um das System für eine richtige Simulation vorzubereiten, müssen jetzt einige Equilibrierungsschritte durchgeführt werden. Zunächst machen wir wieder eine Energieminimierung, um eine vernünftige Startstruktur zu erhalten. Benutzen Sie dazu die Datei `em_steep.mdp`, die noch nicht eingesetzt wurde. Sie können folgende Befehle durchführen:

```
gmx grompp -f em_steep -c k2_ion -p topol -o em
gmx mdrun -v -deffnm em
```

Nun wird die energieminierte Struktur bei konstantem Volumen aufgeheizt (NVT Simulation). Dazu wird das Peptid festgehalten, so dass sich nur die Wassermoleküle um es herum bewegen können. Um das Peptid festzuhalten, erstellen wir sog. Position Restraints für jedes Peptid-Atom mit `gmx genrestr`:

```
gmx genrestr -f em
```

Hier können Sie die schweren Atome (alle außer H) des Peptids berücksichtigen, d.h. die Gruppe "Protein-H". Wie Sie in der erstellten Datei `posre.itp` sehen können, wird jetzt die Position jedes schweren Peptid-Atoms durch ein harmonisches Potential (Feder) mit der Kraftkonstante $1000 \text{ kJ}/(\text{mol}\cdot\text{nm}^2)$ mit seiner Anfangsposition festgehalten. (Siehe auch Gromacs Manual, Abschnitt 4.3.1.)

Schauen Sie sich nun die Datei `nvt.mdp` an. Auch hier finden Sie Auskunft zu den einzelnen Parametern. Hier einige zusätzliche Informationen:

- `define = -DPOSRES`
Hier wird die Variable POSRES definiert, also Position Restraints werden angewendet. Suchen Sie in der Datei `topol.top` nach "POSRES", dort wird die von Ihnen erstellte Datei `posre.itp` eingebunden. Wenn Sie später eine freie MD Simulationen muss diese Zeile in der entsprechenden `.mdp` Datei auskommentiert oder gelöscht sein.
- `gen_temp = 10.`
Alle Atome bekommen am Anfang der Simulation durch Geschwindigkeiten stochastisch (zufällig) zugewiesen, deren Verteilung der Maxwell-Boltzmann-Verteilung für die erwünschte Temperatur entspricht. Bei 10 K sind die Geschwindigkeiten gering und es kann keine Atombewegung in einer ungeschickten Richtung entstehen.
- `ref_t = 300. 300.`
Im Verlauf der Simulation werden die Geschwindigkeiten über einen Thermostaten so skaliert, dass die Verteilung möglichst konstant der MB-Verteilung für die erwünschte Temperatur `ref_t` entspricht.

Führen Sie nun wieder den Preprozessor und die eigentliche Simulation aus:

```
gmx grompp -f nvt -p topol -c em -o nvt
gmx mdrun -v -deffnm nvt
```

Wie Sie über die Bilschirmausgabe sehen können, dauert die NVT Equilibrierung etwa eine Minute.

Plotten Sie mit `gmx energy` den Verlauf der Temperatur:

```
gmx energy -f nvt -o nvt_temp
```

Betrachten Sie die Datei mit Xmgrace:

```
xmgrace nvt_temp.xvg
```

(dies kann direkt mit dem Switch `-w` für `gmx energy` aufgefordert werden).

Ihr Diagramm sollte in etwa wie Abb. 1 (links) aussehen. Sie sehen, dass bei 0 ps die Temperatur sehr niedrig ist. Die Ursache dafür ist die Zuteilung der Geschwindigkeiten gemäß der Boltzmann-Verteilung für 10 K. Durch den verwendeten Thermostat wird dann das System während der ersten ca. 2 ps auf die Zieltemperatur von 300 K aufgeheizt.

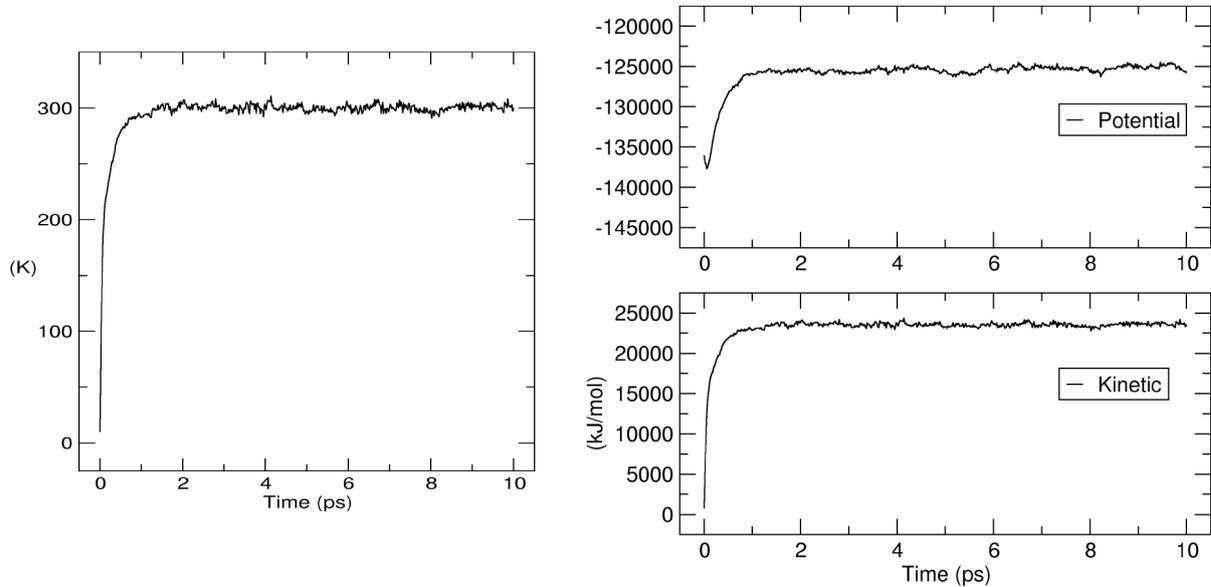


FIG. 1: NVT Equilibration: Verlauf der Temperatur (links), sowie der potentiellen und kinetischen Energie (rechts).

Erstellen Sie mit `gmx energy` auch Diagramme vom Verlauf der potentiellen und kinetischen Energie, die ähnlich wie in Abb. 1 (rechts) aussehen werden. Nach einer kleinen Absenkung steigt die potentielle Energie, und illustriert so den Fakt, dass das Molekülsystem sich während der Simulation nicht mehr exakt im Energieminimum aufhält. Die kinetische Energie verläuft gleich wie die Temperatur. (Warum?)

Nun muss das System noch auf den richtigen Druck von 1 bar gebracht werden. In diesem Equilibrierungsschritt wird gleichzeitig der Druck und die Temperatur konstant gehalten, daher nennt man ihn auch NPT Equilibrierung.

Schauen Sie sich die Datei `npt.mdp` an. Das Peptid wird weiterhin mit Position Restraints festgehalten. Die Zeile

```
continuation = yes
```

zeigt an, dass die Endgeschwindigkeiten der letzten Equilibrierung jetzt als Startgeschwindigkeiten genutzt werden. Es gibt jetzt einen neuen Block zur Kontrolle des Drucks:

```
pcoupl          = Parrinello-Rahman
pcoupltype      = isotropic          ; uniform in x,y,z directions
tau_p           = 0.5                ; relaxation time in ps
ref_p           = 1.0                ; desired pressure in bar
compressibility = 4.5e-5             ; value for water in 1/bar
refcoord_scaling = com
```

Zum Ausführen der Equilibrierung benutzen wir wieder `grompp` und `mdrun`:

```
gmx grompp -f npt -c nvt -p topol -o npt
gmx mdrun -v -deffnm npt
```

Diese 100 ps lange Equilibrierung wird etwa 5 Minuten dauern. Schauen Sie sich danach wieder mit `gmx energy` und `Xmgrace` die Entwicklung der Temperatur und zusätzlich der Dichte an. (Oder alternativ des Volumens. Die Information ist die gleiche wie bei der Dichte, die allerdings als intensive Eigenschaft vielleicht mehr informativ ist.) Ist das System ausreichend equilibriert? Innerhalb welcher Grenzen schwanken Temperatur und Dichte?

3. MD Simulation

Nachdem wir unser System nun ausreichend equilibriert haben, starten wir eine MD Simulation, in der wir das Peptid nicht mehr festhalten. Betrachten Sie die dazugehörige Eingabedatei `md.mdp`. Was hat sich verändert? Woran sehen Sie, dass das Peptid sich nun frei bewegen kann? Welche Zeitspanne wird simuliert? Benutzen Sie nun `npt.gro` als Startstruktur und starten Sie die Simulation. Die Simulation wird etwa eine Stunde dauern.

B. Visualisierung

Verwenden Sie nun VMD um den Verlauf der Trajektorie zu visualisieren. Übergeben Sie dazu VMD als Eingabeparameter erst eine Startstruktur und dann die Trajektorie:

```
vmd npt.gro md.xtc
```

Experimentieren Sie mit dem Trajektorienplayer im Fenster “VMD Main”.

Blenden Sie zuerst die Wassermoleküle aus. Die Dynamik des Peptids ist zunächst noch schwer zu verfolgen, da die Rotation des Moleküls alle anderen Konformationsänderungen überdeckt, dies lässt sich aber mit Hilfe des *RMSD-Tools* in VMD beheben. Wählen Sie unter *Extensions – Analysis* das Fenster *RMSD Trajectory Tool*. Selektieren Sie das Peptid (mit “protein”) und wählen Sie *Align*. Damit sollte die Trajektorie translations- und rotations-bereinigt sein.

Speichern Sie den berechneten RMSD Verlauf der Simulation und plotten Sie die erhaltene Datei (Sie müssen dazu vermutlich die ersten beiden Zeilen der Datei entfernen). Verfolgen Sie als nächstes die Änderungen der beiden Peptiddiederwinkel φ und ψ in Ihrem Modell zunächst visuell. Diese beiden Diederwinkel sind entscheidend für die Sekundärstruktur von Proteinen (φ : C(-1)-N-C α -C, ψ : N-C α -C-N(+1)). Markieren Sie diese in VMD im Diedermodus (Taste 4). Wählen Sie im Fenster *Main* unter *Extensions – Analysis* das Fenster *Ramachandran Plot*. Im neuen Fenster müssen Sie unter *Molecule* nochmal “npt.gro” anwählen, dann sollten Sie die Position des aktuell angezeigten Koordinatensets im Ramachandran Plot sehen. Verfolgen Sie die Bewegung des Peptids in diesem Konformationsraum während der Simulation indem Sie im Fenster *Main* die Trajektorie abspielen lassen. Die normalerweise bevorzugten Konformationen für Proteine sind angezeigt, ein einzelnes Peptid kann aber durchaus stark davon abweichen. Ordnen Sie den farblich unterlegten Feldern Konformationen zu.

C. Normalmodenanalyse mit Gromacs

Alternativ zu MD Simulationen bieten Normalmodenanalysen (NMAs) eine Möglichkeit, sich einen Überblick über den Konformationsraum eines Moleküls zu verschaffen. Zudem lässt sich aus diesen, zumindest in harmonischer Näherung, die Konfigurationsentropie eines Moleküls berechnen, eine normalerweise nur schwer zugängliche Größe. Wir werden für ein kleines Testmolekül die Normalmoden zunächst berechnen und dann mit Hilfe von VMD visualisieren. Für das Ausführen einer NMA benötigt man drei wesentliche Schritte:

- Minimierung der konformellen Energie als Funktion der Koordinaten (Geometrieoptimierung),
- Berechnen der Hessematrix, die zweiten Ableitungen der potentiellen Energie nach den Koordinaten,
- Diagonalisierung der Hessematrix, um die Normalmoden mit Hilfe der Eigenwerte (Schwingungsfrequenzen) und Eigenvektoren (Schwingungskordinaten) zu erzeugen.

1. Normalmoden eines Wassermoleküls

Als einfaches System werden wir zunächst die Normalmoden eines Wassermoleküls berechnen. Die größte Rechnung ist die Geometrieoptimierung, da für eine NMA muss eine Struktur so nahe wie möglich an einem Minimum gefunden werden muss. Die erste Rechnung ist eine *Steepest Descent* Minimierung (mit `nma_em_steep.mpd`) und anschließend eine *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) Minimierung. Zunächst erstellen wir eine Topologie:

```
gmx pdb2gmx -f system.gro -o water.gro
```

Wir wählen ein universelles Amber Kraftfeld und das TIP3P Modell fuer Wasser. Für den nächsten Schritt ist es sehr wichtig, dass jeweils die *double precision* Programme des Gromacs Pakets benutzt werden (gekennzeichnet jeweils durch ein Suffix `_d` am jeweiligen Programm):

```
gmx_d grompp -f nma-em-steep.mdp -c water.gro -p topol.top -o water-min1.tpr
gmx_d mdrun -v -deffnm water-min1
```

In der zweiten Optimierung ist es besonders wichtig, den switch `-t` mit `gmx_d grompp` zu benutzen, da somit die vollständigen *double precision* Koordinaten binär ausgelesen werden (anstatt der gerundeten in der `.gro` Datei):

```
gmx_d grompp -f nma-em-bfgs.mdp -c water-min1.gro -t water-min1.trr -p topol.top -o water-min2.tpr
gmx_d mdrun -v -deffnm water-min2
```

Nachdem wir nun ein minimiertes Molekül erhalten haben, können wir die Hessematrix dieser Struktur erstellen:

```
gmx_d grompp -f nma.mpd -c water-min2.gro -t water-min2.trr -p topol.top -o water-nma.tpr
gmx_d mdrun -v -deffnm water-nma
```

Die Matrix hat eine Größe von $3N \times 3N$ (wobei N die Anzahl der Atome ist und die drei für die Anzahl der kartesischen Koordinaten steht). Nun muss diese diagonalisiert werden:

```
gmx_d nmeig -f water-nma.mtx -s water-nma.tpr
```

Dieser Prozess resultiert in einem Set von $3N$ Eigenwerten in `eigenval.xvg` und einem Set von $3N$ Eigenvektoren in `eigenvec.trr`, wobei jeder Vektor $3N$ -dimensional ist. Sechs Eigenwerte sind nahe 0 und entsprechen den drei Translations- und drei Rotationsfreiheitsgraden. Alle anderen Eigenwerte sollten positiv sein (sonst würde sich das System nicht an einem Minimum befinden). Die Korrektheit der Werte können Sie in der Datei `eigenval.xvg` überprüfen. Das Wassermolekül hat drei Schwingungsfreiheitsgrade, die wir nun mit VMD visualisieren werden.

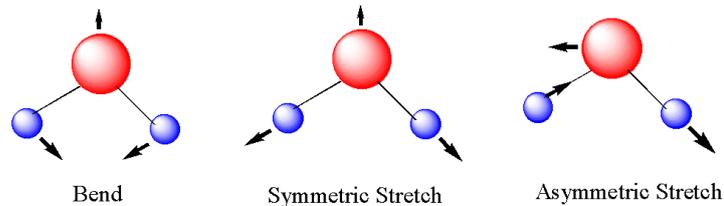


FIG. 2: Vibrational degrees of freedom of a water molecule.

Hierzu muss zunächst eine Trajektorie aus der Normalmode konstruiert werden:

```
gmx_d ana eig -first 7 -last 7 -v eigenvec.trr -nframes 50 -max 0.1 -s water-nma.tpr -extr node7.pdb
```

Bei diesem Befehl steht die sieben für die Nummer der Mode, die visualisiert werden soll. Die generierte PDB Datei kann nun mit VMD geöffnet und betrachtet werden. Die gleiche Methode kann auch für alle anderen Moden wiederholt werden.

2. Normalmodenanalyse eines Peptidmoleküls

Die gleiche Analyse, die gerade am Wassermolekül durchgeführt wurde, kann auch für das K2 Peptid getätigt werden. Benutzen Sie die Struktur und Topologie des bereits erstellten K2 Peptids in vacuo und wiederholen Sie die Schritte des vorigen Kapitels.

D. Hauptkomponentenanalyse

Ein häufig durchgeführtes Verfahren ist die Hauptkomponentenanalyse. Sie wird in der Mathematik/Physik auch Hauptachsentransformation genannt, im Rahmen von Simulationen ist der Begriff *essential dynamics* (ED) geläufig. Hierbei wird die kollektive Bewegung der Atome identifiziert und dadurch die unterliegenden atomaren Fluktuationen der Struktur zugänglich gemacht. Diese Fluktuationen sind nach Definition gekoppelt, da eine Schwingung eine andere beeinflussen kann. Diese Korrelation zwischen den Teilchenbewegungen wird hier genutzt um die gesamte Fluktuationsbewegung des Moleküls u.a. (bio)physikalischen Eigenschaften oder Funktionen zuzuordnen. Folglich kann die Untersuchung einer Struktur mit Hilfe einer Simulation wertvolle Einsichten in das Verhalten von Makromolekülen bieten.

Der erste Schritt der ED ist die Konstruktion der sog. Kovarianzmatrix, die ein Maß für das kolineare Verhalten jeder Atompaarbewegung zueinander ist. Diese Matrix wird nachfolgend diagonalisiert, wobei hier die Hauptkomponenten (Eigenvektoren) die kollektive Bewegung der Teilchen beschreiben. Jede Koordinate des Vektors wiederum ist ein Maß für die Stärke der Beteiligung eines Atoms an einer Bewegung. Die Eigenwerte hingegen sind die Summen der Fluktuationen pro Atom und geben somit Aussage über die totale Motilität einer Hauptkomponente, d.h. wie deutlich diese Bewegung im Vergleich mit den anderen Hauptkomponenten ist. In der Regel ist ein Großteil der Bewegung in einem System mit zehn oder weniger Hauptkomponenten beschrieben.

Um die Trajektorie der Simulation des K2 Proteins zu verwenden, müssen wir zunächst die Kovarianzmatrix erstellen und diagonalisieren:

```
gmx covar -s md.tpr -f md.xtc -o eigenvalues.xvg -v eigenvectors.trr -xpma covar.xpm -mwa
```

Der Switch `mwa` stellt sicher, dass die Analyse massengewichtet wird (anstatt von reiner Positionsgewichtung). Die ersten fünf Eigenvektoren beschreiben meist die Hauptbewegungen des Systems, normalerweise über 80% der Gesamtmotilität. Wenn der Wert geringer ist, bedeutet dies, dass nur wenige klar korrelierte Hauptbewegungen existieren. Man kann auch einen genaueren Blick auf den ersten Eigenvektor zu werfen:

```
gmx anaeig -first 1 -last 1 -s md.tpr -f md.xtc -v eigenvectors.trr -eig eigenvalues.xvg -extr ev1.pdb
```

Die Eigenvektoren beschreiben auch die Richtungen der Bewegungen. Der Switch `-extr` extrahiert die Extremastrukturen entlang der Eigenvektoren aus der Trajektorie. Diese können wir uns nun in VMD ansehen (`ev1.pdb`).

E. Ramachandran Plot und Zeitserien für Diederwinkel

Zur Analyse der Struktur und Dynamik können Sie auch bequem die entsprechenden Gromacs Programme benutzen. So berechnen Sie die RMS Abweichung von der Startstruktur

```
gmx rms -s md -f md.xtc
```

(am besten zweimal "Protein-H" auswählen), und so erzeugen Sie den Ramachandran Plot

```
gmx rama -s md -f md.xtc
```

Die resultierenden `.xvg` Dateien sind mit `xmgrace` zu betrachten. In dem Ramachandran Plot können Sie die Symbole ändern und verkleinern, um das Bild deutlicher zu machen.

In welchem Bereich des Ramachandran Plot finden Sie die dominante Konformation? Gibt es vielleicht auch eine andere Konformation, die zwar weniger populiert aber trotzdem noch relativ deutlich ist? Wenn ja, wie wird diese meistens bezeichnet?

Als nächstes werden die Zeitserien der beiden Peptiddiederwinkel φ und ψ aus der Simulation extrahiert. Dazu verwenden wir das Gromacs Programm `gmx angle`. Wie Sie unter `gmx angle -h` nachlesen können, benötigen Sie dazu eine Indexdatei, in der die zu einem Diederwinkel gehörenden Atom-Indices aufgelistet werden. In der Datei `dihedrals.ndx` wurde diese Liste schon begonenn. Vervollständigen Sie sie.

Schauen Sie sich nun die Datei `calc_dihedrals.sh` an. Hier wird `gmx angle` für jeden Diederwinkel in einer Schleife aufgerufen. Führen Sie die Datei mit den erforderlichen Eingabedateien aus.

Plotten Sie die Ausgabedateien und vergleichen Sie die Ergebnisse der `gmx angle` Berechnung mit den Werten von VMD. Berechnen Sie alternativ auch den Winkel über eine Peptidbindung (ω), der zwischen $C\alpha-C-N(+1)-C\alpha(+1)$ gemessen wird.

III. EXTENDED SAMPLING METHODEN

In vielen Situationen ist es viel zu ineffizient, Konfigurationen des Molekülsystems mittels einer gewöhnlichen MD Simulation zu generieren. Das ‘Abtasten’ des Konfigurationsraumes würde einfach viel zu lange dauern – man müsste viel zu lange simulieren, um alle relevanten Konfigurationen (z.B. Konformationen eines Peptids) mit den richtigen Wahrscheinlichkeiten in der Trajektorie zu sehen. Um diesen schwerwiegenden Nachteil umzugehen, wurden sog. Extended Sampling Methoden entwickelt.

A. Wiederholung: Vorbereitung und Simulation des Dipeptids

1. Vorbereitung des Dipeptids

Die Vorbereitung des zu simulierenden Systems wird für beide Methoden gemeinsam sein. Wir werden ein sehr einfaches System untersuchen – ein Dipeptid in wässriger Lösung.¹ Um die Übung interessanter zu machen, möge jede/r Teilnehmer/in eine andere Aminosäure simulieren, so dass jede/r ein anderes Ergebnis bekommen wird. Am Ende werden die Ergebnisse verglichen und diskutiert.

Das Dipeptid kann im `xleap` gebaut werden, mit folgenden Kommandos, wobei das Alanin (`ALA`) mit einer anderen Aminosäure ersetzt werden kann:

```
source leaprc.ff14SB
pep = sequence { ACE ALA NME }
savepdb pep pep.pdb
quit
```

Die resultierende PDB Datei kann anschließend mit Programmen aus dem Gromacs Paket bearbeitet werden, um die Topologie zu bekommen und das System solvatisieren.

Die Vorbereitung des Systems folgt der Übung ID und II, und wird hier nur kurz zusammengefasst:

1. Topologie erstellen.

```
gmx pdb2gmx -f pep.pdb -ignh -o pep.gro
```

Wählen Sie als Kraftfeld wieder `AMBER99SB-ILDN` und `TIP3P` als Wassermodell aus.

2. Kubische Box erstellen und solvatisieren.

```
gmx editconf -f pep.gro -o pep.edit -c -d 1 -bt cubic
gmx solvate -cp pep.edit -o pep.box -cs spc216 -p topol
```

3. Falls eine geladene Aminosäure gewählt wurde – System neutralisieren.

```
echo > dummy.mdp
gmx grompp -f dummy -p topol -c pep.box
gmx genion -s dummy -o pep.ion -p topol -neutral
```

Bei `genion` muss die Wasser enthaltende Gruppe explizit gewählt werden.

Damit ist das zu simulierende System gebaut und die Topologiedatei inklusive Wasser und ggf. Gegenion erstellt.

Weiter muss das System equilibriert werden. Das Verfahren folgt wieder Übung II und besteht aus folgenden Schritten:

¹ Mit ‘Dipeptid’ verstehen wir einen Aminosäurenrest, der mit den Schutzgruppen $\text{CH}_3\text{CO}-$ (Acetyl) und $-\text{NHCH}_3$ (Methylamino) ein vollständiges Molekül darstellt. So ein Dipeptid weist ein komplettes Paar von Winkeln $\varphi - \psi$ auf, und es ist daher sinnvoll sich mit einem Ramachandrandiagramm zu befassen.

1. Kurz energieminimieren. Falls ohne Gegenion simuliert wird (ungeladene Aminosäure), soll `pep.box` anstatt `pep.ion` genommen werden.

```
gmx grompp -f steep -c pep.ion -p topol -o steep
gmx mdrun -deffnm steep
```

2. Auf 300 K aufheizen.

```
gmx grompp -f heat -c steep -p topol -o heat
gmx mdrun -deffnm heat
```

3. Barostat einschalten und Dichte equilibrieren.

```
gmx grompp -f equil -c heat -p topol -o equil
gmx mdrun -deffnm equil
```

2. Simulation des Dipeptids

Nachdem wir unser System nun ausreichend equilibriert haben, starten wir eine MD Simulation, in der wir das Peptid nicht mehr festhalten. Betrachten Sie die dazugehörige Eingabedatei `md_dipeptide.mdp`. Benutzen Sie nun `equil.gro` als Startstruktur und starten Sie die Simulation. Die Simulation wird vorbereitet und gestartet mit den Befehlen

```
gmx grompp -f md -c equil -p topol -o md
gmx mdrun -v -deffnm md
```

B. Metadynamik eines Dipeptids

1. Intro

In einer Metadynamik Simulation wird zu der Potentialenergiefunktion des Molekülsystems (also zu dem üblichen Kraftfeld) eine zusätzliche, ‘künstliche’ Energiefunktion addiert (biasing potential). Diese dient dazu, das Molekül aus dem Energieminimum zu bringen, in dem es sich gerade befindet. Die biasing potential Funktion hat die Form einer Summe von vielen Gaussfunktionen, die erst während der Simulation eine nach der anderen addiert werden. Die Gaussfunktionen nehmen als Argument eine oder einige (einfache oder komplizierte) Funktion(en) der Atomkoordinaten; diese wird/werden Collective Variables (CV) oder Reaktionskoordinate genannt. Beispiele von CV: Diederwinkel φ und ψ in einem Peptid; Abstand eines Liganden von der Mitte der Bindungstasche; Gyrationradius eines Proteinmoleküls.

Eine wichtige Eigenschaft der Metadynamik Methode ist, dass die Summe der Gaussfunktionen zu der negativen freien Energie (ΔF bzw. ΔG) konvergieren, so lange die Simulation länger wird. Die kritische Voraussetzung für Erfolg ist hier die angemessene Wahl der Collective Variables. Eine passende, natürliche Wahl der Reaktionskoordinaten für ein Dipeptid sind die Diederwinkel φ und ψ , so dass man als Ergebnis die freie Energie in Form eines Ramachandrandiagramms bekommen würde.

Das Gromacs Paket an sich kann leider keine Metadynamik Simulation durchführen. Allerdings wurde ein Hilfsprogramm (Plugin) namens Plumed entwickelt, das die erwünschte Funktionalität ergänzt. Praktisch sieht es so aus, dass die Parameter der Metadynamik-Methode durch eine zusätzliche Input-Datei übermittelt werden; wir benutzen hier die Datei `wt-metad.dat`, die folgende Daten enthalten soll:²

```
phi: TORSION ATOMS=5,7,9,15
psi: TORSION ATOMS=7,9,15,17
METAD ...
  LABEL=metad
```

² Zeilen, die mit # beginnen, sind Kommentare und werden vom Programm nicht berücksichtigt.

```

ARG=phi,psi
PACE=100
HEIGHT=0.625
SIGMA=0.349,0.349
GRID_MIN=-pi,-pi
GRID_MAX=pi,pi
FILE=HILLS
BIASFACTOR=7.
TEMP=300.
... METAD
PRINT STRIDE=100 ARG=phi,psi,metad.bias FILE=plumed.svg

```

Hier werden also zuerst die Diederwinkel φ und ψ durch Atomnummern der Atome C, N, C α , C und N definiert. Dann wird schon die Metadynamik aktiviert: Es werden zusätzliche Gaussiane als Funktionen von φ und ψ addiert, und zwar in jedem hundertsten Schritt der MD, und die Höhe w und Breite σ der Gaussiane

$$w \cdot \exp \left[-\frac{(\varphi(t) - \varphi^*)^2 + (\psi(t) - \psi^*)^2}{2\sigma^2} \right]$$

betragen 0.625 kJ/mol und 0.349 rad = 20°. Die letzteren beiden Optionen bestellen die sog. well-tempered Variante der Metadynamik: Hier werden die zusätzlichen Gaussiane mit der Zeit kleiner und kleiner (w wird automatisch reduziert), damit eine bessere Konvergenz der freien Energie gewährleistet wird. Der genaue Wert des Biasfaktors soll dem zu simulierenden System angepasst werden, und es kann eine gute Idee sein, mehrere Werte auszuprobieren. Eine Faustregel empfiehlt allerdings einen Wert von der Hälfte der erwarteten Energiebarriere in kT-Einheiten (1 kT beträgt 2.5 kJ/mol bei 300 K). Die letzte Zeile bestimmt, welche Daten während der Simulation ausgeschrieben werden sollen.

2. Durchführung

Zuerst soll eine Gromacs Simulation vorbereitet werden, genau gleich wie für eine übliche MD Simulation:

```
grompp -f md -c equil -p topol -o md-meta
```

Danach kann schon die eigentliche Metadynamik-Simulation durchgeführt werden:

```
mdrun -deffnm md-meta -plumed wt-metad -v
```

Hier wäre es gut, die Anzahl der Schritte in `md.mdp` so zu wählen, dass die Simulation über Nacht laufen wird und am nächsten Tag früh morgens fertig ist.

C. Analyse

1. Freie Energie der unbiased Simulation

Im ersten Schritt werden wir die Dihedralwinkel der Simulation analysieren:

```
gmx rama -s md.tpr -f md.xtc -xvg none
```

Hierbei erhalten wir, wie erwartet, einen Ramachandran-Plot `rama.svg` als Ausgabe. Daraus erstellen wir wiederum ein Histogramm mit dem Programm `make_histogram.py`. Um nun die Potentialfläche zu erhalten, führen wir `rama-histo-to-png.sh` aus, welches eine PNG Bilddatei erstellt. Ansehen kann man diese, indem man `rama-kcal.png` öffnet, wobei die Energien in kcal/mol dargestellt sind.

Im nächsten Schritt wollen wir die gleiche Analyse aus der Metadynamik erstellen und dann beide Plots vergleichen. Achten Sie darauf, ob alle Minima in der freien Simulation sichtbar sind und ob die Barrieren und Minima in beiden Simulationen die gleichen Höhen besitzen.

2. Freie Energie der Metadynamik

Zur Analyse wird die Datei `HILLS` gebraucht, wo die Position sowie Höhe der Gaussiane ausgeschrieben wird. Wie oben erwähnt, entspricht die Summe aller Gaussiane einem Spiegelbild der freien Energie als Funktion der verwendeten Collective Variables. Hier bekommt man die freie Energie also als Funktion der Diederwinkel $\varphi - \psi$ (Ramachandrandiagramm), und zwar mit dem folgenden Befehl:

```
plumed sum_hills --bin 180,180 --min -pi,-pi --max pi,pi --hills HILLS
```

Die resultierende Landschaft der freien Energie, die in die Datei `fes.dat` ausgeschrieben wurde, kann man am besten grafisch darstellen. Ein Bild im PNG-Format (`fes-kcal.png`) wird mit dem gelieferten Skript erzeugt, das einfach mit dem Befehl `./fes-dat-to-png.sh` ausgeführt wird. Die freie Energie in kcal/mol ist farbkodiert.

Vergleichen Sie nun den Plot der Metadynamik mit dem der gewöhnlichen Simulationen. Was fällt besonders auf? Welche Konformationen sehen Sie in den Ramachandrandiagrammen? Vergleichen Sie die Diagramme für die verschiedenen Aminosäuren.

IV. QM/MM SIMULATION

Die offensichtlichste Limitierung der empirischen Kraftfelder (MM) liegt darin, dass chemische Reaktionen nicht beschrieben werden können, also Prozesse wo kovalente Bindungen gebrochen werden oder entstehen. Dies kann man mit der Anwendung eines kombinierten Verfahrens umgehen, in dem man die Region, wo Chemie passiert, mit einer quantenchemischen Methode beschreibt, und den großen Rest des Systems berechnet man mit MM, wie sonst üblich. Diese hybriden Schemata werden QM/MM genannt, und der Nobelpreis für Chemie 2013 wurde zum Teil für die Entwicklung dieser Methoden vergeben.

In dieser Übung werden wir eine kleine chemische Reaktion in einem kleinen Molekül untersuchen, nämlich den Protonentransfer in einem Malonaldehydmolekül, siehe Bild 3. Der Vorteil ist dabei, dass die Simulation schnell laufen wird. Dies wird noch mehr beschleunigt, wenn eine effiziente QM Methode eingesetzt wird – hier wird es die semi-empirische Dichtefunktionaltheorie-Methode DFTB3 sein. Die technischen Details zu DFTB3 können Sie nachschauen, sie werden allerdings für die Durchführung sowie Analyse der Simulationen nicht dringend benötigt. DFTB3 ist ein Bestandteil von einer lokalen Version von Gromacs, es wird also kein externes Programm benötigt, um die quantenchemischen Berechnungen durchzuführen.

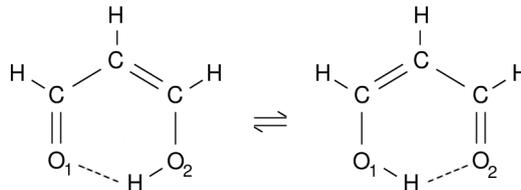


FIG. 3: Malonaldehyd – intramolekularer Protonentransfer.

A. Equilibrierung mit MM

Zuerst soll das Molekülsystem, ein solvatisiertes Malonaldehydmolekül, mithilfe eines üblichen Kraftfeldes, equilibriert werden. Dazu können Sie die gelieferte Topologie benutzen (Datei `mal.top`), sowie die Struktur in `mal.gro`, die Sie noch in eine kubische Wasserbox packen sollen. Die benötigten `.mdp` Dateien können aus dem Kapitel III genommen werden. Der Ablauf wird schematisch wie folgt aussehen:

```
gmx editconf ...
gmx solvate ...
gmx grompp -f steep ...
gmx mdrun -deffnm steep ...
gmx grompp -f heat ...
gmx mdrun -deffnm heat
gmx grompp -f equil ...
gmx mdrun -deffnm equil
```

wobei Simulationslänge recht kurz (unter 100 ps) gehalten werden kann.

B. Vorbereitung

Die wichtigste Frage in jeder QM/MM Simulation ist, wie man das Molekülsystem zwischen QM und MM Regionen teilt. Hier ist es sehr einfach: Ein Malonaldehydmolekül ist so klein, dass es ganz mit QM beschrieben werden kann, und das wässrige Lösungsmittel wird dann mit dem MM-Kraftfeld berechnet. Für eine QM/MM Simulation muss dann auch die Topologie des Moleküls modifiziert werden: Es wird keine Energie innerhalb der QM-Region mehr gerechnet, deshalb müssen alle Kraftfeldterme, die die QM-Region beschreiben, wegfallen.

Kopieren Sie die bestehende Topologie unter einem neuen Namen `mal-qmmm.top`. Bearbeiten Sie dann diese Datei mit einem Texteditor (z.B. `gedit`) wie folgt: Löschen Sie die Sektionen *angles*, *dihedrals* sowie *pairs*. Desweiteren ändern Sie den Typ von jeder Bindung von 1 auf 5, und löschen Sie die Parameter. Die Sektion sieht dann wie folgt aus:

```
[ bonds ]
; ai  aj  funct      b0      kb
  1   2    5
  2   4    5
...
```

Damit ist die Topologie vorbereitet. Es werden gar keine intramolekularen Wechselwirkungen innerhalb des Malonaldehydmoleküls mit dem Kraftfeld berechnet. Die Ladungen der QM-Atome können in der Topologiedatei bleiben, denn sie werden von Gromacs automatisch auf Null gesetzt. Die Wechselwirkungen zwischen der QM und MM Regionen werden geteilt berechnet: die QM-Methode berücksichtigt die elektrostatische (Ladung-Ladung) WW,³ und das MM-Programm wendet das übliche Lennard-Jones Potential an; deshalb müssen auch den QM-Atomen Atomtypen zugewiesen werden, und LJ-Parameter geliefert werden.

C. Ausführung

Die `.mdp` Datei wird nun einige weitere Optionen enthalten, die zum Großteil das Verhalten der quantenchemischen Methode DFTB3 steuern. Diese sind die folgenden, siehe auch die Datei `long.mdp`:

```
QMMM                = yes
QMMM-grps           = MAL
QMmethod            = RHF
QMMMscheme          = normal
QMbasis             = STO-3G
QMcharge            = 0
QMmult              =
MMChargeScaleFactor = 1

QMdftb-sccmode      = 3
QMdftb-telec        = 10.
QMdftb-slko-path    = /usr/local/run/gromacs-5.0-dftb-v6a-plumed/share/gromacs/top/dftb/3ob/
QMdftb-slko-separator =
QMdftb-slko-lowercase = yes
QMdftb-slko-suffix  = -c.spl
QMdftb-partial-pme  = 1
QMdftb-dispersion    = 1
QMdftb-cdco         = 0
```

Zuerst wird festgelegt, dass eine QM/MM Simulation laufen soll, und die QM Region definiert. ‘MAL’ bezieht sich dabei auf den Namen einer Gruppe in der Indexdatei (hier `index.ndx`), die also mit

```
make_ndx -f equil
```

erzeugt werden muss und dem `grompp_d` anschließend mitgegeben. Die weiteren drei Optionen werden ignoriert (Vorsicht, es wird keine RHF/STO-3G Berechnung gemacht). Wichtig kann allerdings sein, die Ladung (sowie Spinmultiplizität) der QM-Region anzugeben. Die verbleibenden Optionen sind dann wirklich DFTB3-spezifisch, und müssen nicht im Detail verstanden werden. Es ist bloß empfehlenswert zu prüfen, ob der Ordner mit DFTB3 Parametern existiert (`QMdftb-slko-path`).

Genauso wie bei reinen MM-Simulationen muss erst der Gromacs Preprozessor aufgerufen werden, gefolgt von dem eigentlichen Simulieren. Ein wichtiger Unterschied ist hier, dass die double-precision Versionen von allen Gromacs Programmen zu benutzen sind. Dies ist einfach – zu jedem Programmname hängen Sie einfach den Suffix `_d` an:

```
gmxd gmpp -f long -c equil -p mal-qmmm -n index -o long
gmxd mdrun -deffnm long -nt 1 -v > mdrun.out
```

Dabei ist zu beachten, dass die QM/MM-Topologie benutzt wird. Die freie QM/MM Simulation wird eine etwas längere Zeit in Anspruch nehmen, ca. 40 min für eine Simulation von 100 ps.

³ Die Ladungen der QM Atome werden mithilfe der Mulliken Analyse in der quantenchemischen Berechnung ermittelt.

D. Auswertung

Betrachten Sie zuerst die Trajektorie visuell (`vmd equil.gro long.xtc`). Sie können das Wasser ausblenden (*Graphics – Representations* and ‘not water’ für Selected Atoms), oder direkt die `.xtc` Datei betrachten (`vmd mal.gro long.xtc`). Sehen Sie einen Protonentransfer? Wenn die originale O–H Bindung ungewöhnlich lang wird, kann es bedeuten, dass diese gebrochen ist.

Um dies korrekt beobachten zu können, muss Visualisierung angepasst werden. Stellen Sie in Ihrer Repräsentation auf *DynamicBonds* (unter *Drawing Method*) und erzeugen Sie eine neue Darstellung mit *Create Rep.* Als *Bond Radius* bietet sich der Wert 0.2 an. Hier wählen Sie *VDW* (Van der Waals Kugeln) und tragen bei der Option *Sphere Scaling* einen kleinen Wert ein (z.B. 0.2). Somit kann man sich eine dynamische *ball and stick* Visualisierung in VMD erreichen.

Um den Protonentransferprozess quantitativ zu verfolgen, sollen wir nun eine passende Reaktionskoordinate wählen – also eine Größe, die wir aus den Koordinaten der Atome berechnen können, und die die Reaktion möglichst ungestört von anderen Einflüssen beschreibt. Bei so einem einfachen Protonentransfer eignet sich sehr gut die Differenz der beiden O–H Abstände.

Die Aufgabe ist nun diese Abstände zu messen und ihre Differenz auszurechnen, das Ganze entlang der Trajektorie aus der QM/MM Simulation. Auch wenn Gromacs selbst natürlich schon Abstände von Atomen messen kann, kann es hier vorteilhaft sein ein externes Tool anzuwenden, nämlich das Plumed Programm. Einer der Zwecke von Plumed ist nämlich das Arbeiten mit verschiedenen Reaktionskoordinaten zu erleichtern.

Erstellen Sie hierfür mittels eines Texteditors eine Eingabedatei für Plumed unter dem Namen `diffdist.dat`:

```
d1: DISTANCE ATOMS=1,9
d2: DISTANCE ATOMS=8,9
d: COMBINE ARG=d1,d2 COEFFICIENTS=1,-1 PERIODIC=NO
PRINT ARG=d FILE=diffdist.svg
```

Inhaltlich selbsterklärend, werden hier beide Abstände definiert, dann ihre Differenz, die dann gemessen und in eine Ausgabedatei gedruckt wird. Sie führen den Vorgang wie folgt durch:

```
plumed driver --mf_xtc long.xtc --plumed diffdist.dat --timestep 0.01
```

Die resultierende Datei können sie mit `xmgrace diffdist.svg` betrachten. Was sehen Sie?

In einer QM/MM Simulation kann uns auch die zeitliche Entwicklung der Elektronendichte interessieren. Diese wird im Rahmen der DFTB Methode durch Partiaalladungen auf den einzelnen QM Atomen dargestellt. Unser Gromacs Programm speichert die Atomladungen in der Datei `qm_dftb_charges.svg`, die Sie mit `xmgrace -nxy` betrachten können. Hier ist auf der *x*-Achse Zeit in ps, und auf der *y*-Achse die Ladungen; jedem QM-Atom entspricht eine Kurve. Können Sie aus dem Verlauf der Ladungen auf Zusammenhang mit dem Protonentransfer schließen?

Oft interessiert uns nicht so ganz der Verlauf einer Größe, sondern eher die Wahrscheinlichkeit, mit der verschiedene Werte eingenommen werden. Um eine Wahrscheinlichkeitsdichte der Differenz der O–H Abstände zu bilden, benutzen Sie eins der Gromacs-Analysprogramme:

```
gmx analyze -f diffdist -dist diffdist-histo -bw 0.005
```

Das Programm meldet den Mittelwert sowie seine Standardabweichung, und gibt das Histogramm in einer neuen Datei aus. Was für einen Mittelwert, und was für eine Form des Histogramms würden Sie erwarten, vorausgesetzt wäre eine ausreichend lange Simulationsdauer? Wird Ihre Erwartung erfüllt?

E. Metadynamik

Um konvergierte freie Energien zu erhalten, kann es notwendig sein lange Simulationen durchzuführen. Das mag insbesondere mit QM/MM Simulationen unmöglich lange Rechenzeiten in Anspruch nehmen. Dieses Problem lässt sich auch hier durch die Anwendung der Extended-Sampling Techniken lösen. Hier werden wir die Protonentransfer-energetik mit Metadynamik untersuchen.

Dazu können Sie die bestehende `.tpr` Datei benutzen, die Sie gerade für die freie Simulation vorbereitet hatten. Was zusätzlich gebraucht wird, ist eine Eingabedatei für Plumed (`wt-metad.dat`), in der eine Metadynamiksimulation aufgerufen wird. Diese kann wie folgt aussehen (zu vergleichen mit der Eingabe für das Dipeptid in Sektion III B), wo wieder die well-tempered Variante der Metadynamik gemacht werden soll:

```
d1: DISTANCE ATOMS=1,9
d2: DISTANCE ATOMS=8,9
d: COMBINE ARG=d1,d2 COEFFICIENTS=1,-1 PERIODIC=NO
METAD ...
  LABEL=metad
  ARG=d
  PACE=100
  HEIGHT=0.625
  SIGMA=0.02
  FILE=HILLS
  BIASFACTOR=5.
  TEMP=300.
... METAD
PRINT STRIDE=100 ARG=d,metad.bias FILE=plumed.xvg
```

Sie führen den Vorgang wie folgt durch:

```
gmx_d mdrun -deffnm long -plumed plumed.dat -nt 1 -v > meta.out
```

Sobald die Simulation fertig ist, oder sogar noch während sie läuft, kann die HILLS Datei analysiert werden, um die Kurve der freien Energie (ΔG) zu konstruieren. Wie vorher, kann dies mit dem Plumed Programm gemacht werden:

```
plumed sum_hills --hills HILLS
```

Die resultierende freie Energie in der Datei `fes.dat` kann mit Xmgrace betrachtet werden. Stimmt die Kurve mit der aus der freien Simulation überein? Weist die Kurve die erwartete Symmetrie auf? Wie hoch ist die Barriere? Wenn wir die Protonentransferrate aus der Barrierenhöhe berechnen möchten, würden wir die Rate möglicherweise unterschätzen. Warum?