

Proteinlerin Moleküler Dinamik Simölasyonları

Doç. Dr. Mustafa TEKPINAR

2025-10-13

İçindekiler

1	Önsöz	1
2	Teorik Temeller	3
2.1	Bu Bölümde Neler Öğreneceğiz?	3
2.2	Moleküler Dinamik Simülasyonları Nedir?	3
2.3	Tarihsel Arka Plan	4
2.4	Temel Mantık	4
2.5	Güçlü ve Zayıf Yönler	7
2.6	Neden Önemli?	7
2.7	Sonuç	8
2.8	Bölüm Özeti	8
3	Gromacs ile MD Simülasyonlarına Hazırlık	9
3.1	Bu Bölümde Neler Öğreneceğiz?	9
3.2	Gromacs programının kurulması	9
3.3	Sistemin inşa edilmesi	10
3.3.1	Proteinin hazırlanması	10
3.3.2	Sisteme solvent eklenmesi	11
3.3.3	Sisteme doğal ortam iyonlarının eklenmesi	12
3.4	Bölüm Özeti	12
4	MD Simülasyonlarını Çalıştırma	13
4.1	Bu Bölümde Neler Öğreneceğiz?	13
4.2	Enerji minimizasyonu	13
4.3	Dengeleme simülasyonları	13
4.4	Üretim simülasyonları	14

İçindekiler

4.5	Yüksek başarılı hesaplama sistemleri	15
4.6	Bölüm Özeti	15
5	MD Simülasyon Analizleri	17
5.1	Bu Bölümde Neler Öğreneceğiz?	17
5.2	Yapısal Analizler	17
5.3	RMSD (Root Mean Square Deviation)	17
5.3.1	RMSD'nin Anlamı	18
5.3.2	Referans Yapı Seçimi	19
5.4	RMSF (Root Mean Squared Fluctuations):	19
5.5	Rg (Radius of gyration):	19
5.6	SASA (Solvent Accessible Surface Area):	19
5.7	İkincil Yapı	19
5.8	Enerji Analizleri	19
5.9	Bölüm Özeti	19
6	Ek 1: Amino Asitler ve Proteinler	21
6.1	Giriş	21
6.2	Amino Asitler	21
6.3	Proteinlerin Yapı Düzeyleri	22
6.4	Protein Yapısı Belirleme Yöntemleri	23
6.4.1	DeneySEL Yöntemler	24
6.4.2	Teorik Yöntemler	25
6.5	Protein Veri Tabanları	26
6.6	Özet	26
7	Ek 2: Protein Moleküler Dinamik Simülasyonlarında Kuvvet Alanları	27
7.1	Giriş	27
7.2	Kuvvet Alanlarının Genel Yapısı	27
7.2.1	Bağ (Bond) terimi	28
7.2.2	Açı (Angle) terimi	28
7.2.3	Dihedral (Torsiyon) terimi	28
7.2.4	Bağırsız (Non-bonded) etkileşimler	29

7.3	Protein Simülasyonlarında Kullanılan Başlıca Kuvvet Alanları	29
7.3.1	AMBER	29
7.3.2	CHARMM	30
7.3.3	OPLS	30
7.3.4	GROMOS	30
7.3.5	Yeni Nesil Kuvvet Alanları	30
7.4	Kuvvet Alanı Seçiminde Dikkat Edilecek Noktalar	30
7.5	Sık Yapılan Hatalar ve Öneriler	31
7.6	Özet	31
8	Ek 1: Bash Komut Satırı	33
8.1	Giriş	33
8.2	Dosya ve Dizin İşlemleri	33
8.3	Dosya Görüntüleme	34
8.4	Kopyalama ve Taşıma	34
8.5	Arama ve Filtreleme	34
8.6	Süreç ve Kaynak Yönetimi	35
8.7	Dosya Aktarımı	35
8.8	Program Kurma	35
8.9	Alıştırmalar	36
8.10	Sonuç	37
9	Kaynaklar	39
9.1	Ders Kitapları	39
9.2	Derleme Makaleler	39

1 Önsöz

Moleküler dinamik simülasyonları fizik, kimya, moleküler biyoloji, eczacılık, tıp ve biyomühendislik gibi çeşitli alanlarda kullanılan bilişimsel bir araştırma tekniğidir. Bu tekniğin geniş bir uygulama alanı olmasında rağmen bu alandaki Türkçe kaynak sayısı oldukça azdır. Bu nedenle, bu kitapta proteinlerin moleküler dinamik (MD) simülasyonlarını pratik bir yaklaşımla ele alacağız. Kitapta ağırlıklı olarak MD simülasyonlarının açık kaynak kodlu Gromacs paketi ile yapılması hedeflenmektedir. Kitap, kısa bir teorik temeller bölümü ile başlayacaktır. Teorik temeller bölümünden sonra MD simülasyonlarına hazırlık aşamaları anlatılacaktır. Son bölümde ise MD simülasyonu analiz teknikleri sunulacaktır. Kitabın teorik bilgilerle dolu bir kitap olmasından ziyade pratik bir uygulama kitabı olması hedeflenmektedir. Kitabın, bu alanda araştırma yapmak isteyen ancak yabancı dildeki kaynaklardan henüz tam yararlanamayan ileri lisans öğrencileri ve lisansüstü öğrenciler için faydalı olacağını ümit ediyorum.

Kitabı PDF olarak indir: Proteinlerin Moleküler Dinamik Simülasyonları (PDF)

Başlamak için Teorik Temeller bölümüne göz atın.

2 Teorik Temeller

2.1 Bu Bölümde Neler Öğreneceğiz?

Bu bölümde moleküler dinamik simülasyonlarının temel kavramlarını ele alacağız.

2.2 Moleküler Dinamik Simülasyonları Nedir?

Moleküler dinamik (MD) simülasyonları, atomik ölçekteki sistemlerin zaman içindeki hareketlerini bilgisayar aracılığıyla incelemeye olanak sağlayan güçlü bir yöntemdir. Temel fikir, doğada var olan parçacıkları sayısal olarak temsil etmek ve onların birbirleriyle etkileşimlerini bilgisayar üzerinde adım adım takip etmektir. Böylece laboratuvarda gözlenmesi zor veya imkânsız olan süreçler sanal ortamda canlandırılabilir.

MD simülasyonlarını cazip kılan noktalardan biri, karmaşık sistemlerin dinamiklerini doğrudan *izleyebilmemizdir*. Deneysel yöntemler genellikle bu süreçleri anlık görüntüler hâlinde sunarken, MD simülasyonları bize tüm zaman çizgisini ayrıntılı olarak verir.

2.3 Tarihsel Arka Plan

Moleküler dinamik yöntemleri ilk olarak 1950'lerin sonunda, bilgisayarların henüz oldukça sınırlı kapasitelere sahip olduğu dönemde geliştirilmiştir. Başlangıçta birkaç düzine atomdan oluşan sistemler üzerinde uygulanan bu teknik, bilgisayar gücündeki olağanüstü gelişmeler sayesinde günümüzde milyonlarca atom içeren biyolojik komplekslere kadar genişletilebilmiştir.

Bu ilerleme yalnızca hesaplama gücündeki artışla değil, aynı zamanda kuvvet alanlarının, algoritmaların ve yazılımların gelişmesiyle de mümkün olmuştur. Bugün MD, ilaç tasarımıyla nanoteknolojiye kadar pek çok alanda vazgeçilmez bir araçtır.

2.4 Temel Mantık

Bir MD simülasyonu, kabaca üç aşamadan oluşur:

1. **Başlangıç Modelinin Hazırlanması**

Çalışılmak istenen sistemin atomik koordinatları tanımlanır. Bu bir proteinin yapısı, bir polimer zinciri ya da basit bir sıvı karışımı olabilir.

2. **Sistemin Enerjisinin Belirlenmesi**

Atomlar arası etkileşimleri tanımlayan “kuvvet alanı” seçilir. Kuvvet alanı atomlar arası etkileşimlerin hesaplanabilmesi için ihtiyaç duyacağımız parametreler setidir. Bu parametre seti kullanılarak, öncelikle sistemin toplam enerjisi hesaplanır.

3. **Etki Eden Kuvvetlerin Belirlenmesi**

Korunumlu kuvvetlerin olduğu bir sistemde enerji bilinirse her atom üzerine etki eden kuvvetler hesaplanabilir.

4. Etki Eden İvmelerin Belirlenmesi

Eğer bir atoma etkiyen net kuvvet bilinirse $F=ma$ denklemi ile ona etkiye ivme hesaplanabilir.

5. Zamanın Çok Küçük Adımlarla İlerletilmesi

İvme bilinirse hızı hesaplamak sadece temel fizik denklemlerinin uygulanmasıdır. Hız kullanılarak küçük bir zaman dilimi sonrasındaki (dt: genellikle bir kaç femtosaniye) konumlar hesaplanabilir. Konumlar güncellendiği için sistemin enerjisi artık değişmiştir. Bu durumda tekrar 2. adıma gidip bu döngüyü tekrarlamak gerekir. Gündelik hayatta SI birim sistemine göre en küçük zaman birimi saniyedir. Ancak MD simülasyonlarında femtosaniye zaman birimi yaygın olarak kullanılır. Femtosaniye, pikosaniye, nanosaniye, mikrosaniye, milisaniye ve saniye zaman birimleri arasındaki ilişki aşağıdaki tabloda gösterilmiştir.

Aşağıdaki tablo, saniyenin alt katmanlarındaki birimlerin birbirine ve saniyeye göre büyüklüklerini göstermektedir.

Birim Adı	Sembol	1 Birimin Saniye Cinsinden Değeri	1 Saniyede Kaç Tane	Açıklama
Femtosaniye	fs	10^{-15} s	1 000 000 000 000 000	1 saniyenin katrilyonda biri
Pikosaniye	ps	10^{-12} s	1 000 000 000 000	1 saniyenin trilyonda biri
Nanosaniye	ns	10^{-9} s	1 000 000 000	1 saniyenin milyarda biri

2 Teorik Temeller

Birim Adı	Sembol	1 Birimin Saniye Cinsinden Değeri	1 Saniyede Kaç Tane	Açıklama
Mikrosaniye	μs	10^{-6} s	1 000 000	1 saniyenin milyonda biri
Milisaniye	ms	10^{-3} s	1 000	1 saniyenin binde biri
Saniye	s	1 s	1	SI temel zaman birimi

Dönüşüm Örnekleri

- $1 \text{ ns} = 1000 \text{ ps}$
- $1 \mu\text{s} = 1000 \text{ ns}$
- $1 \text{ ms} = 1000 \mu\text{s}$
- $1 \text{ s} = 1000 \text{ ms}$

$\text{ps}["\text{Pikosaniye } (10^{-12})"] \rightarrow \text{ns}["\text{Nanosaniye } (10^{-9})"] \rightarrow$
 $\mu\text{s}["\text{Mikrosaniye } (10^{-6})"] \rightarrow \text{ms}["\text{Milisaniye } (10^{-3})"] \rightarrow$
 $\text{s}["\text{Saniye } (1)"]$
 $\text{---} \rightarrow$

2.5 Güçlü ve Zayıf Yönler

Bu adımların yinelenmesiyle sistemin dinamikleri elde edilir. Ortaya çıkan veriler, sonrasında analiz edilerek anlamlı bilimsel sonuçlara dönüştürülür.

2.5 Güçlü ve Zayıf Yönler

Güçlü Yanlar:

- Atomik çözünürlükte ayrıntı sağlar.
- Deneylerde gözlenemeyen süreçler gözlenebilir.
- Koşullar kolayca değiştirilebilir (sıcaklık, basınç, iyon derişimi).

Zayıf Yanlar:

- Erişilebilen zaman ölçeğı sınırlıdır (mikro–milisaniyeler).
 - Kuvvet alanları belirli yaklaşımlara dayanır, kuantum mekaniksel fiziğı tam olarak yansıtmayabilir.
-

2.6 Neden Önemli?

Moleküler dinamik simülasyonlarının önemi, deneysel verilerle **tamamlayıcı** bir ilişki kurabilmesinde yatmaktadır.

- Deneyler bize makroskopik gözlemler sunarken, MD simülasyonları bu gözlemlerin ardındaki atomik ayrıntıları ortaya çıkarır.

2 Teorik Temeller

- Protein kristal yapıları bize “donmuş” bir görüntü verir; MD ise bu yapının esnekliğini ve hareketliliğini gösterir.
 - Malzeme testleri bir polimerin dayanıklılığını ölçebilir; MD ise polimer zincirlerinin kırılma mekanizmasını açıklayabilir.
-

2.7 Sonuç

Moleküler dinamik simülasyonları, modern bilimin en güçlü mikroskoplarından biri gibidir: Atomların dünyasına benzersiz bir pencere açar. Elbette her mikroskop gibi sınırlamaları vardır; ancak doğru kullanıldığında hem biyolojide hem de malzeme biliminde olağanüstü katkılar sağlar.

2.8 Bölüm Özeti

3 Gromacs ile MD Simülasyonlarına Hazırlık

3.1 Bu Bölümde Neler Öğreneceğiz?

Bu bölümde moleküler dinamik simülasyonlarının temel aşamalarını ele alacağız. Kabaca bir moleküler dinamik simülasyonunun hazırlık aşamaları aşağıdaki basamakları içerir:

1. Sistemin inşa edilmesi
2. Sistemin enerjisinin minimize edilmesi
3. Sistemin dengelenmesi (equilibration)
4. Üretim aşaması

Bu basamakları teker teker ele alalım.

3.2 Gromacs programının kurulması

Gromacs açık kaynak kodlu bir yazılımdır ve şu adresten indirilebilir:
<https://manual.gromacs.org/current/download.html>

Eğer Ubuntu Linux kullanıyorsanız aşağıdaki komut Gromacs'ı sizin için kuracaktır:

```
sudo apt install gromacs
```

3.3 Sistemin inşa edilmesi

Sistemin doğru bir biçimde inşa edilmesi için adım adım ilerlemek gerekir:

1. Proteinin hazırlanması 3. Sisteme solvent eklenmesi 4. Sisteme doğal ortam iyonlarının eklenmesi

3.3.1 Proteinin hazırlanması

Hedef protein yapısının bulunması, temizlenmesi ve eksik (ise) hidrojenlerin eklenmesi ve doğru protonasyon durumlarının belirlenmesi için aşağıdaki adımlar uygulanır:

```
initial_pdb=AF-P22303-F1-model_v4

file_prefix=aces_human_monomer_wt

force_field=amber99sb-ildn

#Call pdb2pqr to determine protonation states of the residues
#according to pH 7.4
pdb2pqr30 --ff PARSE --ffout AMBER --titration-state-method\
propka --with-ph 7.4 ${initial_pdb}.pdb \
${initial_pdb}-pwb2pqr30.pqr \
--pdb-output ${initial_pdb}-pwb2pqr30.pdb

#Add hydrogens if absent, select water model and force field
gmx pdb2gmx -f ${initial_pdb}-pwb2pqr30.pdb \
-o ${file_prefix}_processed.pdb \
-water ${solvent_model} -ff ${force_field} -ignh
```


3.3.2 Sisteme solvent eklenmesi

Proteinler genellikle su ortamında işlevlerini yerine getirir. Bu nedenle bir simülasyon yaparken sisteme mutlaka su eklenmesi gerekir. Aşağıdaki bash komutları ile sisteme su ekleyebiliriz.

```
buffer=1.1

#Define box properties
gmj editconf -f ${file_prefix}_processed.pdb \
  -o ${file_prefix}_newbox.pdb \
  -bt octahedron -c -d ${buffer} -princ

#Note that if spc, spce, or tip3p model is used, solvent_model=spc216
solvent_model=tip3p

#Produce box
if [ "$solvent_model" == "tip3p" ] || \
  [ "$solvent_model" == "spc" ] || \
  [ "$solvent_model" == "spce" ];
then
  gmj solvate -cp ${file_prefix}_newbox.pdb -cs spc216.gro \
    -o ${file_prefix}_solv.pdb -p topol.top
fi

if [ "$solvent_model" == "tip4p" ];
then
  gmj solvate -cp ${file_prefix}_newbox.pdb -cs tip4p.gro \
    -o ${file_prefix}_solv.pdb -p topol.top
fi
```

3.3.3 Sisteme doğal ortam iyonlarının eklenmesi

Aşağıdaki komut sisteme Na ve Cl iyonlarını ekleyip sistemin toplam yükünü sıfır yapacaktır. Sisteme iyonlar eklenirken 0.15 Molar konsantrasyon da sağlanmaya çalışılacaktır.

```
#Prepare for ion addition
gmx grompp -f ions2.mdp -c ${file_prefix}_solv.pdb -maxwarn 2 \
    -p topol.top -o ions.tpr

#Produce and place ions
gmx genion -s ions.tpr -o ${file_prefix}_solv_ions.pdb \
    -p topol.top -pname NA -nname CL -neutral -conc 0.15

#Good news: Preparation is over!
#Bad news : Trouble has just started!
```

3.4 Bölüm Özeti

4 MD Simülasyonlarını Çalıştırma

4.1 Bu Bölümde Neler Öğreneceğiz?

Bu bölümde moleküler dinamik simülasyonları çalıştırmanın temel aşamalarını ele alacağız. Tüm çalıştırma aşamalarından önce grompp (gromacs preprocessor) programını çalıştırıp tpr uzantılı bir dosya elde etmek gerekir.

4.2 Enerji minimizasyonu

Hazırladığımız sistemde birbirine aşırı yakın parçacıklar olabilir. Bunların birbirinden uzaklaştırılması ve parçacıklarının belirli bir denge konumuna getirilmesi gerekir. Bunun için sistemin toplam enerjisinin minimize edilmesi gerekir. Bu aşamada parçacıklara herhangi bir hız atanmadığı akılda tutulmalıdır.

```
gmx grompp -f minim.mdp -c protein.pdb -p topologyFile.top -o em.tpr  
gmx mdrun -v -deffnm em
```

4.3 Dengeleme simülasyonları

Dengeleme simülasyonları Gromacs içerisinde iki aşamada yapılır: 1. NVT dengeleme: Bu aşamada sistemin parçacık sayısı (N), hacmi (V) ve sıcaklığı

4 MD Simülasyonlarını Çalıştırma

(T) sabit tutulur. Burada, önceki aşamanın sonunda üretilen em.gro dosyasını input olarak kullanacağız. Minimizasyon dosyasına ek olarak NVT dengelemesi sırasında Gromacs tarafından kullanılacak olan nvt.mdp input dosyasına da ihtiyacımız olacaktır.

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

2. NPT dengeleme: Bu aşamada sistemin parçacık sayısı (N), basıncı (P) ve sıcaklığı (T) sabit tutulur.

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

4.4 Üretim simülasyonları

MD simülasyonunun son aşaması olan üretim simülasyonları, genellikle analizini yapacağımız yörünge ve enerji dosyalarının üretildiği aşamadır. Aslında NPT simülasyonları ile çok benzer parametrelere sahiptir ancak NPT simülasyonlarına göre çok daha uzundur. Aşağıdaki komut üretim simülasyonlarını başlatacaktır.

```
gmx grompp -f run.mdp -c npt.gro -t npt.cpt -p topologyFile.top -o prod-sim1.tpr  
gmx mdrun -v -deffnm prod-sim1
```

Eğer üretim aşaması simülasyonunuz herhangi bir sebeple yarıda kesilirse, aşağıdaki komut ile kaldığı yerden devam edebilirsiniz.

4.5 Yüksek başarılı hesaplama sistemleri

```
gmx mdrun -v -s prod-sim1.tpr -cpi prod-sim1.cpt -deffnm prod-sim1
```

MD simülasyonunuz bittikten sonra simülasyon süresinin uzatılması gerektiğini farkedebiliriz. Bu durumda Gromacs içindeki *extend* parametresinden yararlanabiliriz. Aşağıdaki komut simülasyonu 75 nanosaniye (75000 pikosaniye) uzatmak için yeni bir tpr dosyası oluşturacaktır. Bu yeni tpr kullanılarak simülasyon, kaldığı yerden devam ettirilebilir.

```
gmx convert-tpr -s prod-sim1.tpr -extend 75000 -o prod-sim1-extended.tpr  
gmx mdrun -v -s prod-sim1-extended.tpr -cpi prod-sim1.cpt -deffnm prod-sim1
```

Tüm bilimsel çalışmalarda olduğu gibi MD simülasyonlarını da bir kaç kere tekrarlamak gerekir. Bu, bilimsel çalışmanın tekrarlanabilirliğini ispat etmek için olmazsa olmaz bir şeydir.

4.5 Yüksek başarılı hesaplama sistemleri

MD simülasyonları çok sayıda işlemciye veya ekran kartına ihtiyaç duyulan uzun hesaplamalardır. Bu nedenle bir masaüstü bilgisayar yerine yüksek başarılı hesaplama sistemlerinin kullanılması daha uygundur. Ülkemizde bu açıdan en iyi kaynaklardan biri Ulakbim tarafında sağlanan TRUBA-TRGrid hizmetidir. Bu hizmetten yararlanmak için şu web sitesini ziyaret edebilirsiniz: <https://portal.truba.gov.tr/>

4.6 Bölüm Özeti

5 MD Simülasyon Analizleri

5.1 Bu Bölümde Neler Öğreneceğiz?

Bu bölümde MD simülasyonlarının analizleri için kullanılan bazı standart teknikler anlatılacaktır. Bu analiz teknikleri yapısal ve enerji analizleri olarak ikiye ayrılabilir. Ele alacağımız yapısal analiz teknikleri şunlardır: RMSD, RMSF, Rg,SASA, ikincil yapı, PCA.

5.2 Yapısal Analizler

Moleküler dinamik (MD) simülasyonları, atomik düzeyde bir sistemin zaman içindeki davranışını incelememizi sağlar.

Bu tür simülasyonlarda elde edilen sonuçların anlamlı şekilde yorumlanabilmesi için çeşitli **yapısal analiz yöntemleri** kullanılır. Yapısal analizleri gerçekleştirebilmeniz için elimizde bir topoloji dosyası (genellikle gro, pdb veya tpr formatında) ve bir yörünge dosyası (xtc veya trr formatında) bulunması gerekir.

5.3 RMSD (Root Mean Square Deviation)

RMSD (Root Mean Square Deviation), bir molekülün belirli bir referans yapısına göre ortalama konum değişimini nicel olarak ölçen bir metriktir.

5 MD Simülasyon Analizleri

Basitçe ifade etmek gerekirse, RMSD zaman içinde yapının **ne kadar değiştiğini** gösterir.

Matematiksel olarak RMSD şöyle tanımlanır:

$$RMSD = \sqrt{\frac{1}{N} \sum_{i=1}^N (\mathbf{r}_i(t) - \mathbf{r}_i^{ref})^2}$$

Burada:

- N : Karşılaştırılan atom sayısıdır.
- $\mathbf{r}_i(t)$: Zaman t 'deki i . atomun koordinatıdır.
- \mathbf{r}_i^{ref} : Referans yapıdaki i . atomun koordinatıdır.

5.3.1 RMSD'nin Anlamı

RMSD değeri bize sistemin **yapısal kararlılığı** hakkında bilgi verir:

RMSD Değeri	Yorum
0 – 1 Å	Yapı referansa çok benzer; yüksek kararlılık
1 – 3 Å	Küçük yapısal değişiklikler; olası uyum veya gevşeme
> 3 Å	Belirgin yapısal değişim veya konformasyonel geçiş

RMSD'nin tek başına bir ölçüt olmadığı unutulmamalıdır. Yüksek RMSD değeri her zaman sistemin dengesiz olduğu anlamına gelmez; yapının farklı ama kararlı bir konformasyona geçmesi de mümkündür.

5.4 RMSF (Root Mean Squared Fluctuations):

5.3.2 Referans Yapı Seçimi

RMSD analizinde kullanılacak **referans yapı** son derece önemlidir. Genellikle aşağıdaki seçeneklerden biri tercih edilir:

1. **Başlangıç yapısı** (örneğin enerji minimizasyon sonrası yapı)
2. **Zaman ortalaması alınmış yapı**
3. **Kristal yapı veya deneysel referans**

Her durumda, RMSD hesaplanmadan önce **hizalama (alignment)** yapılmalıdır.

Bu hizalama genellikle **C (alfa karbon)** veya **backbone** atomları üzerinden gerçekleştirilir.

5.4 RMSF (Root Mean Squared Fluctuations):

5.5 Rg (Radius of gyration):

5.6 SASA (Solvent Accessible Surface Area):

5.7 İkincil Yapı

5.8 Enerji Analizleri

5.9 Bölüm Özeti

6 Ek 1: Amino Asitler ve Proteinler

6.1 Giriş

Proteinler, canlılığın iş yapan makromolekülleridir. Kataliz (enzimler), taşıma (hemoglobin), savunma (antikorlar), sinyal iletimi (reseptörler) ve yapı (keratin, kollajen) gibi pek çok görevi üstlenirler. Tüm bu çeşitliliğin ortak kökeni **amino asitler**dir—yani proteinlerin temel yapıtaşları.

Bu bölümde önce amino asitlerin ortak mimarisini ve kimyasal çeşitliliğini özetleyecek, ardından proteinlerin **birincil, ikincil, tersiyer ve kuaterner** yapı düzeylerini anlaşılır bir dille ele alacağız. Bölüm denklem içermez; kavramsal bir çerçeve sunar.

6.2 Amino Asitler

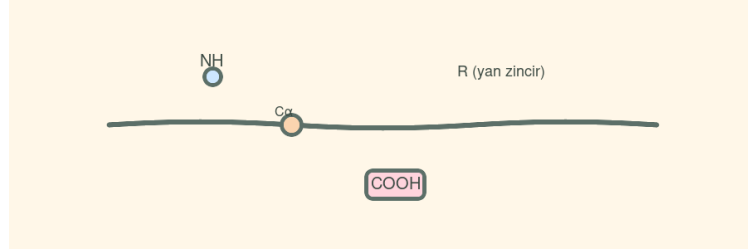
Her amino asit, aynı omurgayı paylaşır: bir **amino grubu** ($-NH$), bir **karboksil grubu** ($-COOH$), kiral bir **alfa karbonu** (C) ve değişken **yan zincir** (**R grubu**). R grubu; polar/apolar olma, yük taşıma, aromatiklik veya kükürt içeriği gibi özellikleriyle amino asitlerin kimyasal davranışını belirler.

- **Apolar (hidrofobik):** Val, Leu, Ile, Met, Ala, Phe, Trp, Pro

6 Ek 1: Amino Asitler ve Proteinler

- **Polar yüksüz:** Ser, Thr, Asn, Gln, Tyr, Cys
- **Asidik (negatif yüklü):** Asp, Glu
- **Bazik (pozitif yüklü):** Lys, Arg, His

Yan zincirlerin bu çeşitliliği, proteinlerin katlanma dinamiklerinden etkileşim ağlarına kadar her şeyi etkiler.



Şekil 6.1: Amino asidin genel yapısı

TODO: Şekil 6.1'yi düzeltmeyi unutma. !!!

6.3 Proteinlerin Yapı Düzeyleri

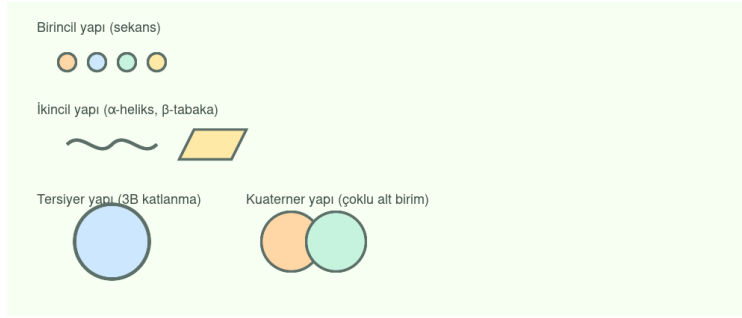
Protein yapısını kavramak için dört düzeyi ayırt ederiz:

1. **Birincil yapı (primary):** Amino asitlerin doğrusal dizilimi (sekans).
2. **İkincil yapı (secondary):** Yerel düzenli motifler — **-heliks** ve **-tabaka** gibi.

6.4 Protein Yapısı Belirleme Yöntemleri

3. **Tersiyer yapı (tertiary):** Tek zincirin üç boyutlu, işlevsel katlanmış hali.
4. **Kuaterner yapı (quaternary):** Birden fazla polimer zincirinin (alt birimlerin) bir araya gelerek oluşturduğu kompleks.

Bu düzeyler hiyerarşiktir: Sekans, ikincil motif eğilimlerini belirler; yan zincir etkileşimleri (hidrofobik paketlenme, hidrojen bağları, tuz köprüleri, aromatik etkileşimler, disülfid bağları) tersiyer mimariyi şekillendirir; uygun arayüzler ise kuaterner etkileşimleri mümkün kılar.



Şekil 6.2: Yapı düzeylerinin şematik gösterimi

TODO: Şekil 6.2'yi düzeltmeyi unutma. !!!

6.4 Protein Yapısı Belirleme Yöntemleri

Proteinlerin üç boyutlu yapısını anlamak, biyolojik işlevlerini çözümlmek için temel bir gerekliliktir. Bir proteinin dizisi (birincil yapısı) genomdan doğrudan elde edilebilir, ancak bu dizinin nasıl katlanarak üç boyutlu bir yapıya dönüştüğünü bilmek için ek yöntemlere ihtiyaç vardır. Bu

6 Ek 1: Amino Asitler ve Proteinler

yöntemler iki ana başlık altında toplanabilir: **deneysel** yöntemler ve **teorik** yöntemler.

6.4.1 Deneysel Yöntemler

6.4.1.1 X-ışını Kristalografisi

X-ışını kristalografisi, protein yapısını çözmek için uzun yıllardır kullanılan “altın standart” yöntemidir. Protein kristallerinden elde edilen kırınım desenleri analiz edilerek atomların üç boyutlu yerleşimi ortaya çıkarılır.

- **Avantajı:** Yüksek çözünürlük (genellikle 2 Å veya daha iyi).
- **Dezavantajı:** Kristal büyütmek zordur ve bazı proteinler kristallenmez.

6.4.1.2 Nükleer Manyetik Rezonans (NMR) Spektroskopisi

NMR, çözeltideki proteinlerin yapısını belirlemeye yarar. Atom çekirdekleri arasındaki manyetik etkileşimler ölçülür ve bunlardan uzaysal kısıtlamalar elde edilir.

- **Avantajı:** Kristalleşme gerektirmez, proteinin doğal ortamına daha yakındır.
- **Dezavantajı:** Genellikle 40 kDa altındaki proteinler için uygundur.

6.4.1.3 Kriyo-Elektron Mikroskopisi (Kriyo-EM)

Son yıllarda büyük bir atılım yapan Kriyo-EM, özellikle büyük makromoleküler komplekslerin yapısını belirlemede öne çıkmaktadır. Proteinler hızlıca dondurulur ve elektron mikroskobu ile görüntülenir.

- **Avantajı:** Kristalleşme gerekmez, büyük komplekslerde başarılıdır.
- **Dezavantajı:** Küçük proteinlerde çözünürlük sınırlı olabilir.

6.4.2 Teorik Yöntemler

6.4.2.1 Homoloji Modellemesi

Eğer hedef proteine benzer bir proteinin yapısı biliniyorsa, bu yapı şablon olarak kullanılarak hedef proteinin olası yapısı tahmin edilir.

- **Avantajı:** Güvenilir şablon olduğunda iyi sonuç verir.
- **Dezavantajı:** Şablon bulunmadığında sınırlıdır.

6.4.2.2 Ab initio (de novo) Modelleme

Protein yapısı tamamen diziden yola çıkarak tahmin edilmeye çalışılır. Fiziksel prensipler veya istatistiksel yaklaşımlar kullanılır.

- **Avantajı:** Şablon gerektirmez.
- **Dezavantajı:** Özellikle büyük proteinlerde halen zorluklar vardır.

6.4.2.3 Yapay Zekâ Tabanlı Yöntemler

Son dönemde **AlphaFold** ve benzeri yapay zekâ tabanlı yöntemler, protein yapısı tahmininde çığır açmıştır. Bu sistemler, devasa protein veri tabanlarından öğrenilen bilgileri kullanarak oldukça doğru tahminler yapabilmektedir.

Sonuç olarak, protein yapısını belirleme çalışmaları, biyoinformatik ile yapısal biyolojinin kesişiminde yer almaktadır. Deneysel yöntemler atomik ayrıntıyı verirken, teorik yöntemler hız ve esneklik sağlar. Bu iki yaklaşımın birlikteliği, günümüzde protein biliminin en güçlü araçlarından birini oluşturmaktadır.

6.5 Protein Veri Tabanları

Proteinler için en temel deneysel veri kaynağı Protein Databank'tir. Ancak, milyonlarca protein içeren AlphaFold veritabanı da deneysel yapısı bilinmeyen proteinler için oldukça iyi bir başlangıç noktasıdır.

6.6 Özet

- Amino asitler ortak bir omurga üzerinde farklı **R grupları** ile kimyasal çeşitlilik sunar.
- **Birincil yapı** sekansı, **ikincil yapı** yerel motifleri, **tersiyer yapı** tek zincirin 3B katlanmasını ve **kuaterner yapı** çoklu alt birimlerin montajını ifade eder.
- Bu hiyerarşi; katlanma, stabilite ve işlev arasındaki ilişkiyi anlamamanın anahtarıdır.

7 Ek 2: Protein Moleküler Dinamik Simülasyonlarında Kuvvet Alanları

7.1 Giriş

Moleküler dinamik (MD) simülasyonları, proteinlerin ve diğer biyomoleküllerin atomik düzeydeki hareketlerini zaman içinde modellemek için kullanılan güçlü hesaplamalı yöntemlerdir. Bu simülasyonlarda temel amaç, Newton mekaniği yasalarına dayanarak atomların konumlarını ve hızlarını belirli zaman adımları boyunca güncellemektir. Ancak atomlar arası etkileşimlerin doğrudan kuantum mekaniği ile hesaplanması çok pahalı olduğundan, **empirik kuvvet alanları (force fields)** geliştirilmiştir. Kuvvet alanı, sistemdeki atomlar arasındaki potansiyel enerjiyi tanımlayan matematiksel fonksiyonlar ve parametreler bütünüdür.

Proteinlerin simülasyonunda kullanılan kuvvet alanları, hem kovalent bağların özelliklerini hem de bağısız (non-bonded) etkileşimleri tanımlar. Doğru bir kuvvet alanı seçimi, simülasyon sonuçlarının deneysel verilerle uyumlu olabilmesi için kritik öneme sahiptir.

7.2 Kuvvet Alanlarının Genel Yapısı

Tipik bir biyomoleküler kuvvet alanı, potansiyel enerji fonksiyonunu şu bileşenlere ayırır:

7 Ek 2: Protein Moleküler Dinamik Simülasyonlarında Kuvvet Alanları

$$E_{total} = E_{bond} + E_{angle} + E_{dihedral} + E_{nonbonded}$$

7.2.1 Bağ (Bond) terimi

Atom çiftleri arasındaki kovalent bağların uzamasını harmonik yay modeli ile tanımlar:

$$E_{bond} = \sum k_b (r - r_0)^2$$

Burada k_b bağ kuvvet sabiti, r anlık bağ uzunluğu ve r_0 denge bağ uzunluğudur.

7.2.2 Açı (Angle) terimi

Üç atom arasındaki bağ açıları için harmonik yaklaşım kullanılır:

$$E_{angle} = \sum k_\theta (\theta - \theta_0)^2$$

7.2.3 Dihedral (Torsiyon) terimi

Dört atomdan oluşan zincirlerde dönme bariyerlerini tanımlar:

$$E_{dihedral} = \sum V_n [1 + \cos(n\phi - \gamma)]$$

Burada V_n dönme bariyeri yüksekliği, n periyod, ϕ torsiyon açısı ve γ faz kaymasıdır.

7.3 Protein Simülasyonlarında Kullanılan Başlıca Kuvvet Alanları

7.2.4 Bağırsız (Non-bonded) etkileşimler

Van der Waals ve elektrostatik etkileşimleri içerir. Genelde Lennard–Jones potansiyeli ve Coulomb yasası ile ifade edilir:

$$E_{nonbonded} = \sum \left[\frac{A}{r^{12}} - \frac{B}{r^6} + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right]$$

Burada A ve B Lennard–Jones sabitleri, q_i ve q_j kısmi yüklerdir.

7.3 Protein Simülasyonlarında Kullanılan Başlıca Kuvvet Alanları

Yukarıdaki denklemlerde bir sürü sabit parametre olduğunu gördük. Örneğin bağ enerjisini hesaplamak için k_b terimini bilmek gerekir. İki amino birbirine bağlayan C ve N atomları arasındaki peptid bağı için $k_b(C - N)$ değerini nereden elde edeceğiz? Kuvvet alanlarını buna benzer sabitlerin bulunduğu veri setleri olarak değerlendirebiliriz. Biyomoleküllerin MD simülasyonlarında yaygın olarak -kendi içinde tutarlı-bazı veri setleri, bir başka deyişle kuvvet alanları aşağıda tanıtılacaktır.

7.3.1 AMBER

AMBER (Assisted Model Building with Energy Refinement) aile kuvvet alanları proteinler, nükleik asitler ve ligandlar için yaygın olarak kullanılır. Öne çıkan versiyonlar: *ff14SB*, *ff19SB*. Ligand parametrizasyonu için **GAFF (General AMBER Force Field)** ve kısmi yük hesapları için RESP veya AM1-BCC yöntemleri tercih edilir.

7.3.2 CHARMM

CHARMM kuvvet alanı geniş kapsamlı parametre setleri sunar (protein, lipid, karbonhidrat vb.). *CHARMM36m* membran proteinleri ve lipid-protein etkileşimleri için sık tercih edilen bir versiyondur.

7.3.3 OPLS

OPLS-AA organik moleküllerin sıvı özelliklerini doğru tahmin etmesi ile bilinir; biyomoleküler simülasyonlarda da kullanılır.

7.3.4 GROMOS

GROMOS serisi, özellikle Avrupa kökenli birçok çalışmada tercih edilen klasik kuvvet alanıdır.

7.3.5 Yeni Nesil Kuvvet Alanları

- **Polarizable force fields (AMOEBA vb.):** Elektron bulutunun polarizasyonunu hesaba katarak daha doğru elektrostatik modelleme sağlar.
- **ML tabanlı potansiyeller:** Derin öğrenme ve makine öğrenmesi yöntemleriyle kuantum hesaplamalarına yaklaşıcı potansiyeller geliştirilmektedir.

7.4 Kuvvet Alanı Seçiminde Dikkat Edilecek Noktalar

- **Protein tipi:** Membran proteinleri için CHARMM36; çözünür globüler proteinler için AMBER ff19SB gibi kombinasyonlar uygun olabilir.

7.5 Sık Yapılan Hatalar ve Öneriler

- **Su modeli:** Kuvvet alanına uygun bir su modeli seçilmelidir (ör. TIP3P, TIP4P-EW, OPC). Su modeli, özellikle protein yüzeyi ve iyon dağılımı üzerinde belirgin etkiler yapar.
- **İyonlar ve konsantrasyon:** İyon parametrizasyonu ve iyon konsantrasyonu (NaCl, KCl) biyolojik koşulları yansıtmalıdır.
- **Ligand/kofaktör parametrizasyonu:** Antechamber (AMBER), CGenFF (CHARMM) gibi araçlarla ligand parametreleri çıkarılmalıdır.
- **Polarizasyon gereksinimi:** Yüksek doğruluk gereken hesaplamalarda polarizable kuvvet alanları düşünülmelidir.
- **Simülasyon amaçları:** Katlanma, bağlanma serbest enerjisi hesapları, uzun zaman ölçeği dinamikleri gibi farklı amaçlar farklı yaklaşımlar gerektirir.

7.5 Sık Yapılan Hatalar ve Öneriler

- Kuvvet alanı ile uyumsuz su modeli kullanmak.
- Ligand parametrizasyonunu ihmal etmek veya hatalı parametre kullanmak.
- Kısa dengeleme süreleriyle üretime geçmek.
- Kesme (cutoff) ve uzun menzilli etkileşimler için uygun algoritmalar (PME vb.) kullanmamak.

7.6 Özet

Kuvvet alanları, MD simülasyonlarının temelini oluşturur. AMBER, CHARMM ve OPLS gibi klasik kuvvet alanları günümüzde yaygın olarak kullanılmakta; ancak polarizable kuvvet alanları ve makine öğrenmesi tabanlı yaklaşımlar daha yüksek doğruluk vaat etmektedir. Simülasyonun amacına göre uygun kuvvet alanı, su modeli ve parametrizasyon stratejisi seçilmesi simülasyonun başarısı için kritik önemdedir.

8 Ek 1: Bash Komut Satırı

8.1 Giriş

Moleküler dinamik simülasyonları gibi hesaplamalı çalışmalar yapılırken komut satırı kullanımı vazgeçilmezdir. Linux/Unix tabanlı sistemlerde **Bash** (Bourne Again Shell), en yaygın kullanılan kabuk (shell) ortamıdır. Ubuntu ve Mac OS'ta komut satırını kullanabilmek için Terminal adındaki uygulamaya başlatmak zorundasınız. Terminal komutlarını çalıştırmak ilk başta size zor gelse de uzun vadede işlerinizi inanılmaz kolaylaştıracaktır. Aşağıda günlük işlerde sıkça kullanılan temel Bash komutları özetlenmiştir.

8.2 Dosya ve Dizin İşlemleri

Komut	Açıklama	Örnek
<code>pwd</code>	Bulunduğunuz dizini gösterir	<code>pwd</code>
<code>ls</code>	Dizin içeriğini listeler	<code>ls -l</code>
<code>cd</code>	Dizin değiştirir	<code>cd simulations/</code>
<code>mkdir</code>	Yeni dizin oluşturur	<code>mkdir data</code>
<code>rm</code>	Dosya veya dizin siler	<code>rm file.txt, rm -r old_dir</code>

8.3 Dosya Görüntüleme

Komut	Açıklama	Örnek
<code>cat</code>	Dosya içeriğini görüntüler	<code>cat notes.txt</code>
<code>less</code>	Uzun dosyaları sayfa sayfa gösterir	<code>less results.log</code>
<code>head</code>	İlk satırları görüntüler	<code>head -n 20 data.txt</code>
<code>tail</code>	Son satırları görüntüler	<code>tail -f log.txt</code>

8.4 Kopyalama ve Taşıma

Komut	Açıklama	Örnek
<code>cp</code>	Dosya veya dizin kopyalar	<code>cp file1.txt backup.txt</code>
<code>mv</code>	Dosya veya dizin taşır/yeniden adlandırır	<code>mv old.txt new.txt</code>

8.5 Arama ve Filtreleme

Komut	Açıklama	Örnek
<code>grep</code>	Dosyada metin arar	<code>grep "ERROR" log.txt</code>
<code>find</code>	Dosya/dizin arar	<code>find . -name "*.pdb"</code>
<code>wc</code>	Satır, kelime, karakter sayar	<code>wc -l data.txt</code>

8.6 Süreç ve Kaynak Yönetimi

Komut	Açıklama	Örnek
<code>ps</code>	Çalışan süreçleri listeler	<code>ps aux</code>
<code>top</code>	Gerçek zamanlı süreç takibi	<code>top</code>
<code>kill</code>	Belirtilen süreci sonlandırır	<code>kill 12345</code>

8.7 Dosya Aktarımı

Komut	Açıklama	Örnek
<code>scp</code>	Dosyayı uzak makineye kopyalar	<code>scp file.txt user@server:/home/user/</code>
<code>rsync</code>	Senkronize kopyalama yapar	<code>rsync -av data/ backup/</code>

8.8 Program Kurma

Ubuntu ve türevi işletim sistemlerin program kurmanın en kolay yolu program adının önüne `sudo apt install` yazmaktır. Örneğin Gromacs programını kurmak için `sudo apt install gromacs` yazmak yeterli olur.

8 Ek 1: Bash Komut Satırı

Burada Gromacs kelimesinin ve komuttaki öteki harflerin küçük olduğuna dikkat etmek gerekir. Program, Ubuntu veri tabanlarında mevcutsa işletim sistemi sizden yönetici şifresini soracak ve sonra programı kuracaktır.

8.9 Alıştırmalar

Not

1. Bulunduğunuz dizinin tam yolunu yazdırın.
İpucu: `pwd`
2. Bir dizindeki `.pdb` uzantılı dosyaların sayısını bulun.
İpucu: `ls *.pdb | wc -l`
3. `simulation.log` adlı dosyanın son 20 satırını ekranda gösterin.
İpucu: `tail -n 20 simulation.log`
4. `data/` klasörünü `backup/` klasörüne kopyalayın.
İpucu: `cp -r data backup`
5. `log.txt` dosyasında “ERROR” geçen satırların sayısını bulun.
İpucu: `grep -c "ERROR" log.txt`
6. `pdb2pqr` programını kurun.
İpucu: `sudo apt`

8.10 Sonuç

Bu komutlar, günlük kullanımda en çok ihtiyaç duyulan Bash araçlarıdır. Moleküler dinamik simülasyonlarının kurulumu ve analizi sırasında bu komutların bilinmesi, iş akışlarını büyük ölçüde hızlandırır.

9 Kaynaklar

9.1 Ders Kitapları

- Frenkel, D., & Smit, B. (2001). *Understanding Molecular Simulation*.

i Not

Nisbeten eski bir kitap. Hala C ile örnekler içeriyor.

- Rapaport, D. C. (2004). *The Art of Molecular Dynamics Simulation*.

9.2 Derleme Makaleler

- Hollingsworth, S. A., & Dror, R. O. (2018). Molecular dynamics simulation for all. *Neuron*, 99(6), 1129-1143.
- Karplus, M., & McCammon, J. A. (2002). Molecular dynamics simulations of biomolecules. *Nature structural biology*, 9(9), 646-652.

