

二重らせん DNA: polyA-polyT の設定

ここでは、sander プログラムで構造最適化と分子動力学計算を行うために必要な入力ファイルを設定するために使用する、AMBER パッケージに含まれるプログラムやツールの説明をします。

このセクションで使用する主プログラムは、LEaP です。

また、ここで説明するのは以下の項目です。

- 1) 分子座標の入手方法
- 2) シミュレーションレベルの決定
計算開始前に考慮すべき点
- 3) xleap を使用した、prmtop と inpcrd ファイルの作成

モデル構造の座標の作成

モデリングの最初のステップは、初期構造を作成することに始まります。原理上、xleap を使用してモデル構造を手で入力できますが、大きな生体分子系では難しい作業になります。その代わりに、実験的に求められた構造を使用することができます。これらは、結晶や NMR 構造データベースやなどを探し、見つけることができます。核酸も、などで検索できます。

実験による構造が得られなかった場合、ホモロジーモデリングの手法等を使用する様々なプログラムを利用して、モデル構造を構築することになります。全部をあげることはチュートリアルの範囲を超えますのでしませんが、核酸に関しては、Scripps 研究所の Dave Case と Tom Macke が作成した、複雑な核酸構造を構築するための NAB 分子操作言語などを検討してみてください (<http://www.scripps.edu/case/>)。タンパク質の構造を予測する手法が数多く存在していますが、通常それらの構造は未だ初歩的なものに留まっています。ですから、実験で求められた良い構造を使用する方が、通常は望ましいです。しかしもしも予測構造しか得られなかったとしたら、これらのシミュレーションでは念入りな最適化と平衡化の計算が本番の動力学シミュレーションの前に必要です。

困難その1：注意しなければいけないのは他の AMBER チュートリアルでも述べられているように、Brookhaven PDB ファイルを扱う時には様々な命名法が用いられているため、かなり用心しないとイケないということです。

さらに、全ての実験構造が PDB フォーマットで得られるとは限りませ

ん。その場合、OpenBabel

(<http://sourceforge.net/projects/openbabel/>)

のような変換プログラムを利用してください。

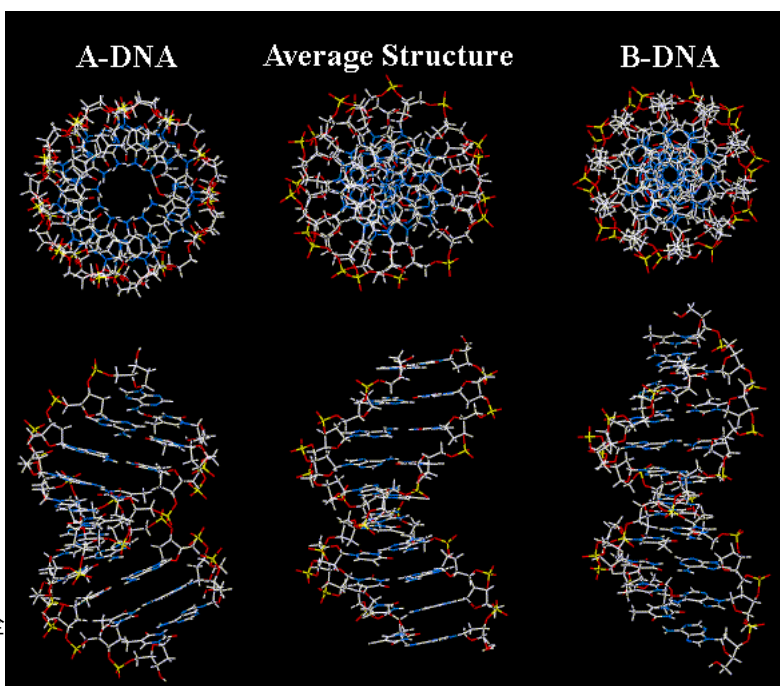
詳しい分子シミュレーションを行うために超えなければいけない最初のハードルが、命名法とファイルフォーマットの問題です。このすぐ後に、少なくとも xleap が必要とする命名法に関する内容に触れます。また、plastocyanin のチュートリアルにも命名法に関する内容が記述されています。

通常 AMBER で標準的なアミノ酸や核酸を用いている場合、必要なのは PDB フォーマットと原子名のちょっとした調整です。しかし複雑な糖鎖、脂質や非標準残基を含むタンパク質のシミュレーションを行いたい場合、パラメーターやトポロジー情報の開発が必要になります。これはさらに高度な内容なので、ここでは説明しません。

nucgen : DNA 二重らせん構造の作成

ここで使用する 10-mer DNA 二重らせん構造は実験的に得られていませんので、ゼロから作成する必要があります。幸い AMBER には、核酸の標準的な A- と B- 型の二重らせん構造を作成するプログラム「nucgen」が付いています。この nucgen によって出力された pdb ファイルは、AMBER 規則に従った原子名で記述されています。水素原子は全て記述されていないので追加する必要があります。

しかし、xleap は各残基が必要とする位置へ正しく追加してくれます。例えば "ADE" という名前の残基を読み込んだ時に欠落した原子があった場合、xleap はそれらを自動的に追加します。余分の原子は、モデル中に残されます。これらの原子に関して結合情報等は必要ありませんが、xleap にはそれらがどのようにつながっているかは分かりません。この事に関しては、後で述べます。



左側のものは標準的な A-DNA、中央は計算で求めた平均構造、右のものは標準的な B-DNA です。

力場についての注意

AMBERには、いろいろな力場が含まれています。バージョン 5.0 と 6.0 では、Cornell ら (1995) による parm94.dat でした。これは、AMBER 8.0 や 9.0 では FF94 と呼んでいます。バージョン 7.0 以降は、デフォルトの力場を設定していません。そのため、「prmtop」「inpcrd」ファイルを作成する際に、ユーザーがどの力場を使用するかを選択する必要があります。最新バージョンでは、実際の溶媒中でのタンパク質と核酸のシミュレーションの場合、FF03 力場を推奨します。これは、FF94 力場と比べ改善点がいろいろあります。最も著しい点は、均一溶媒下での DFT 量子計算に基づく、タンパク質の新しい部分電荷です。また、Phi-Psi 角のねじれ項も更新されており、FF94 と FF99 力場を使用した時に起きていた α ヘリックスの過大評価が改善されました。しかし FF03 では、核酸に関しては何も新しいものは含まれていません。電荷は真空中の ab initio HF 計算に基づいたもので、結合角とねじれパラメーターは FF99 力場と同一のもので、よってこのチュートリアルでは、FF99 全原子力場を使用します。利用できる力場に関して詳しい情報は、AMBER マニュアルとそこに記述されている参照文献をご覧ください。

力場 FF99 では、リン酸はヌクレオチドの一部として扱われます。末端残基 (3'- や 5'- 末端残基など) はここが他と違いますので、名前を変えて適切なトポロジー (結合や原子名など) とパラメーターにする必要があります。DNA の場合、残基名は 5' 末端のアデニンの場合 A5、末端でないものは A そして 3' 末のものは A3 となります (孤立したアデニンは、AN となります)。

以上の事をふまえて、nucgen で 10-mer DNA 二重らせんを構築するために必要なインプットファイルを作成します。構造は、標準的な Arnott B-DNA 構造にします。インプットファイルは、以下ようになります。このインプットで、A-T ペアの DNA 二重らせんを構築します。様々なオプションなどの詳細な内容は、マニュアルをご覧ください。

```
      NUC  1
D
A5  A   A   A   A   A   A   A   A   A   A3

      NUC  2
D
T5  T   T   T   T   T   T   T   T   T   T3

END
$ABDNA
```

注意：AMBER に含まれるプログラムを実行する前に、Unix シェルの環境変数（\$AMBERHOME）を AMBER をインストールした場所に正しく設定する必要があります。これは、xleap の実行に必須です。また、実行ファイルが有る場所（\$AMBERHOME/exe）をパス（\$PATH）に設定しておくとう便利でしょう。

環境変数 AMBERHOME のチェックをするには、以下のコマンドを入力します。

```
echo $AMBERHOME
```

もしも csh あるいは tcsh シェルの環境下で

```
AMBERHOME: Undefined variable.
```

と表示されるか、bash シェルの場合に空行が表示されるかしたら、AMBERHOME 環境変数が正しく設定されていません。もしも AMBER が /usr/local/amber にインストールされているとしたら、

csh あるいは tcsh シェル：

```
setenv AMBERHOME /usr/local/amber
```

bash：

```
export AMBERHOME=/usr/local/amber
```

と入力してください。

ホームディレクトリーの .cshrc (csh の場合) あるいは .bash_login (bash の場合) ファイルに、この変数設定を書き AMBER の実行ファイルディレクトリーをパスに加えておくと便利です。ホームディレクトリーの .bash_login に以下の行を加えます。あるいは bash を使用する全員用に /etc/bashrc に書いてもいいかもしれません。引数は、ご自分の機械へのインストール場所に合わせてください。

```
export AMBERHOME=/usr/local/amber
export PATH=$PATH:$AMBERHOME/exe
```

インストールの詳細は、マニュアルをご覧ください。

nucgen の実行

実行時には、nucgen に必要なインプット (nuc.in) アウトプット (nuc.out) nucgen データベース (nucgen.dat) 作成する PDB ファイル名 (nuc.pdb) を指定します。具体的には、以下のように入力します。ここでの「-O」は出力の上書きを指示します。

```
nucgen -O -i nuc.in -o nuc.out -d $AMBERHOME/dat/leap/parm/nucgen.dat -p nuc.pdb
```

これにより、2つのファイルが作成されます：nuc.out, nuc.pdb

モデルを作成する際の進行状況のログが、nuc.out ファイルに書き込まれます。興味があれば、読んでみてください。今回の例では、あまり有益な情報は含まれておりません、インプットファイルの簡単な説明だけです。重要なのは nuc.pdb で、これに DNA の二重らせん構造が含まれています。読み込まれたパラメーターファイル nucgen.dat を基にして決定された、全原子のデカルト座標が出力されています。ファイルは、438 行からなり各行がそれぞれの原子に対応します。

行の定義	原子番号	原子名	残基名	残基番号	X	Y	Z
ATOM	1	HST	A5	1	-0.808	-8.873	-2.080
ATOM	2	O3'	A5	1	4.189	-7.682	-0.250
...

Leap への構造の読み込み

次に、モデルの構造を確認します。モデルを使用する前に、構造をきちんと見るようにしてください。そうする事により、長い計算を行う前に問題点を識別する事ができる場合があります。適切な残基ファイルが xleap に読み込まれており、PDB ファイル中で原子名が xleap の期待するものになっていれば、正しく表示されます。また他にも、pdb ファイルを表示できるフリーウェアや商用のソフトがあります。その中で、学術研究目的では無料で利用できる VMD は、とても優れたプログラムです。しばらくはシミュレーション用のファイルを作成するために xleap を使用します。他の表示方法に関しては、このチュートリアルの後半で出てきます。

それから xleap では、残基名に加えて主鎖の接続情報が必要です。残基は、両側がつながっているのか、片方だけなのか、それとも結合して

VMD :
<http://www.ks.uiuc.edu/Research/vmd/>

いないのか、といった情報です。残基データには、この xleap 用語で言うところの "head" と "tail" のどちらか一方あるいは両方の設定が含まれています。

- head and tail : タンパク質あるいは核酸中の内部残基のように、両方に結合を持っているものです。例えば、アラニン "ALA" やチミン "T" などです。
- tail のみ : 鎖の開始末端の残基です。例えば、タンパク質のN末端残基 "NALA" や核酸の5'末端 "T5" などです。
- head のみ : 鎖の最終末端の残基です。例えば、タンパク質のC末端残基 "CALA" や核酸の3'末端残基 "T3" などです。
- head, tail無し : 他の残基と結合していないものです。例えば、孤立したヌクレオチド "TN"、水 "WAT"、イオン "Na+" などです。

様々な残基が定義されたライブラリーが、xleap の起動時に読み込まれます。PDB ファイル中の残基名は、デフォルトの xleap ライブラリーファイルやユーザー定義のライブラリーファイルに記述されているものと一致しなければなりません。詳しい内容は、後ほど述べます。

xleap で PDB ファイルを読み込む場合は、単純に全ての残基が記述されている順番につながっているものとします。ただし、"TER" カード ("TER" で始まる単一行) で仕切られて読み込んだ最初の残基が "tail のみ" で最後の残基が "head のみ" の場合は、その限りではありません。"TER" カードは、それまでの鎖を終わらせ新しい鎖を始めます。この事はシミュレーション用のファイルを作成した時に重要になりますので、覚えておってください。このチュートリアルのおすぐ後に出てくるように xleap に読み込む前に、"TER" が正しい位置に入るよう pdb ファイルを修正し、残基の命名法が正しいかチェックしなければいけない場合があります。

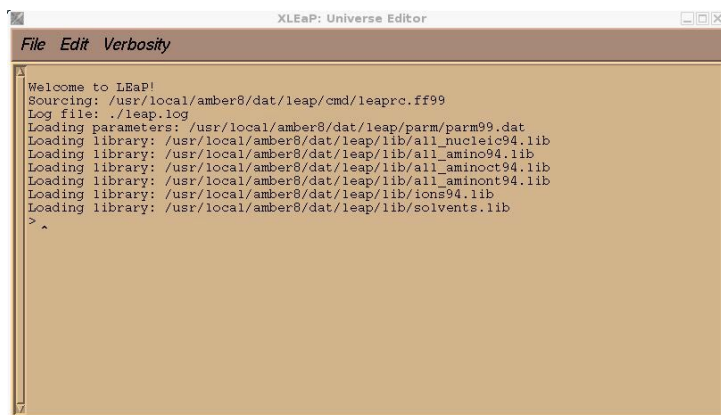
ここで作成した DNA を例に、上記の事がどのように関わるか見てみましょう。まずは、グラフィックバージョンの LEaP を起動します。

```
$AMBERHOME/exe/xleap -s -f $AMBERHOME/dat/leap/cmd/leaprc.ff99
```

パスの設定がきちんとされているのであれば、以下でもかまいません。

```
xleap -s -f leaprc.ff99
```

これにより xleap が起動し、以下のようなウィンドウが表示されます。



コマンドラインの説明：xleap が起動する時には、一連のライブラリーと力場パラメーター、残基マップなどを読み込む必要があります。AMBER は様々な力場を含んでいるため、それぞれのシミュレーションに適したものを選ぶ事が重要になります。ここで上記のオプション "-s" はデフォルトの設定を無視する事

XLEAP のメニューに関する注意事項：もしも xleap のメニューが働かないようでしたら、キーボードの numlock のライトが消えているかどうか確認してください。何らかの理由で、numlock がオンになっているとメニューの動作を止めてしまいます。

注：コマンドを入力する時は xleap のウィンドウが最前面に来て、マウスカーソルがウィンドウ内に入っているようにしてください。

を指示し、"-f \$AMBERHOME/dat/leap/cmd/leaprc.ff99" は FF99 力場のためのスクリプトを実行するよう指示しています。このスクリプトには、xleap が AMBER FF99 力場に必要な全ての構造ファイルの読み込みの指示も含まれています。また、\$AMBERHOME/dat/leap/cmd/ ディレクトリーを見ると、leaprc.ff03 (FF03 力場用)、leaprc.ff02ep (FF02、ローンペア電子を含む分極性力場用) など様々な leaprc ファイルが有ります。

PDB ファイルを xleap に読み込むには、loadPdb コマンドを使用します。このコマンドにより新しい UNIT が xleap 内に作成され、指定した PDB がそこに読み込まれます。引き続き edit コマンドを使用して、新しい UNIT を表示させたり編集したりできます。

そこで、以下のコマンドを xleap のウィンドウに入力して、「dna1」という新しいユニットを作成し PDB ファイルを読み込みます。

```
dna1=loadpdb "nuc.pdb"
```

以下のような出力が、xleap ウィンドウに表示されると思います。

```
Loading PDB file: ./nuc.pdb
Unknown residue: A3 number: 9 type: Nonterminal
Unknown residue: T5 number: 10 type: Nonterminal
Creating new UNIT for residue: A3 sequence: 10
One sided connection. Residue: missing connect0 atom.
Created a new atom named: P within residue: .R<A3 10>
Created a new atom named: O1P within residue: .R<A3 10>
...中略...
Created a new atom named: N1 within residue: .R<A3 10>
Created a new atom named: H3T within residue: .R<A3 10>
```

```

Creating new UNIT for residue: T5 sequence: 11
Created a new atom named: H5T within residue: .R<T5 11>
Created a new atom named: O3' within residue: .R<T5 11>
...中略...
Created a new atom named: C5 within residue: .R<T5 11>
Created a new atom named: C7 within residue: .R<T5 11>
One sided connection. Residue: missing connect1 atom.
total atoms in file: 438
Leap added 180 missing atoms according to residue templates:
180 H / lone pairs
The file contained 43 atoms not in residue templates

```

出力の意味は、指定した「dna1」ユニットに残基 10 (A3) と残基 11 (T5) が認識できなかったという事です。その他は、正しく読み込まれています。認識されなかったのは、一つ目のストランドの最終残基と二つ目のストランドの開始残基です。原因は、ストランドを分ける "TER" 行が無い場合、それぞれを末端残基として認識できなかったことです。そのため、読み込まれたときに "head" および "tail" 両方に残基が来るものとして認識されてしまっています。しかし、10 番目の残基 "A3" は末端の "connect0" 原子が見つからないため残基が正しく認識されず、原子は読み込まれたものの結合していない状態になっています。上のアウトプットには、以下のように新しい "unknown" 残基が作成された事が出力されています。

```

Creating new UNIT for residue: A3 sequence: 10
One sided connection. Residue: missing connect0 atom.

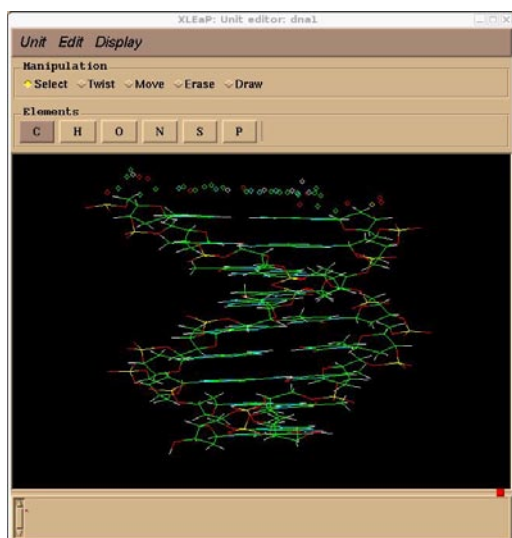
```

同様に、残基 11 の T5 も開始の原子 "connect1" が有りません。正しい動作をさせるためには、PDB ファイルを読み込む前に "TER" 行を追加する必要があります。通常、このように "connect0" や "connect1" 原子が無いというメッセージが出たら、鎖の結合状況が原因で、おそらくファイルに "TER" 行が無いか末端残基が正しく設定されていないかのどちらかです。また、「missing atom named "X"」とか「adding atom named "Y"」などと言ったメッセージが出た場合には、PDB ファイル内の残基名が xleap が認識できるものになっていないと言った命名ミスが考えられます。これに関しては、後ほど述べます。

構造を見るために、以下のように "edit" コマンドを入力してください。
ここでは、作成したばかり（エラーを含んだままの）ユニット "dna1" を指定します。

```
edit dna1
```

これにより、xleapのエディターウィンドウが開き、以下のような表示が
出ます（最初の分子の向きは以下と違っていても構いません）。



このウィンドウ内でマウスの左ボタンを操作すると、原子の選択になります（ドラッグ&ドロップも可）。中央ボタン（付いていれば）は分子の回転で、右ボタンは平行移動の操作になります。中央ボタンが無い場合は、コントロールキーを押しながら左マウスボタンをクリックする事によって同じ操作ができます。表示を拡大縮小するには、中央と右ボタン（あるいはコントロールキーと右ボタン）を同時に押して操作をします。マウスを上にかすと拡大で、下にかすと縮小になります。左マウスボタンで原子を選択した後に選択解除するには、シフトキーを押しながら何も無い所で選択ボックスをドラッグ&ドロップで表示させます。全部を選択するには、分子上でダブルクリックします。選択解除は、シフトキーを押しながら同じダブルクリック操作をします。

別プログラムを利用した構造表示

主として xleap は、AMBER の MD シミュレーション用のインプットファイルを作成するために、基本的で一般的な X-window グラフィックのみを用いて開発されました。他に、より洗練された分子表示プログラムを利用できます。それらのプログラムでは、より良い奥行き感やアンチエイリアス処理、システムによってはステレオ 3D で、分子を表示できます。いくつかは完全に自由に利用でき、またある物はアカデミックでは自由に利用できます。大変に高価なプログラムもあります。いくつか自由に利用できる物を以下に挙げ、簡単な説明を添えます。

VMD：イリノイ大学で開発（大変洗練されており、自由に利用できる。ダウンロードには、オンライン登録が必要）

RasMol：Roger Sayleにより開発（単純な静止構造表示、自由に利用可）

MOIL-View : SUNY-Stony BrookのCarlos Simmeringのグループにより開発（自由に利用できるが、SGIのみ）

PMV : ScrippsのMolecular Graphics Laboratory により開発（自由に利用可、pythonが必要、オンライン登録が必要）

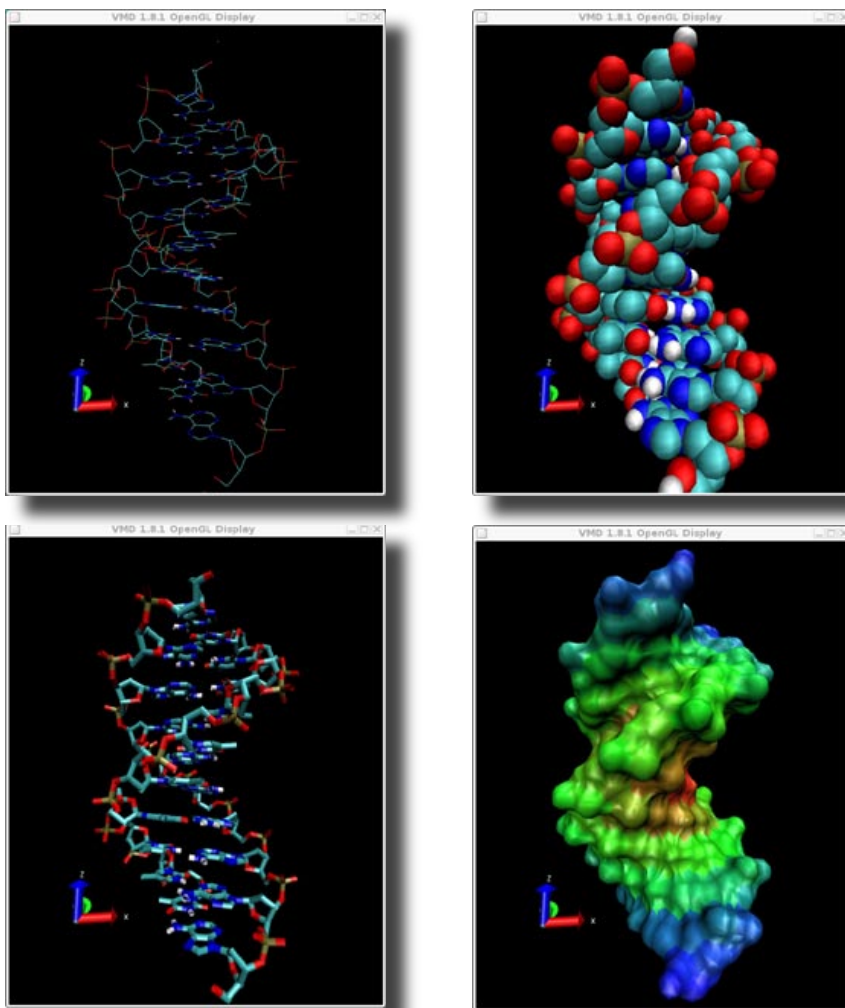
VMD - Visual Molecular Dynamics

大変洗練された分子表示プログラムで、静止画の表示と分子動力学トラジェクトリーの動画表示ができます。非常に多くのオプションがあります。また常に開発が続けられており、ほとんど全てのシステムで利用できますので、強く利用をお薦めします。

インストールしたら、コマンドvmdを使用して起動できます。起動時に、読み込むファイルを指定することもできます。

```
vmd nuc.pdb
```

いくつかの表示例を、以下に示します。



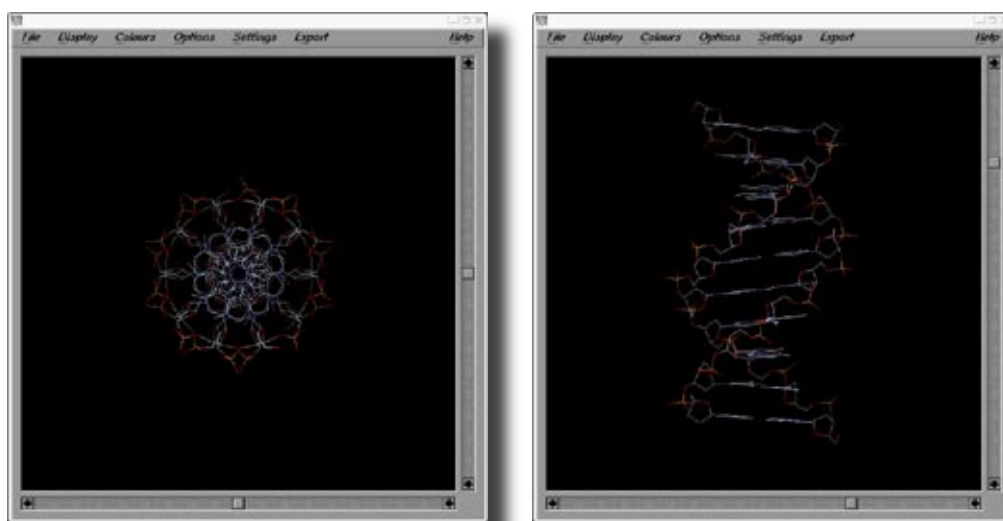
AMBER チュートリアル

rasmol

おそらく最初の、公共利用できるようになった分子表示プログラム。幾分機能も少なく既に活発に開発されてはいませんが、美しい静止画を作成でき利用も簡単です。

プログラムを起動し、"nuc.pdb" を表示させるには以下のように入力します。

```
rasmol nuc.pdb
```



これらのプログラムを使用して、構造を表示させてみてください。完全に対称的な標準 B- 型の DNA 構造が見られます。多くの水素原子が欠落している事にも、気づくと思います。上記の通り、xleap はデフォルトのテンプレートに従い水素原子を自動追加します。ここでの対称性は、長い DNA 鎖の解析を基にしています。ただ、いったんこの 10-mer の分子動力学計算を始めれば、完全な対称性は必要ないと言う事は明らかでしょう。MD シミュレーションから得られる構造の予測は、力場・溶媒モデル・長距離静電効果の扱いの「質」に大変依存します。構造や動き、とりわけシーケンスに依存した核酸の微細構造を正確にシミュレーションするモデルの開発は、みんながとても興味を持っている現在の問題です。

シミュレーションレベルの決定

適切なモデル構造ができあがったら次は、問題となっている系にどのレベルのシミュレーションを適用するのかを決めなければなりません。計算の複雑さの主な要因は、非結合およびクーロン相互作用の計算です。他には、周期境界条件と長距離の誘電効果を扱う Ewald 法や誘電分極などの非加算的な効果の評価が影響します。

水は核酸の構造に不可欠な部分であるので、溶媒の表現はかなり重要になります。気相のシミュレーションでは、距離依存あるいは S 字双極関数 (AMBER には導入されていません) などで溶媒をモデル化します。さらに、塩基対がはがれないように、ワトソン-クリック塩基対の拘束条件を加えるというトリックも行ったりします。リン酸上の電荷も少なくしたりします。新しいバージョンの AMBER (6.0 以上) では、一般化した Born モデルを仮想溶媒に適用し、時間はかかるものの距離依存の誘電率を用いるよりもより良い結果が得られるようにもできます。

核酸の気相シミュレーションを行う場合には、通常の Na⁺ カウンター・イオンを使うよりは、より柔らかく大きなカウンターイオンを使用した方が効果的な水のシェルを表現できます。

数々の DNA のシミュレーションが実際の溶媒モデル中で行われました。初期の実溶媒中の分子力学の計算では、時間が短く (100 ps 以下) 塩基対がはがれたようなおかしい構造が出ていました。

これらのシミュレーションから、溶媒のより正確な記述を含める必要があることがわかりました。最近のより長い計算 (およそ 1 ns) からは、長距離の静電相互作用を正確に扱うことも重要であることがわかってきました。

しかしながらコンピューター・パワーや、核酸の実溶媒とカウンターイオンを含んだナノ秒レンジの計算を日常的に行える Ewald 法の応用のような計算手法の進歩にも関わらず、未だ計算結果は分子力場に依存しています。ですから、使用している力場の欠点を理解する事は重要です。

まとめますと、可能でしたら、シミュレーションに溶媒を含めてください。そして系を中和するため、必ずカウンター・イオンを入れてください。

Singh, H., Weiner, S.J. & Kollman, P.A. *Proc. Nat. Acad. Sci.* 82, 755-759 (1985).

Beveridge and Ravishankar, *Cur. Op. Struct. Biol.* 4, 246-255 (1994),
Louis-May, S., Auffinger, P. & Westhof, E. *Cur. Op. Struct. Biol.* 6, 289-298 (1996) 参考文献も参照

Cheatham et al. *J. Amer. Chem. Soc.* 117, 4193-4194 (1995),
Cheatham & Kollman, *J. Mol. Biol.* 259, 434-444 (1996) 参考文献も

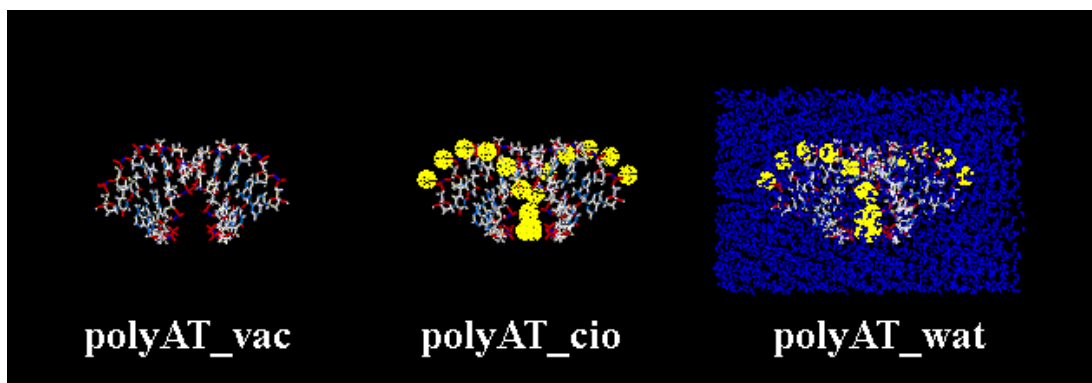
参照

また、使用する力場の制限なども気をつけるようにしてください。可能なら、Ewald 法のような正確に長距離静電効果を扱えるものを使用してください。しかし水を加えるのは、かなり計算量が増えます。気相のナノ秒オーダーの DNA シミュレーションを 3GHz Pentium4 のマシンで行うと数分しかかからない物が、およそ 10 Å の厚さの水を周りに加え周期境界ボックスを追加すると数日に計算時間が伸びます。通常、エラーやサンプリングなどで数回のシミュレーションを行うことになると思いますが、このような計算は、かなり時間がかかります。

シミュレーションタイプ

このチュートリアルでは、3つの異なったモデルを構築します。気相モデルの poly(A)-poly(T) 構造 (名称: polyAT_vac)、カウンター・イオンを付与した気相構造 (名称: polyAT_cio) そして周期境界ボックス内の TIP3P (水) 溶媒和モデル (名称: polyAT_wat) を作成します。気相モデルで、分子動力学計算のイメージをつかみます。続いて溶媒和モデルで、パーティクル・メッシュ Ewald 法を用いた周期境界条件下の計算を行います。

トラジェクトリーの解析を後で簡単にするために、3セットの prmtop



ファイルを用意します。これは、トラジェクトリーの解析中に溶媒はそんなには必要ではないからです。それに、読み込む前にトラジェクトリーから溶媒を除いておけば、表示用プログラムがより速く動きます。確かに、水は動径分布関数の計算、水の構造の解析をする時に必要ですが、二重らせんのパラメーターや平均構造などの計算には必要ありません。それで、ディスク・スペースの節約と解析のスピード・アップのために、よく水と時にはカウンターイオンを分離します。この3つの prmtop ファイルは、ptraj, rdparm, VMD などといったプログラムで (分離等を行った) トラジェクトリーの構造を使用したい場合に便利です。

トポロジーおよび座標ファイルの作成

出発構造のファイル (nuc.pdb) ができあがり、違うタイプの MD シミュレーションに関するいくつかの項目が理解できたと思いますので、AMBER の MD エンジンである sander で使用する入力ファイルを作成します。

最初のステップは、残基の構築です。多くのタンパク質は、標準的なアミノ酸と同様に補酵素も含んでいます。これらの補酵素は通常、AMBER データベースにあらかじめ定義されていないので、非標準残基として認識されます。シミュレーションに含まれる全ての非標準残基のための構造情報と力場パラメータを、sander の入力ファイルを作成する前に準備する必要があります。幸いにも、標準的な核酸あるいはアミノ酸残基だけを使用していれば (このチュートリアルがそう)、全ての残基が AMBER データベース中にあらかじめ作成されているので、そのような操作は必要ありません。

独自の残基を作成したい場合は、プラストシアニンのチュートリアルに良い例が載っています。

LEaP

もしも xleap が起動していないのであれば、以下のコマンドで起動してください。

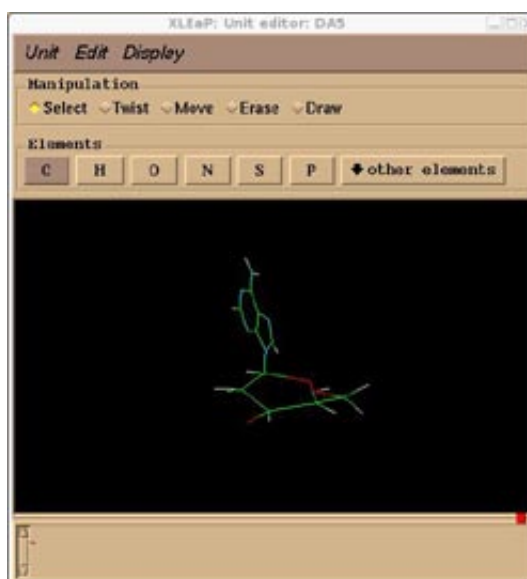
```
$AMBERHOME/exe/xleap -s -f $AMBERHOME/dat/leap/cmd/leaprc.ff99
```

その際、-f フラグで指定した適切なライブラリーが読み込まれているか確認します。

この例に必要な残基が読み込まれているか確認するには、ユニット DA5 を edit コマンドで開いてみます。

```
edit DA5
```

空白ではなく図のような画面が現れたら、残基は正しく読み込まれています。



トップメニューから、Unit -> Close と選んで、ウィンドウを閉じます（メニューが働かなかったら、NumLock キーをオフにしてください）。

定義されている残基をリストするには、LEaP で「list」と入力してください。以下のような表示が現れます。

```
> list
ACE   ALA   ARG   ASH   ASN   ASP   CALA  CARG
CASN  CASP  CCYS  CCYX  CGLN  CGLU  CGLY  CHCL3BOX
CHID  CHIE  CHIP  CHIS  CILE  CIO   CLEU  CLYS
CMET  CPHE  CPRO  CSER  CTHR  CTRP  CTYR  CVAL
CYM   CYS   CYX   Cl-   Cs+   DA    DA3   DA5
DAN   DC    DC3   DC4   DC5   DCN   DG    DG3
DG5   DGN   DT    DT3   DT5   DTN   GLH   GLN
GLU   GLY   HID   HIE   HIP   HIS   HOH   IB
ILE   K+    LEU   LYN   LYS   Li+   MEOHBOX MET
MG2   NALA  NARG  NASN  NASP  NCYS  NCYX  NGLN
NGLU  NGLY  NHE   NHID  NHIE  NHIP  NHIS  NILE
NLEU  NLYS  NMABOX NME   NMET  NPHE  NPRO  NSER
NTHR  NTRP  NTYR  NVAL  Na+   PHE   PL3   POL3BOX
PRO   RA    RA3   RA5   RAN   RC    RC3   RC5
RCN   RG    RG3   RG5   RGN   RU    RU3   RU5
RUN   Rb+   SER   SPC   SPCBOX THR   TIP3PBOX TIP4PBOX
TP3   TP4   TP5   TRP   TYR   VAL   WAT   parm99
>
```

使用する残基は、どれでしょうか？以下に、上で作成した PDB から O3' 原子を含む残基名をそれぞれ抜き出しました。

```
ATOM  2  O3'  A5    1    4.189  -7.682  -0.250
ATOM 25  O3'  A     2    7.904  -3.753   3.130
ATOM 209 O3'  A3    10   -1.127  -8.677  30.170
ATOM 231 O3'  T5    11    7.904   3.753  30.670
ATOM 252 O3'  T     12    8.601  -1.610  27.290
ATOM 420 O3'  T3    20    4.189   7.682   0.250
```

しかしながら、お気づきのように xleap は「A」という残基は認識しません。試しに「edit A」と入力すると、「A」という新しいユニットが作成されます。しかし、PDB ファイルには「A」残基が含まれています。た

だ、xleapは「nuc.pdb」を読み込んだ時に、「A」という残基に関するエラーを出しませんでした。問題は、残基10番と11番でした。それでは、xleapはどのようにして残基名（RNA vs. DNA、末端 vs. 非末端）を認識しているのでしょうか？

これは、xleapの起動時に指定したleaprc.ff99に定義されている残基マップでなされています。このファイル内を見ても、以下のような核酸残基マップを定義した部分が見つかります。

```
#
# Define the PDB name map for the amino acids and DNA.
#
addPdbResMap {
  { 0 "ALA" "NALA" } { 1 "ALA" "CALA" }
  .
  .
  .
  { 0 "GUA" "DG5" } { 1 "GUA" "DG3" } { "GUA" "DG" }
  { 0 "ADE" "DA5" } { 1 "ADE" "DA3" } { "ADE" "DA" }
  { 0 "CYT" "DC5" } { 1 "CYT" "DC3" } { "CYT" "DC" }
  { 0 "THY" "DT5" } { 1 "THY" "DT3" } { "THY" "DT" }
  { 0 "G" "DG5" } { 1 "G" "DG3" } { "G" "DG" } { "GN" "DGN" }
  { 0 "A" "DA5" } { 1 "A" "DA3" } { "A" "DA" } { "AN" "DAN" }
  { 0 "C" "DC5" } { 1 "C" "DC3" } { "C" "DC" } { "CN" "DCN" }
  { 0 "T" "DT5" } { 1 "T" "DT3" } { "T" "DT" } { "TN" "DTN" }
  { 0 "C5" "DC5" }
  { 0 "G5" "DG5" }
  { 0 "A5" "DA5" }
  { 0 "T5" "DT5" }
  { 1 "C3" "DC3" }
  { 1 "G3" "DG3" }
  { 1 "A3" "DA3" }
  { 1 "T3" "DT3" }
}
}
```

この定義は、PDBの残基名をxleapでの残基名に変換するものです。デフォルトで定義されているのは、「A」を「DA」に変換するといったDNA関連のものだけです。さらに「O4*」から「O4!」のように、PDBスタイ

ルの命名を AMBER 用に変換するものも含まれています。これは、xleap で DNA:RNA ハイブリッドのような非標準的な残基を含む分子を扱う場合、単に「loadPdb」コマンドを使用するだけでは望む結果は得られないことも指します。その場合、「loadPdbUsingSeq」といった、より進んだコマンドを使用します。これは、PDB 残基名をコントロールすることができます。

使用できる xleap のコマンドをリストするには、「help」と xleap ウィンドウで入力します。特定のコマンドのヘルプを見るには、「help command」とします。例えば loadPdb コマンドのヘルプを見る場合は、「help loadpdb」とします。以下のような出力が得られます。

```
> help loadpdb

variable = loadPdb filename
STRING _filename_

Load a Protein Databank format file with the file name _filename_.
The sequence numbers of the RESIDUES will be determined from the
order of residues within the PDB file ATOM records. For each
residue in the PDB file, LEaP searches the variables currently
defined for variable names that match the residue name. If a match
is found, then the contents of the variable are copied into the
UNIT created for the PDB structure. If no PDB `TER' card separates
the current residue from the previous one, a bond is created
between the connect1 ATOM of the previous residue and the connect0
atom of the new one. As atoms are read from the ATOM records, their
coordinates are written into the correspondingly named ATOMS within
the residue being built. If the entire residue is read and it is
found that ATOM coordinates are missing, then external coordinates
are built from the internal coordinates that were defined in
the matching UNIT (residue) variable. This allows LEaP to build
coordinates for hydrogens and lone pairs which are not specified
in PDB files.
```

それでは最初に戻ります。設定は全て、正しく行われているとします。上に書いた事は、背景で何が起きているのか、少し説明しただけです。特に研究において、このようなソフトウェアをブラックボックスとして利用するのは危険であるという事を、念頭に置いてください。

それでは、作成してある PDB ファイルを読み込んでみましょう。まずは、xleap が最初のストランドがどこで終わり、次のストランドがどこから始まるのか正しく認識できるように、PDB ファイル (nuc.pdb) に "TER" ラインをストランドの間に挿入します。これは残念ながら、nucgen プログラムでは行ってくれません。普段でも、この「TER」行が DNA のストランドやアミノ酸鎖の間にきちんと入っているか、確認するようにしてください。修正したファイルを、「nuc_ter.pdb」とします。以下に、挿入箇所とその前後を示します。

```
~~~~~  
ATOM   229  H3T  A3      10    -0.808  -8.873   31.720  
TER  
ATOM   230  H5T  T5      11     4.562   7.653   32.500  
~~~~~
```

では、この PDB ファイルを LEaP に読み込みましょう。以下のコマンドを、LEaP で入力してください。

```
model = loadpdb "nuc_ter.pdb"
```

うまく行けば、エラーは無く以下のような情報メッセージだけが出力されます。

```
Loading PDB file: ./nuc_ter.pdb  
total atoms in file: 438  
Leap added 200 missing atoms according to residue templates:  
200 H / lone pairs
```

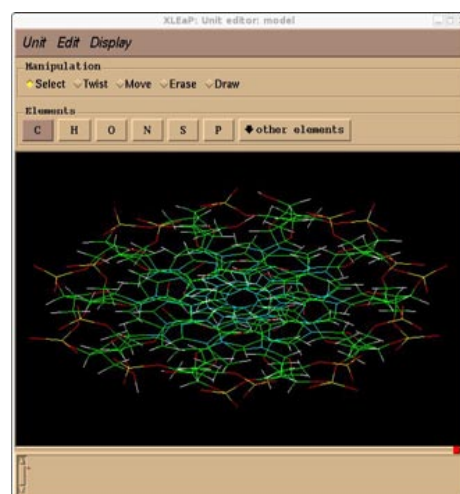
この出力からは、xleap が自動的に不足している水素原子を定義済みのテンプレートを基にして追加した事が分かります。

新しく作成した、「model」という名前のユニットを見るには以下のようにします。

```
edit model
```

すると、図のようなユニット・エディターのウィンドウが開きます。

AMBER チュートリアル



今度は結合していない原子は無く、残基 10 と 11 がきちんと認識されています。

そして、「prmtop」と「inpcrd」ファイルを作成するために、xleap ウィンドウで以下のコマンドを入力します。

```
saveamberparm model polyAT_vac.prmtop polyAT_vac.inpcrd
```

ウィンドウに、以下のような出力が出ます。警告が系を中和していないために出されていますが、これに関しては後ほど述べます。

```
> saveamberparm model polyAT_vac.prmtop polyAT_vac.inpcrd
Checking Unit.
WARNING: The unperturbed charge of the unit: -18.000000 is not zero.

-- ignoring the warning.

Building topology.
Building atom parameters.
Building bond parameters.
Building angle parameters.
Building proper torsion parameters.
Building improper torsion parameters.
total 110 improper torsions applied
Building H-Bond parameters.
Not Marking per-residue atom chain types.
Marking per-residue atom chain types.
(no restraints)
```

これにより、polyAT_vac.prmtop と polyAT_vac.inpcrd の 2 ファイルが作成されます。それぞれは、以下のようになります。

- polyAT_vac.prmtop：パラメーター／トポロジーファイル。モデルの接続情報とパラメーターが定義されています。この情報は静的なもので、シミュレーション中に変更されることはありません。
- polyAT_vac.inpcrd：座標ファイル。もし有れば、ボックスの座標と、ベロシティ情報も含まれます。ファイルは変更されませんが、メモリー中の座標は静的ではなく、シミュレーション中に更新されます。結果は、他のファイルに出力されます。

では次に、系を中和するカウンターイオンを追加した、トポロジーを作成します。構造にイオンを追加する方法はいくつかありますが、ここでは xleap が持っている addlons コマンドを使用します。この方法では、1.0Å のグリッド上にクーロンポテンシャルを構築し、最低 / 最高の静電ポテンシャルを持っている位置に、一つずつカウンターイオンを置きます。コマンドは、以下のように入力します ("0" は中和の意味です)。

```
addlons model Na+ 0
```

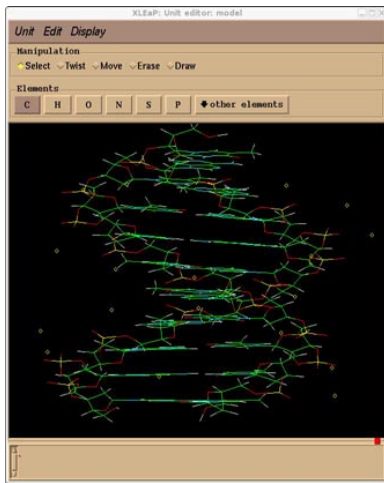
これにより、全部で18のナトリウムカチオンが追加され、DNA鎖が持つ-18の電荷が中和されます。

コマンドの出力は、以下のようになっていると思います。

```
> addlons model Na+ 0
18 Na+ ions required to neutralize.
Adding 18 counter ions to "model" using 1A grid
Grid extends from solute vdw + 1.87 to 7.97
Resolution:      1.00 Angstrom.
grid build: 0 sec
(no solvent present)
Calculating grid charges
charges: 0 sec
Placed Na+ in model at (-0.73, 10.83, 18.51).
Placed Na+ in model at (6.27, -8.17, 18.51).
Placed Na+ in model at (10.27, 4.83, 11.51).
Placed Na+ in model at (-6.73, -8.17, 11.51).
Placed Na+ in model at (-5.73, 3.83, 13.51).
~~~中略~~~
Placed Na+ in model at (6.27, 7.83, 15.51).
Placed Na+ in model at (10.27, -3.17, 8.51).
Placed Na+ in model at (-1.73, -12.17, 16.51).
Placed Na+ in model at (-9.73, 3.83, 4.51).
Placed Na+ in model at (-7.73, 8.83, 21.51).

Done adding ions.
```

注：アウトプットをいつも注意深くチェックし、意図しただけの数のカウンターイオンが定義されているかどうか確認してください。構造を表示させ、イオンが意図したとおり追加されているか確認してもいいでしょう。



以下のコマンドで、"model" を表示させイオンの位置を確認できます。

```
edit model
```

上と同様にして、この中和した系を以下のコマンドで2ファイルに保存します。

```
saveamberparm model polyAT_cio.prmtop polyAT_cio.inpcrd
```

作成する最後の入力ファイルは、カウンターイオンを含む溶媒和させた DNA です。既にカウンターイオンを含んだ系は作成しましたので、次のステップでは実際の水で溶媒和させます。このために、"solvatebox" というコマンドを使用します。ここでは、DNA の周りそれぞれの方向におよそ 8 Å の厚さで水を置きます。こうすると、DNA の出発構造内の全原子は、水のボックスの端から 8 Å 以上離れます。しかし一旦これを行う前に、コピーを作成し名称を「model2」とします。理由は、後で明らかになります。

```
model2 = copy model
```

以下のコマンドで、DNA の周りに直方体の水ボックスを発生させます (AMBER7 の場合は、TIP3PBOX の所が WATBOX216 になります)。

```
solvatebox model TIP3PBOX 8.0
```

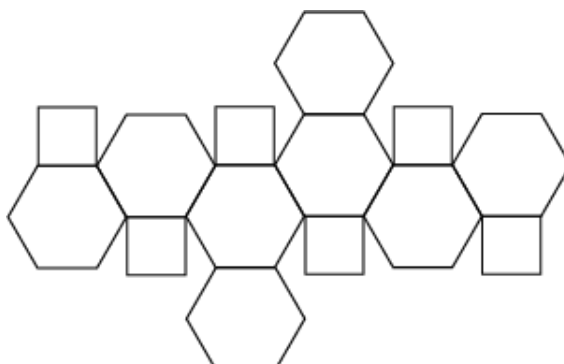
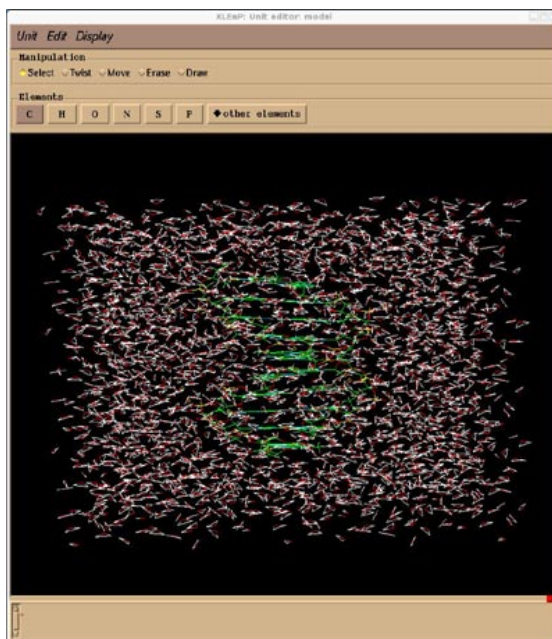
これにより、以下の出力が表示されます (コンピューターの丸め誤差によって、数値は微妙に違ってくると思います)。

```
> solvatebox model TIP3PBOX 8.0
Solute vdw bounding box:           25.736 26.736 40.240
Total bounding box for atom centers: 41.736 42.736 56.240
Solvent unit box:                   18.774 18.774 18.774
Total vdw box size:                 44.537 45.963 58.910 angstroms.
Volume: 120593.276 A^3
Total mass 53972.060 amu, Density 0.743 g/cc
Added 2638 residues.
```

以下のコマンドを入力すると、水ボックスの中に入った DNA が表示されます。

```
edit model
```

出力を見ると、xleap が 2,628 個の水分子を追加し、 $44.5 \times 46.0 \times 58.9 \text{ \AA}$ ($120,593.3 \text{ \AA}^3$) の直方体の箱が出来上がっています。立方体になっていないのは、DNA が円筒状の分子だからです。ここでの問題は、DNA の長軸が自己拡散により回転し得るということです。それにより長軸がボックスの短い方向に向き、DNA の端が自分自身の周期境界イメージに近づいてしまうことがあります。これを避けるために、`solvateBox` コマンドで数値のリストを指定し、ボックスを強制的に $58.9 \times 58.9 \times 58.9$ の立方体にすることができます。しかしこうすると、より多くの水分子が計算に追加され、計算がかなり遅くなってしまいます。代わりに、別の形状の水ボックスを使用することもできます。直方体というのは 3 次元空間にはめ込む際に当然の選択ですが、3 次元的に複製できる唯一の形状というわけではありません。溶質の回転問題を少なくするという観点から見ると、より効果的な形状となる切頭八面体のボックスを使用することもできます。



切頭八面体の水ボックスを DNA に加えるには、"`solvateoct`" コマンドを使用します。ここでは既に "model" に直方体の水ボックスを溶媒和させてしまっているので、先ほど複製した "model2" を使用します。以下のコマンドで、DNA の周りに水ボックスを発生させます (AMBER7 の場合は、

TIP3PBOX の所が WATBOX216 になります)。

```
solvateoct model2 TIP3PBOX 8.0
```

この結果、以下の出力が得られます。

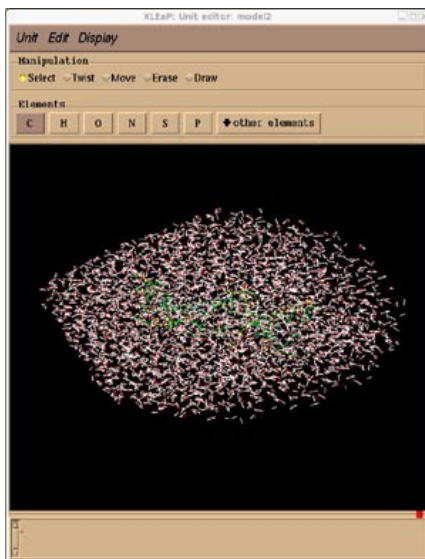
```
> solvateoct model2 TIP3PBOX 8.0
Scaling up box by a factor of 1.320477 to meet diagonal cut criterion
Solute vdw bounding box:           24.918 26.588 39.153
Total bounding box for atom centers: 60.281 60.281 60.281
(box expansion for 'iso' is 65.4%)
Solvent unit box:                 18.774 18.774 18.774
Volume: 115065.025 A^3 (oct)
Total mass 59917.340 amu, Density 0.865 g/cc
Added 2968 residues.
```

切頭八面体の水ボックスを見るには、以下のコマンドで edit モードに入ります。

```
edit model2
```

作成できていたら、AMBER parmtop と inpcrd ファイルを作成します。

```
saveamberparm model2 polyAT_wat.prmtop polyAT_wat.inpcrd
```



以上で、次の章以降で使用する入力ファイルができました。構造最適化と分子動力学計算に移ります。