

Lafire マニュアル

(バージョン 2.6)

(Lafire version 2.6 用, 2007 年 12 月 31 日)

姚 関

目次

1. 概説	3
2. Lafire の現時点での制限	4
3. インストール	4
3-1. Lafire インストール	4
3-2. Qt インストール	5
4. Lafire の使用	6
4-1. Lafire 用のデータ	6
4-2. Lafire の実行の用意	6
4-3. Lafire の実行 (実行, 停止, 再実行)	11
4-4. Lafire 出力ファイル	11
4-5. Lafire 結果の見方	14
5. Lafire の問題などに関する連絡	16
6. エラー・メッセージ及びその対応	17
7. Lafire_moleview の簡易の使い方	19

1. 概説

プログラム Lafire(Local-correlation-coefficient-based automatic fitting for refinement)は、初期モデル(断片的な部分モデルも可)から水分子を拾うまでの、タンパク質分子構造精密化の過程を全自動で行うためのソフトウェアである(図1)。

Lafireは、まず、アミノ酸配列ファイルに基づいて、初期モデルの配列を確認し、必要ならばアミノ酸の置換を行なう。精密化を含むサイクルに入ると、最初に、欠失部分或いは電子密度(初期位相から計算された電子密度, 2Fo-Fc, Fo-Fc)と大きくずれている部分の構造の構築を行う。次に、モデルと電子密度の一致度を1残基ずつ計算し、一致していない残基のフィッティングを行う。Lafireでは精密化は既存のソフト CNS 或いは REFMAC5 を利用している。また、精密化のモニター用に Rfree/R 因子のファイル, モデルの Ramachandran プロット用ファイルを出力する。これらはグラフィックプログラム Lafire_moleview により、簡単に可視化することができる。

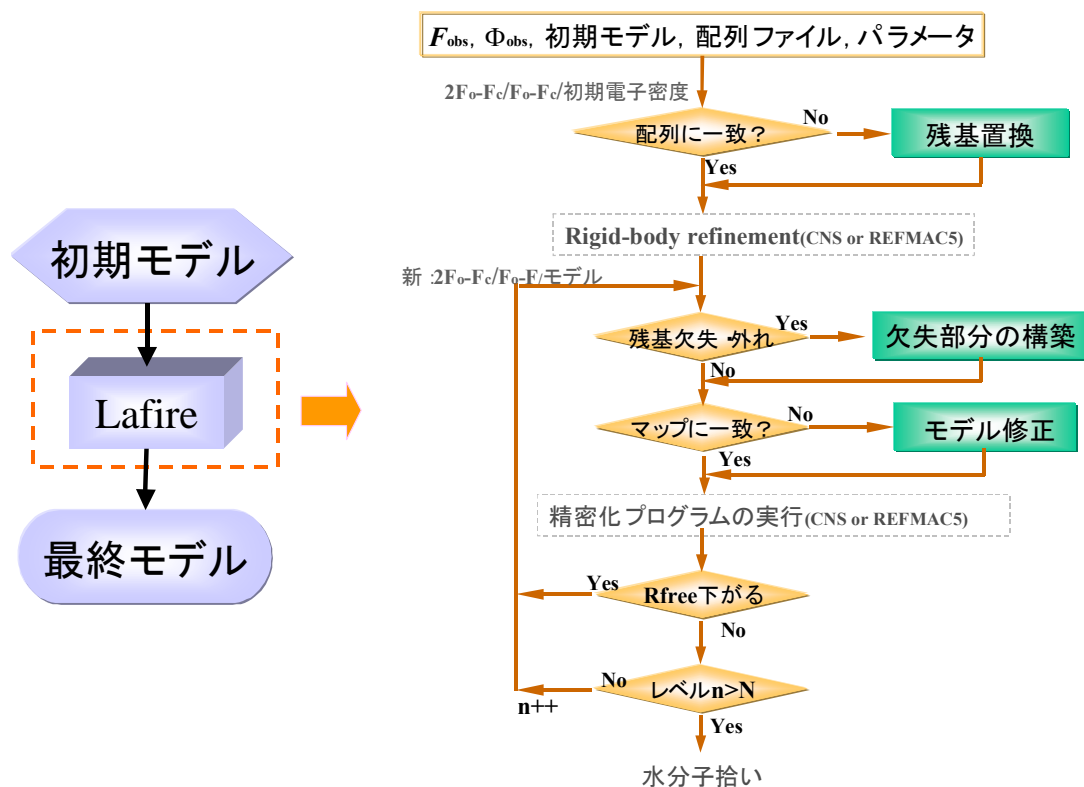


図1. Lafireによる自動精密化の流れ
レベルは Lafire に使われた構築と修正のストラテジーのレベルである

2. Lafire v2.6 の制限

- (1) Lafire は現在, アミノ酸側鎖の自動認識機能を持っていないため, 精密化の初期モデルの残基番号は必ずアミノ酸配列ファイルのそれと一致しなければならない.
- (2) 核酸とリガンドの構築及びフィッティングはできない.
- (3) マルチ・コンフォーメーションに対応しない.
- (4) No-Pro cis-peptid に対応しない.
- (5) Lafire と連動する精密化プログラムとして, CNS と REFMAC5 しか使えない.
- (6) 蛋白質以外の核酸や化合物, イオンなどの精密化用の topology ファイル(CNS のデフォルトの以外)が自動作成できない.
- (7) Lafire は現在, SGI 用 (IRIS6.5 以上) と Linux 用 (RedHat9.0 あるいは SUSE10.0 以上, Mandrake10.0,) のバージョンがある (IRIS6.4, RedHat7.0 には, Lafire の計算部分の提供が可能).
- (8) グラフィックスプログラム Lafire_moleview は OpenGL と Qt を利用している. SGI での (IRIS6.5) の実行が確認されている.

3. インストール

プログラム Lafire を用いて精密化するには, CCP4 のパッケージと CNS (精密化のために CNS を使用する場合のみ) が必要である. Lafire をインストールする前に, CCP4, CNS (CNS を使う場合のみ) をインストールして下さい. また, グラフィックスプログラム Lafire_moleview を実行するためには, Qt を /usr/local の下にインストールして下さい.

3-1. Lafire インストール

3-1-1. Lafire.tar.gz の解凍

例えば Lafire を /home/xtal の下にインストールする場合には, Lafire.tar.gz を /home/xtal の下に入れて, 次のように解凍して下さい

```
gunzip Lafire.tar.gz  
tar xf Lafire.tar  
(或いは gunzip Lafire.tar.gz | tar xf -)
```

解凍した Lafire ディレクトリの下に4つのディレクトリ bin, client, lib, example と configure ファイルができる.

3-1-2. Lafire の実行環境のテスト

configure を実行して, Lafire の実行に必要な CCP4, CNS, Qt の存在を確認する.

```
configure
```

3-1-3. Lafire の実行環境の設定

Lafire を実行するための環境変数及びパスの設定を .cshrc ファイルに入れる*. 例えば Lafire が /home/xtal の下にインストールされた場合には, 次の2行を .cshrc ファイルに入れて下さい.

```
setenv LAFIRE_DIR /home/xtal/Lafire (Linux の場合は, export LAFIRE_DIR=/home/xtal/Lafire)  
set path=($path /home/xtal/Lafire/bin)
```

alias cns_solve cnx_solve (CNX を使用する場合のみ)

* Lafire が C-shell の環境で実行するので、環境変数などの設定は C-shell の環境で行って下さい。

3-2. Qt ライブラリーのインストール (グラフィックスプログラム Lafire_moleview を使用する場合のみ)

Qt はノルウェーのソフトウェア企業 Troll Tech 社によって開発・配布されている、C++言語用の GUI(Graphical User Interface)クラスライブラリーである。通常、SGI にはインストールされていない。RedHat 9.0 以上の Linux にはデフォルトで /usr/local 下にインストールされている。もし、インストールされていない場合は、以下の手順で download してインストールして下さい。

3-2-1. Qt の download

次の Web から qt-x11-free-3.3.2.tar.gz を download して、/usr/local の下に解凍して下さい。

<http://www.trolltech.com/download/qt/x11.html>

3-2-2. コンパイル環境変数の設定

```
setenv QTDIR /usr/local/qt
```

```
setenv PATH ${PATH}:$QTDIR/bin
```

```
setenv MANPATH ${MANPATH}:$QTDIR/doc/man
```

```
setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:$QTDIR/lib
```

```
setenv QMAKESPEC irix-g++
```

(Linux の場合には、export を使って下さい。)

3-2-3. コンパイル

```
./configure -qt-gif -system-libpng -system-zlib -system-libjpeg
```

```
./make
```

Qt のインストール方法については、次の Web も参照してください。

<http://paison.hp.infoseek.co.jp/paison/qt/qtinstall.html>

4. Lafire の使用

Lafire の設定は全て C-shell の環境を利用するので、C-shell の環境で Lafire を実行して下さい。Lafire は動的にメモリを使用しているため、解析するタンパク質分子の大きさにより、実行時のメモリの使用量が異なる。通常は、500-600MB、大きなタンパク質の場合には 1GB 程度のメモリを使用する。

精密化プログラムとして CNS を使用する場合は、Lafire は `cns_configure` ファイルに基づいて、自動的にコマンドファイルを修正する。但し、非対称単位中に複数分子が存在する場合には、NCS-restraint の定義 (`~/Task_files/ncs.inp` の作成) と `~/Task_files/rigid-body` 精密化用のグループ定義 (8 分子以上のみ) の修正 (`rigid.inp` ファイルの修正)、リガンドがある場合には、トポロジーファイル、エネルギーのパラメータファイル、結合パッチの作成と `~/Task_files/generate.inp_org` の修正をしなければならない (4-2-8 を参照して下さい)。

同様に、精密化プログラム REFMAC5 を使う場合では、非対称単位中に複数分子が存在する場合は NCS-restraint の定義及び rigid-body 精密化用のグループ定義を自分で修正しなければならない。また、TLS を使う場合の TLS の定義ファイル `tls.inp` (`Task_files` 下にある) の修正、及びユーザ用リガンドを使うときのトポロジーファイル、エネルギー・パラメーターファイル (`*.cif`) の定義をしなければならない。これらの定義と修正は `~/Task_files` のディレクトリの下にある `refmac_*.com` ファイルを使って行って下さい (4-2-8 を参照して下さい)。

4-1. Lafire 用のデータ

- (1). 初期モデルの PDB ファイル (モデルのアミノ酸残基の番号はアミノ酸配列ファイル中の残基番号と必ず一致しなければならない)。
- (2). 初期モデルを構築するときに使われた位相と Fobs, また精密化に使う Fobs がある `mtz` ファイル。この `mtz` ファイルに、必ず R free 因子用の flag を設定して下さい。設定していない場合には Lafire が自動的に 7% を設定する。
- (3). アミノ酸配列ファイル (1 文字表示)
- (4). 2 次構造情報を使いたい場合には、2 次構造情報

R free 因子用の flag をまだ設定していない場合には、Lafire の `Task_files` ディレクトリの下に、flag を設定するコメントファイル `set_rfree.com` (CCP4 ソフト) がある。これを使って R free 因子用の flag を設定して下さい (パラメータを自分のサンプルに合わせて修正する必要がある)。

4-2. Lafire の実行の用意

4-2-1. ワーキングディレクトリの用意

Lafire_setup コマンドを使って、ワーキングディレクトリの下に、Lafire の実行環境を用意する。例えば `/users/phoIDH/Lafire ($working-directory)` がワーキングディレクトリであるならば、次のように Lafire_setup を実行して下さい。

```
Lafire_setup /users/phoIDH/Lafire
```

Lafire_setup の実行後に、`/users/phoIDH/Lafire` の下に 5 つのディレクトリ : `Data`, `Pdb`, `Task_files`, `Log`, `Result` と 5 つのファイル : `lafire.inp`, `sequence.inp`, `helix.inp`, `sheet.inp`, `cns_configure` が作られる (図 2)。また、Lafire にあらかじめ用意している例を使ってテストする場合には、次のように Lafire_setup を実行して下さい。

```
Lafire_setup /users/phoIDH/Lafire example
```

実行後に、テスト用のサンプルのデータが `Data`, `Pdb` の下に、パラメータが `phoIDH.inp`,

sequence.inp, cns_configure の中に用意されており, Lafire を実行できる状態になっている. 但し, phoIDH.inp ファイルの中にあるワーキングディレクトリ名と初期モデルの pdb ファイルのディレクトリ名を変更しなければならない(4-2-7を参照して下さい). そして, 4-3-1に従って, Lafire を実行して下さい.

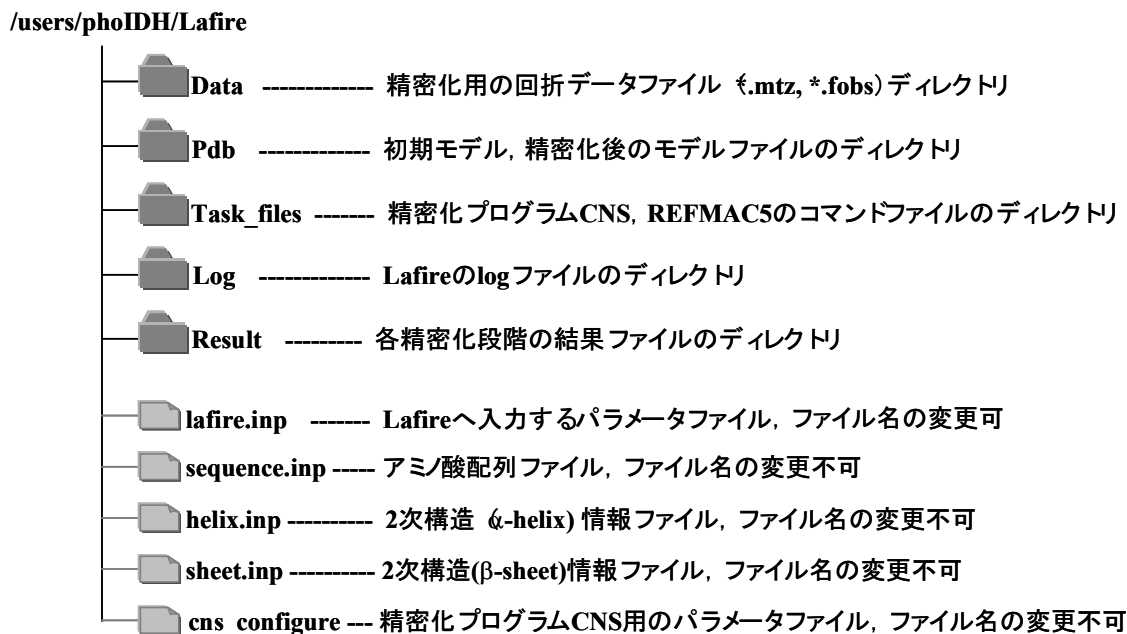


図 2. Lafire ワーキングディレクトリーの構造

4-2-2. 精密化用の回折データファイルの用意

R free 因子用の flag をセットした*.mtz ファイルを Data ディレクトリに置く.

CNS 用の回折データファイルが*.mtz ファイルから自動的に Data ディレクトリの下に lafire.fob という名前で作成される. もし, ユーザ自身で作成した CNS 用の反射データファイルを使いたい時には, 自分で作成したファイルに lafire.fob という名前を付けて~/Data の下に置いてください. もし, mtz ファイルがない場合には, 必ず CNS 用の反射データファイルから mtz ファイルを作って下さい.

4-2-3. 初期モデルの PDB ファイルの用意

初期モデルを1つの PDB ファイルにして 4-2-7 のパラメータファイルに書いてあるディレクトリの下に用意しておく. PDB ファイルを用意するときには次のように注意しなければならない. 核酸や化合物などがある場合には, Lafire は核酸や化合物などを修正できないが, 精密化プログラムによる精密化はできる. CNS を使用する場合には, CNS (デフォルト) に提供されている核酸や化合物, 金属イオンなど**以外**のトポロジー, エネルギーのパラメータを使用するとき, **ユーザ自身**で核酸や化合物などのトポロジー, エネルギーのパラメータ, 結合パッチを cns_confitg, ~/Task_files/generate.inp_org に定義しなければならない (4-2-8 を参照して下さい). REFMAC5 を使用する場合には, トポロジー, エネルギーのパラメータの定義を CCP4 のマニ

マニュアルに参照して下さい。そして、~/Task_files/refmac*.com を修正して下さい。

- (1). この PDB ファイルにある残基の番号は必ずアミノ酸配列ファイル中の残基番号と一致しなければなりません。
- (2). 非対称単位中に複数の分子あるいは核酸や化合物などの複合体が存在する場合には、1つの分子 (chain) の後に必ず一行の **END** 或いは **TER** を付けて下さい。Lafire は **END/TER** を認識して自動的に segid (CNS の場合) 或いは chain 名 (REFMAC5 の場合) を付ける。例えユーザが自分で segid 或いは chain 名を正しく付けても、タンパク質や核酸、化合物などの1分子ずつに **END/TER** をつける必要がある。Lafire はユーザが付けた segid 或いは chain 名を無視する。CNS の場合には、Lafire が、タンパク質分子ごとの segid を PROA, PROB, PROC, í , PROX, PROY, PROZ 順に、DNA の鎖ごとの segid を DNAA, DNAB, DNAC, í , DNAX, DNAY, DNAZ 順に、RNA の鎖ごとの segid を RNAA, RNAB, RNAC, í .., RNAX, RNAY, RNAZ 順に、化合物、金属イオンなどの非タンパク質分子ごとの segid を LIDA, LIDB, LIDC, í , LIDX, LIDY, LIDZ 順に付けている。REFMAC5 の場合には、Lafire がタンパク質分子ごとの chain 名を A, B, C, í 順に、タンパク質分子以外の分子 chain 名を Z, Y, X, í 順に付けている。
- (3). 基質などの化合物がペプチド鎖である場合には、ペプチド鎖はタンパク質鎖として取り扱って下さい。即ち、sequence.inp にペプチド鎖配列をタンパク質として入れ (4-2-4 を参照して下さい)、Lafire 実行用のパラメータ complex を 1 にする (4-2-7 を参照して下さい)。もし、基質がただアミノ酸 1 残基の場合には、20 種類のアミノ酸以外の名前を使って下さい。
- (4). Hetero-oligomer の場合には、PDB ファイルに並べる分子の順番は必ずアミノ酸配列ファイル (sequence.inp) に入れた配列の並び順番と一致しなければならない。このような複合体が非対称単位中に複数個が存在する場合には、PDB ファイルに並べる分子の順番は、まず1つの複合体の内部、そして、複合体間の順番で並べる。例えば、2つのタンパク質 (chain1, chain2) の複合体が非対称単位中に3個ある場合には、PDB ファイルの分子座標の並びは、chain1, chain2, chain1, chain2, chain1, chain2 になる。

4-2-4. アミノ酸配列ファイルの用意

ファイル sequence.inp の中身を自分のタンパク質の配列に編集する。

アミノ酸配列は一字表記で、1行 120 文字以内にして下さい。また配列以外の文字を入れないで下さい (スペース、改行は問題がない)。ファイルの format は以下のようにになっている。アミノ酸配列ファイルの例は \$LAFIRE_DIR/example にある。

```
-----
2                                     (非対称単位中にある異なる配列の分子の数)*
>chain1                               (鎖名)
MSRKLKAGVPYESVRDSLVRQAQDYIGSKEAKIKFVERE      (配列 (120 文字以内) )
KHEVKVKIGVFPNPYIRVIMRAFIADVNESVRDSL         (配列 (120 文字以内) )
LARLEAPFRARLRGTGYFPNEG                       (配列 (120 文字以内) )
>chain2                               (鎖名)
RDELAPYGVMRKS IPGWDKPFKPHITLARRKAPAPRVPPVL   (配列 (120 文字以内) )
FGLEWPVEGFALVRSELKPKGPV                     (配列 (120 文字以内) )
-----
```

*非対称単位中に homo-oligomer が存在する場合には、このパラメータが 1 である。

4-2-5. CNS 実行用のパラメータファイルの編集 (CNS 使用のみ)

精密化プログラム CNS を使用する場合には、cns_configure ファイルの中にある () 及び “ ” の中身などを以下の斜体部分が示しているように自分のサンプルに合わせて修正して下さい。その中に、öPARAM:ögがユーザにより作った topology, parameter, ncs-restraint ファイルのディレクトリ、öCNS_TOPPAR:ögが CNS により作った topology, parameter ファイルのディレクトリを定義するので、必ず書くことである。ö。

```
-----
(===>) high_res=2.05;                                     (高分解能側のリミット (Å))
-----
```


(==>) atom_select=(*known and not (hydrogen)*); (精密化する時に使う原子の選択)
 (==>) ncs_infile=öPARAM:ncs_main_chain.defö; (ncs-restraint ファイル名)^a
 (==>) restraint_infile=öPARAM:dna-rna.defö; (DNA/RNA-restraint ファイル名)^b
 (==>) parameter_infile_3=öCNS_TOPPAR:ion.paramö; (イオンのパラメータファイル名)^c
 (==>) parameter_infile_4=öCNS_TOPPAR:dna-rna.paramö; (DNA/RNA のパラメータファイル名)^c
 (==>) parameter_infile_5=öPARAM:atp.paramö; (リガンドのパラメータファイル名)^d
 (==>) porst_topology_infile=öPARAM:atp.topö; (リガンドのトポロジーファイル名)^e
 (==>) porst_parameter_infile=öPARAM:atp.topö; (リガンドのパラメータファイル名)^e

-
- a. ncs-restraint を加える場合には、ncs-restraint ファイルを必ず Task_files の下に入れてください。
 ‘PARAM : ø’も必ずファイル名の前に入れてください。ncs-restraint ファイルの内容を目的に応じて自分で修正して下さい。
- b. 核酸がある精密化する場合に、塩基対間のスタックの制限を加えることができる。制限条件ファイルが必ず Task_files の下に入れてください。‘PARAM : ø’も必ずファイル名の前に入れてください。例えば、Lafire には、Task_files の下にデフォルトに den-ran.def ファイルがあるので、öPARAM:dna-rna.defö。
- c. 金属イオンや核酸がある精密化する場合に、使用するエネルギーのパラメータファイルである。CNS に用意されている核酸、金属イオンのパラメータファイルを使用する場合には、編集する必要がない。ユーザが自分で定義したものを使用する場合には、下の c に示しているように編集して下さい。
- d. ユーザが自分で定義したりガンドのエネルギーのパラメータファイルを使用する場合には、必ずそのファイルを Task_files の下に入れて下さい。<PARAM : >も必ずファイル名の前に入れてください。
- e この 2 行は generate.inp 用である。ユーザが自分で作成したりガンドのトポロジーファイルとエネルギーのパラメータファイルを必ず Task_files の下に入れて下さい。<PARAM : >も必ずファイル名の前に入れてください。

4-2-6. 2 次構造情報の用意

2 次構造の情報を加えたい場合には、helix.inp, sheet.inp の内容を自分のサンプルに合わせて修正する。helix と sheet をその始めの残基番号と終了の残基番号を使ってそれぞれを定義する。helix.inp と sheet.inp の format は次のようになっている。

```

2 (非対称単位中にある異なる配列の分子の数)a
>chain1 (鎖名)
12:26 and 51:63 and 102:120 and 210:222 and 263:282 (各 helix (stand)の始めと終了の残基番号)b
300:316 and 333:345 (各 helix (stand)の始めと終了の残基番号)b
>chain2 (鎖名)
5:16 and 31:43 and 70:87 and 126:140 and 163:177 (各 helix (stand)の始めと終了の残基番号)b

```

-
- a. 非対称単位中に 1 種類のコピーが存在する場合には、このパラメータが 1 である。helix と sheet の定義も 1 つの chain に対して、定義する。
- b. それぞれの helix と sheet の定義 (始めの残基番号: 終了の残基番号) の間に and でつながる。1 行に最大の文字数が 120 である。

4-2-7. Lafire 実行用のパラメータの用意

Lafire.inp の以下の斜体部分だけを自分のサンプルに合わせて修正して下さい。

```

work_dir = /users/proIDH/Lafire (ワーキングディレクトリ名)
dir_prefix = cycle (各精密化段階のディレクトリ名の共通部分)
init_pdb = /users/proIDH/Lafire/Pdb/phoIDH_init.pdb (初期モデルのファイル名)

```

protein_prefix = *phoIDH_* (サンプル名, 出力ファイル名の一部として使われる)
 resolution = *2.05* (高分解能側のリミット(Å))
 MR = *0* (位相決定法のフラグ; 0:分子置換法以外の方法, 1:分子置換法)
 complex = *0* (複合体のフラグ; 0:単量体/ホモ多量体; 1:ヘテロ多量体)
 nmol = *4* (非対称単位中にあるコピー(homo)分子/(hetero)複合体の数)^a

 jump = *0* (Lafire ジョブのコントロール; 0: フィッティングからスタート
 1: 精密化からスタート(1)^b
 2: 精密化からスタート(2)^b
 3: フィッティングせず, 水分子拾って精密化)^c
 4: フィッティングしてから, 水分子拾って精密化)^c
 MSPP = *1* (部分構築にストラテジーを使用する; 1: 使用; 0:非使用)^d
 CNS = *1* (精密化に使用するプログラム; 0:REFMAC5; 1: CNS)
 annealing = *1* (精密化プロトコール, CNS のみ有効; 1:SA 使用; 0:SA 不使用)
 mtz_in = *resolve_remote.mtz* (mtz ファイルの名)
 labo = *remote* (mtz ファイルにある精密化用の Fobs と σ (Fobs)のラベル名)^e
 labi = *P* (mtz ファイルにある初期モデル構築する時の Fobs と σ (Fobs)のラベル名)^e
 phi = *M* (初期位相がある場合のみ有効, mtz ファイルにある位相のラベル名)^f

-
- a. 複合体の場合には, 非対称単位中にある複合体の数である.
- b. 精密化からスタートする場合には, 2 種類がある. (1)の場合には, 既に Lafire を実行して, 自動フィッティングと構築が行われた後, 精密化からやり直し, ~Result の下にサイクルゼロ (*cycle_0*)が必要である. (2)の場合には, Lafire を実行したことがなく, 直接的に精密化からスタートする. 例えば, 初期電子密度マップノイズが大きいと予測した場合には *jump=2* をお勧めである.
- c. 水分子を拾う場合にも, 2 種類がある. *jump =3* の場合には, フィッティングせず, 水分子を拾って精密化を行う. *jump =4* の場合には, フィッティングしてから, 水分子を拾って精密化を行う.
- d. バージョン 2.0 以降の Lafire には,構築・フィッティングに multi-level のストラテジーを導入した. このストラテジーによって, 構築の効率が高くなって,側鎖のフィッティングもよくなっている. 但し, プログラムの実行時間が長くなる.
- e. ここの F と σ (F)のラベル名は mtz ファイルにあるラベルの F と SIGF 後の部分である. 例えば, mtz にある F と σ (F)のラベルが FP, SIGFP である時は, ラベル名は P であり, FPremote, SIGFPremote である時ラベル名は Premote である. また, mtz ファイルのラベルを F, SIGF にしないで下さい, 必ず, F と SIGF の後に文字を入れて下さい. もし, 精密化用の Fobs と初期モデル構築用の Fobs が同じであれば, labo と labi に同じものを入力して下さい.
- f. このパラメータは初期位相がある場合のみ有効, 分子置換法の場合(MR=1)には無効である. 位相名は mtz ファイルにある位相のラベル PHI 後の部分である. 例えば, mtz にある位相のラベルが PHIM であるときラベル名は M, PHIdm であるときラベル名は dm である. また, mtz ファイルのラベルを PHI にしないで下さい, 必ず, PHI の後に文字を入れて下さい. HL(Hendrickson and Lattman)係数, HLA, HLB, HLC, HLD の後に, 必ず PHI と同じ文字を入れて下さい.

4-2-8. NCS-restraint や化合物などを加えた精密化の用意

(a). CNS の NCS-restraint

Lafire は CNS 用の NCS- restraint ファイルを 2 つ (ncs_all.inp, ncs_main.inp) ~Task_files の下に用意しており, 非対称単位中のコピーが 8 分子までである場合には, そのまま使える. 8 分子以上の場合には, ユーザ自分で NCS-restraint ファイル及び rigid.inp を修正しなければならない. 修正するときの segid の付け方については 4-2-3 を参照して下さい.

(b). REFMAC5 の NCS-restraint

REFMAC5 の NCS- restraint を ~Task_files の下にある reffmac*.com に入れなければならない. NCS の定義方法については, REFMAC5 のマニュアルを参照して下さい. また chain 名の付け

方については4-2-3を参照して下さい。

(c). CNS を使用する場合の金属イオンなどの化合物のパラメータの用意

CNS には核酸と金属イオンのトポロジーとエネルギーのパラメータファイルがあるので、そのまま使うことができる。但し、結合パッチの定義がないので、結合パッチを使う場合には、パッチのトポロジー、エネルギーのパラメータファイルを作成し、~/Task_files の下に置いて、そしてこの作成したトポロジー、エネルギーのパラメータファイル名を `cns_configure` に入れて下さい (4-2-5 を参照して下さい)。さらに、パッチの指定も~/Task_files の下にある `generate.inp_org` に入れなければならない。

また、トポロジーとエネルギーのパラメータファイルの作成は Web サイト HIC-up (<http://alpha2.bmc.uu.se/hicup/>) を利用して下さい。

(d). REFMAC5 を使用する場合の金属イオンなどの化合物のパラメータの用意

REFMAC5 の化合物のトポロジー、エネルギーのパラメータファイル名を~/Task_files の下にある `refmac*.com` に入れなければならない。トポロジーとエネルギーのパラメータファイルの作成については、REFMAC5 のマニュアルを参照して下さい。

4-3. Lafire の実行 (実行, 停止, 再実行)

4-3-1. Lafire の実行

Lafire 実行用の環境の設定, データ及びパラメータの用意ができれば, 次のように実行して下さい。

Lafire [Lafire へ入力するパラメータファイル名] &

例えば, パラメータファイル名が `phoIDH.inp` である場合, 実行コマンドは次のようになる。

Lafire phoIDH.inp &

4-3-2. Lafire の停止

実行中の Lafire を停止するときには, 次のようにして下さい。

kill_Lafire [Lafire へ入力するパラメータファイル名]

例えば, パラメータファイル名が `phoIDH.inp` である場合には, 停止コマンドは次のようになる。

kill_Lafire phoIDH.inp

4-3-3. Lafire 再実行

Lafire が異常終了後, 或いはパラメータを変更する場合に, Lafire を再実行するためには, まず **clear_Lafire** を使って既に作った不要なファイルとディレクトリーを以下のように削除する必要がある。

clear_Lafire [Lafire へ入力するパラメータファイル名]

例えば, パラメータファイル名が `phoIDH.inp` である場合に, 削除コマンドは次のようになる。

clear_Lafire phoIDH.inp

4-4. Lafire 出力データ

Lafire の出力ファイルは2種類ある。計算の結果ファイルとジョブのモニタファイル (*.log) である。ジョブのモニタファイルは `auto.log` と `job.log` であり、Lafire デバッグ用のファイルである。モニタファイルは、通常、見る必要はない。Lafire が異常終了したときに、問題を解決するために、作者に送ることがある。

4-4-1. Lafire 用のファイル

lafire.fos

CNSを用いて精密化する場合に、プログラムLafireは~/Dataの下にある*.mtzから自動的にCNS用の反射データファイルLafire.fobを作成する。もし、ユーザ自身で作成したCNS用の反射データファイルを使いたい時には、自分で作成したファイルにLafire.fobという名前を付けて~/Dataの下に置いてください。

サンプル名_*.seq

Lafire用のアミノ酸配列ファイル。ワーキングディレクトリの下にある。通常、読む必要がない。

サンプル名_*.helix

Lafire用の2次構造制限ファイル。ワーキングディレクトリの下にある。通常、読む必要がない。

サンプル名_*.sheet

Lafire用の2次構造制限ファイル。ワーキングディレクトリの下にある。通常、読む必要がない。

4-4-2. Log ファイル(~/Log の下にあるファイル)

error.log

Lafireを実行したときのエラー・メッセージ。

サンプル名_mut.out

初期モデルを配列と比較して、異なった残基について置換を実施したリスト。

サンプル名_built.out

Lafireにより構築した残基のリスト。

サンプル名_rfree.out

各精密化サイクルのRfree/R因子のリストファイルであり、4カラムがある。

1 番目のカラム：Rfree 因子の値

2 番目のカラム：R 因子の値

3 番目のカラム：精密化ステップのフラッグ

0：Lafireによる構築と修正後のRfree値

1：conjugate-gradient minimization 精密化後のRfree/R値

2：温度因子の精密化後（REFMAC5の場合にはML精密化後）のRfree/R値

3：simulated-annealing 精密化後のRfree/R値

- 4 : 水分子を拾う前の各精密化段階の最終の Rfree/R 値
- 5 : 水分子を拾ったときの各精密化段階の最終の Rfree/R 値
- 4 番目カラム : 各精密化ステップのファイル名.

4-4-3. 座標と電子密度マップファイル(~/Pdb の下にあるファイル)

サンプル名 [_Lafire.pdb](#)

Lafire により精密化した, 水分子を拾う直前の最良 free R を持つ PDB ファイル.

サンプル名 [_Lafire_torsion.pdb](#)

Lafire により精密化した, 水分子を拾う直前の最良 torsion angle を持つ PDB ファイル.

サンプル名 [_Lafire_water.pdb](#)

Lafire により精密化し, 水分子を拾った最良の PDB ファイル.

最良 torsion angle を持つ PDB の free R が最良 free R より 2%以下増加した場合には, 最良 torsion angle を持つ PDB を使って水分子を拾う. 最良 torsion angle を持つ PDB の free R が最良 free R より 2%以上増加した場合には, 最良 free R を持つ PDB を使って水分子を拾う.

サンプル名 [_Lafire_remove_water.pdb](#)

Lafire により精密化し, 水分子を拾った後に大きな塊のところにある水分子を削除した PDB ファイル.

何かの化合物の可能性を見出すために, 拾った水分子と水分子を拾う前の fo-fc マップから, 水分子に占有された大きな塊を検出する. 大きな塊を見つけたら, それを占有している水分子を削除する.

サンプル名 [_Lafire_removed_water.pdb](#)

Lafire により精密化し, 水分子を拾った後に大きな塊のところにある水分子の PDB ファイル. 拾った水分子と水分子を拾う前の fo-fc マップから, 水分子に占有された大きな塊を検出する. 大きな塊を見つけたら, それを占有している水分子をピックアップする. これらの水分子の残基名は BLB であり, 残基番号は[サンプル名 \[_Lafire_water.pdb\]\(#\)](#) ファイルと同じである.

サンプル名 [_Lafire_01.ps](#)

Lafire により精密化した, 水分子を拾う直前の最良 free R を持つ PDB の Ramachandran プロットの PS ファイル.

サンプル名 [_Lafire_torsion_01.ps](#)

Lafire により精密化した, 水分子を拾う直前の最良 torsion angle を持つ PDB の Ramachandran プロットの PS ファイル.

サンプル名 [_Lafire_water_01.ps](#)

Lafire により精密化した, 最良の PDB の Ramachandran プロットの PS ファイル.

サンプル名 [_Lafire_2fo-fc.map](#) (CCP4-format)

Lafire により精密化した, 水分子を拾う直前の最良 free R を持つ PDB の 2fo-fc マップ.

サンプル名 [_Lafire_fo-fc.map](#) (CCP4-format)

Lafire により精密化した、水分子を拾う直前の最良 free R を持つ PDB の fo-fc マップ。

サンプル名 [_Lafire_torsion_2fo-fc.map](#) (CCP4-format)

Lafire により精密化した、水分子を拾う直前の最良 torsion angle を持つ PDB の 2fo-fc マップ。

サンプル名 [_Lafire_torsion_fo-fc.map](#) (CCP4-format)

Lafire により精密化した、水分子を拾う直前の最良 torsion angle を持つ PDB の fo-fc マップ。

サンプル名 [_Lafire_water_2fo-fc.map](#) (CCP4-format)

Lafire により精密化した、最良の PDB の 2fo-fc マップ。

サンプル名 [_Lafire_water_fo-fc.map](#) (CCP4-format)

Lafire により精密化した、最良の PDB の fo-fc マップ。

サンプル名 [_Lafire_remove_water_2fo-fc.map](#)

Lafire により水分子を拾った後に、検出した大きな塊を占有している水分子を削除した PDB ファイルを使って精密化した 2fo-fc マップ。

何かの化合物の可能性を見出すために、拾った水分子と水分子を拾う前の fo-fc マップから、水分子に占有された大きな塊を検出する。大きな塊を見つけたら、それを占有している水分子を削除し、さらに精密化して 2fo-fc マップを出力する。ユーザがこの 2fo-fc マップと削除された水分子 (サンプル名 [_Lafire_removed_water.pdb](#)) から、化合物があるかどうかをチェックすることができる。

サンプル名 [_Lafire_remove_water_fo-fc.map](#)

Lafire により水分子を拾った後に、検出した大きな塊を占有している水分子を削除した PDB ファイルを使って精密化した fo-fc マップ。

何かの化合物の可能性を見出すために、拾った水分子と水分子を拾う前の fo-fc マップから、水分子に占有された大きな塊を検出する。大きな塊を見つけたら、それを占有している水分子を削除し、さらに精密化して fo-fc マップを出力する。ユーザがこの fo-fc マップと削除された水分子 (サンプル名 [_Lafire_removed_water.pdb](#)) から、化合物があるかどうかをチェックすることができる。

4-5. Lafire 結果の見方

Lafire は複数のプログラムからなり、例えば 1 サイクルの fitting 或いは building のプロセスにエラー (Segmentation fault (core dumped) エラー) が出ても、精密化の全過程には影響が無い場合がある。ですから、サンプル名 [_rfree.out](#) を見て Lafire による精密化がうまく行われているかどうかを判断して下さい (4-4 を参照して下さい)。

また、Lafire が計算する前に、ユーザが用意したデータをチェックし、Lafire の制限を満足しない場合には、実行せず、エラー・メッセージとエラー・コードを `~/Log/error.log` に出力する。そのエラーの原因及び対応について 6 を参照して下さい。

4-6-1. Rfree/R 因子のチェック

Lafire による精密化の過程での、Rfree/R の変化は~/Log の下にある[サンプル名_rfree.out](#) に出力されている。グラフィックスプログラム Lafire_moleview を使える場合には、Lafire_moleview で Rfree/R プロットを見ることができる (Lafire_moleview の簡易使い方の(4)を参照して下さい)。グラフィックスプログラムを使えない場合には、Microsoft Excel などのソフトで表示することができる。

4-6-2. モデル構築状況のチェック

Lafire では、モデルの構築状況、各精密化段階に構築した残基と、最終的に欠失している残基を~/Log の下にある[サンプル名_built.out](#) に記入している。

4-6-3. モデルのチェック

Lafire では、水分子を拾う前に Rfree/R が 1 番低い値を持つ PDB ([サンプル名_Lafire.pdb](#))、torsion angle が 1 番よい状態を持つ PDB ([サンプル名_Lafire_torsion.pdb](#))、水分子を拾った後の Rfree/R が 1 番低い値を持つ PDB ([サンプル名_Lafire_water.pdb](#)) ファイルを~/Pdb の下に出力する。また、それぞれの 2Fo-Fc、Fo-Fc のマップも~/Pdb の下に出力する。Coot, O, Turbo-frodo, Quanta, Lafire_mol などのプログラムを使って、モデルをチェックして下さい。

4-6-4. Ramachandran プロットのチェック

グラフィックスプログラム Lafire_moleview を使える場合には、PDB ファイルから直接そのモデルの Ramachandran プロットを見ることができる (Lafire_moleview の簡易の使い方の(4)を参照して下さい)。また、Lafire は、精密化過程での水分子拾う前に Rfree/R が 1 番低い値を持つ PDB ([サンプル名_Lafire.pdb](#))、torsion angle が 1 番よい状態を持つ PDB ([サンプル名_Lafire_torsion.pdb](#))、水分子をと拾った後の 1 番低い Rfree/R 値を持つ PDB ([サンプル名_Lafire_water.pdb](#)) の Ramachandran プロットを計算し、Ramachandran プロットの PS ファイルを出力している。グラフィックスプログラム Lafire_moleview を使えない場合には、これらの PS ファイルを見て下さい。

4-6-5. 化合物のチェック

バージョン 2.6 の Lafire に、fo-fc マップから化合物の可能性のある大きな塊を検出する機能を加えたので、Coot, O, Turbo-frodo, Quanta, Lafire_mol などのプログラムを用いて、[サンプル名_Lafire_remove_water_fo-fc.map](#), [サンプル名_Lafire_removed_water.pdb](#) ファイルを使ってチェックして下さい。

5. **Lafire** の問題などに関する連絡

Lafire を使うときに、プログラムのバグや問題或いはアドバイスなどがあつたら、次の email へ連絡して下さい。

lafire@castor.sci.hokudai.ac.jp

6. エラー・メッセージ及びその対応

Lafire はプログラム実行する前に、ユーザが用意したデータのチェックする。用意されたデータが Lafire の要求と異なる場合には、エラー・メッセージとコードを **~/Log/error.log** に出力する。

6-1. 精密化サイクルに入る前にエラーと対応

6-1-1. エラー・コード 1

初期モデルの PDB ファイルが~/Pdb に存在しない或いは*.inp ファイル(Lafire.inp)に定義したファイル名が間違った(4-2-7 を参照して下さい)。

6-1-2. エラー・コード 2

反射データファイル (mtz) が~/Data に存在しない或いは*.inp ファイル(Lafire.inp)に定義したファイル名が間違った(4-2-2, 4-2-7 を参照して下さい)。

6-1-3. エラー・コード 3

*.inp ファイル(Lafire.inp)に定義した mtz が存在しない、或いは、指定した Fobs, 位相などのラベル名が反射データファイル (mtz) に存在しない、或いは Rfree flag が存在しない(4-2-7 を参照して下さい)。

6-1-4. エラー・コード 4 (jump=1 のみ)

Lafire のパラメータファイルに jump を 1 に設定し、Lafire が既に building-fitting したモデルを用いて精密化からスタートするが、Result の下にサイクル 0 が存在しない (4-2-7 を参照して下さい)。この場合には、jump=0 にしてもう一回 Lafire を実行する。

6-1-5. エラー・コード 5

初期モデルの PDB ファイルにある残基番号が sequence.inp に定義した残基数より大きい。PDB ファイルをチェックして下さい。

6-1-6. エラー・コード 6

初期モデルの PDB ファイルにある残基番号が小さい順ではなかったか、あるいは異なる分子の鎖の間に **END/TER** を入れていなかったか、PDB ファイルをチェックして下さい。

6-1-7. エラー・コード 7

初期モデルの PDB ファイルにおいて、タンパク質の中に核酸や金属イオン、化合物などの非 20 種類のアミノ酸が含まれている。核酸や金属イオン、化合物などがある場合には、タンパク質の鎖の間に **END/TER** を入れてください。PDB ファイルの用意は 4-2-3 を参照して下さい。

6-1-8. エラー・コード 8 (非対称単位中にコピー分子が存在する場合のみ)

初期モデルの PDB ファイルにあるタンパク質の鎖の数が、Lafire に入力した (4-2-7 を参照) パラメータ nmol で定義した数と合わない (多い或いは少ない)。この場合には、PDB ファイルタンパク質の鎖の座標がない、或いはタンパク質の鎖の間に **END/TER** がない (4-2-3 を参照)。

6-1-9. エラー・コード 9 (非対称単位中にコピー分子が存在する場合のみ)

初期モデルの PDB ファイルにおいて、非対称単位中のコピー分子間の残基番号が揃っていない。つまり、PDB ファイルにあるコピー分子の残基番号が sequence.inp に定義した残基と合わない。

6-1-10. エラー・コード 10

反射データ mtz ファイルに定義していた (Lafire 実行用のパラメータファイル(Lafire.inp)にも定義している) Fobs と σ (Fobs) のラベル名が FC (Fc), SIGFC (SIGFc)或いは位相のラベル名が PHIC (PHIc) になっている。これらのラベル名 FC (Fc), SIGFC (SIGFc), PHIC (PHIc)が

Lafire の中に既定ラベル名として使われているので、FC (Fc), SIGFC (SIGFc), PHIC (PHIc) 以外のラベル名を使わなければならない。

6-2. 他のエラー・メッセージと対応

6-2-1. エラー・メッセージ Available memory is below 512 MB

このエラーはユーザの使えるメモリを調べるときに起こったものである。その原因は、ユーザが使えるメモリが小さすぎ (<0.5GB 程度のメモリ)、あるいは、コンピュータのシステムの問題で、使えるメモリの情報を正しく取れないからである。

このエラーが起こった時には、コンピュータのメモリサイズをチェックして下さい。もし、ユーザが使えるメモリは十分であれば、LAFIRE の作者に連絡して下さい (lafire@castor.sci.hokudai.ac.jp)。連絡するときには、使っている機種、OS、CPU 名を作者に送って下さい。

6-2-2. エラー・メッセージ Memory overflow in building

このエラーは構造の欠失している部分 (連続している) が大きすぎ、あるいはユーザが使えるメモリのサイズが小さいからである。

このエラーが起こった時には、ジョブの流れには影響がない。精密化の進行によって欠失している部分が減少して、エラーがなくなる可能性がある。

6-2-3. エラー・メッセージ Memory overflow in fitting

このエラーはフィッティングが必要な部分 (連続している) が大きすぎ、あるいはユーザが使えるメモリのサイズが小さいからである。

このエラーが起こった時には、ジョブの流れには影響がない。精密化の進行によってフィッティングが必要な部分が減少して、エラーがなくなる可能性がある。

7. Lafire_moleview の簡易の使い方

Lafire_moleview は精密化をモニターするグラフィックスプログラムである。このグラフィックスプログラムはこれまでの O, Turbo-Frodo, Quanta などと異なって、良好な視野を確保するため、3/5/7 残基しか表示しない。また、電子密度の表示範囲は box ではなく、原子座標を中心にして、R 半径の球となる。

Lafire_moleview は SGI (IRIS6.5 以上) をサポートしている。ここでは Lafire_moleview の簡易的な使い方を説明するが、詳細な使い方は「タンパク質分子構造精密化モニタープログラム (Lafire_moleview) 操作説明書」を参照して下さい。

7-1. Lafire_moleview の実行

Lafire_moleview は次のようなコマンドで実行する。

```
Lafire_moleview &
```

7-1. PDB ファイルの読み込み

Lafire_moleview 画面の左上の「File」→「Open PDB file」を使って PDB ファイルを読み込むことができる。Lafire_moleview では 5 つの PDB ファイルまで読むことができる。

7-3. Rfree/R プロット

実行している Lafire_moleview 画面の左上の「Window」→「Rfactor」を使って [サンプル名_rfree.out](#) を読み込んで、Rfree/R プロットを表示することができる。

7-4. Ramachandran プロット

まず、Lafire_moleview 画面の左上の「File」→「Open PDB file」を使って PDB ファイルを読み込む。それから、Lafire_moleview 画面の左上の「Window」→「Ramachandran Plot」を使って、Ramachandran プロットのサブウィンドウを開く。そのサブウィンドウの左下にある「molecule」ボタンで読み込んだモデルを選んでから、サブウィンドウの右下にある「Draw Now」ボタンを押せば、選んだモデルの Ramachandran プロットが表示される。青色で表示されているのは、メインウィンドウに表示されている残基の torsion angle である。