

BINDS 講習会：2020年度

2020/10/07

「画像解析実習--- Relion-3.1 を使ってみる」

別冊テキスト

(更新版 1.32)

三重大学・大学大学院医学系研究科

谷 一寿

目次

1. 前処理 -----	3
2. 粒子のピックアップ -----	11
3. 2D クラス平均 -----	24
4. 3D モデル生成 -----	29
5. 3D クラス分類 -----	32
6. 3D 精密化 -----	37
7. マスク作成とポスト処理 -----	41
8. CTF と収差変数の精密化 -----	45
9. Bayesian Polishing による粒子画質改善 -----	51
10. 局所空間における分解能評価 -----	55
11. 手系の確認 -----	57
12. Relion-3.1 の互換性に関する注意事項 -----	57
Appendix A. 計算結果とディレクトリ構造 -----	58

参考文献

謝辞

1. 前処理

1.0 はじめに

コマンド入力を行う箇所は、“>”で始まります。入力するのは、“>”より最後の“↵”（エンターキー）までになります。

今回使用する画像データは、*Thermoplasma acidophilum* の 20S protomasome です（参考文献 2）。<https://www.ebi.ac.uk/pdbe/emdb/empiar/entry/10025/>（日本のミラーサイト <https://empiar.pdbj.org/entry/10025/>）からダウンロードできます。全データ 196 枚の画像を使用します。

今回は、ディレクトリ /data/binds/Prot20S_Tacid で実習を行います。

```
>cd /data/binds/Prot20S_Tacid ↵
```

⇒Unix のコマンドで、ディレクトリを移動する際に使用します。詳細は文書末の Appendix I を参照ください。このディレクトリに検出器依存の MTF (Modulation Transfer Function) を記述したファイルがあることを確認します。今回は Gatan 社 K2 で加速電圧 300kV の電子顕微鏡を使用しているため K2summit_at_300kV.star を用います。実際に relion を起動させてみましょう。

```
>relion ↵
```

最初に起動させると必ず下のメッセージがでますので、“y”と答えてください。

Only run the relion GUI from your ProjectDirectory. Do you want to start a new project here [y/n]? y

⇒下の画面がポップアップしてきます。（ウィンドウ中の説明は Appendix A を参照のこと）

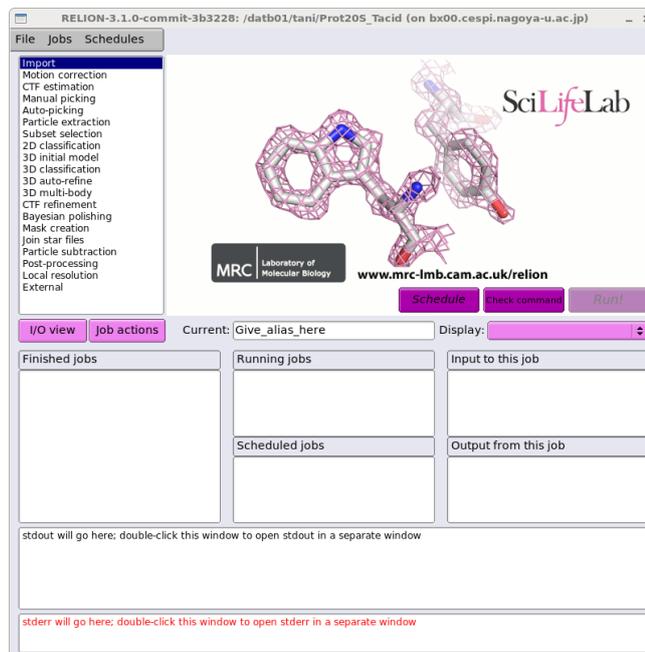


図 1.0 : 起動画面

また、1 時間何も操作を行わない場合には、下記の

“The relion GUI has been idle for more than 3600 seconds, exiting now...”

というメッセージがでてきてGUIが自動的に消されます。その際は、再度 relion と入力してください。

1.1 画像のインポート

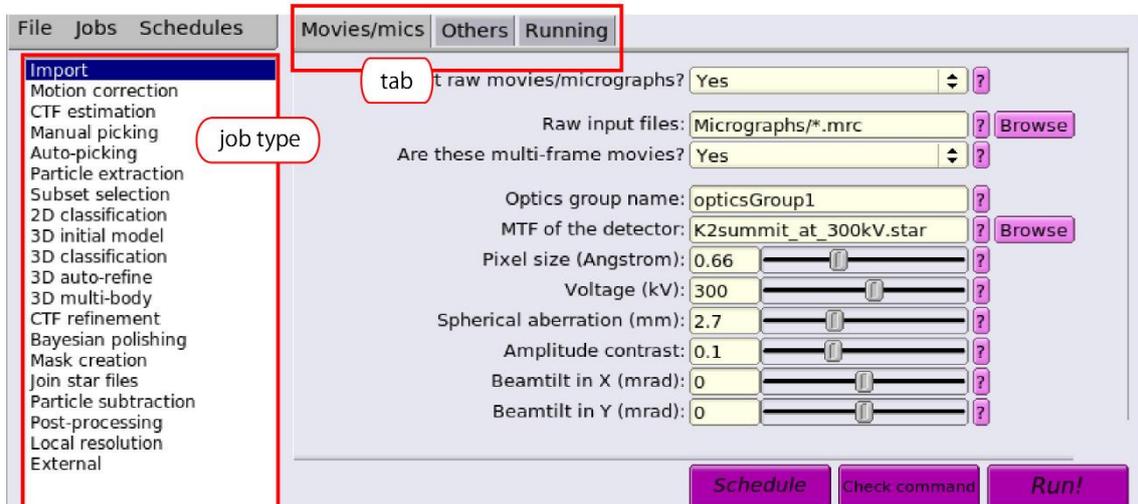


図 1.1 : Import 画面

選択した 196 枚の画像はすべて Micrographs ディレクトリへ置いていますので、左側の job-type ブラウザー(図 1)の **Import** をクリックすると、図 1.1 の画面へ変化します。図 1.1 の I/O タブ中で

- Raw input files: **Micrographs/*.mrc**
⇒これから処理する電子顕微鏡画像の名前ならびにディレクトリ名に応じて変更。
- Are these multi-frame movies?: **Yes**
⇒処理する電子顕微鏡画像のタイプに応じて変更。モーションコレクションをする場合には、Yes になります。

その他、変更する箇所は

- MTF of the detector: **K2summit_at_300kV.star**
⇒電子顕微鏡画像記録時の検出器・加速電圧に依存した MTF を記述したファイル。
Gatan 社の K2(300kV)の場合には、下記URLを参照してください

https://www3.mrc-lmb.cam.ac.uk/relion//index.php?title=K2-summit_at_300kV

Thermo Fisher 社の Falcon III(300kV)の場合には、下記URLを参照してください
https://www3.mrc-lmb.cam.ac.uk/relion//index.php?title=Falcon-III_in_electron-counting_mode_at_300kV

- Pixel size (Angstrom): **0.66**
⇒実習で使用する画像は、K2 (super resolution)モードで記録されています。
- Voltage (kV): **300**
⇒加速電圧は、電子顕微鏡ごとに異なります。
- Spherical aberration (mm): **2.7**

⇒球面収差は、電子顕微鏡ごとに異なります。Titan Krios, CRYO-ARM は、2.7で同じ。

続いて **Running** タブを押して、タブ右下の **Run!** ボタンを押します。

以上で、画像のインポート処理が終了です。

1.2 モーション補正

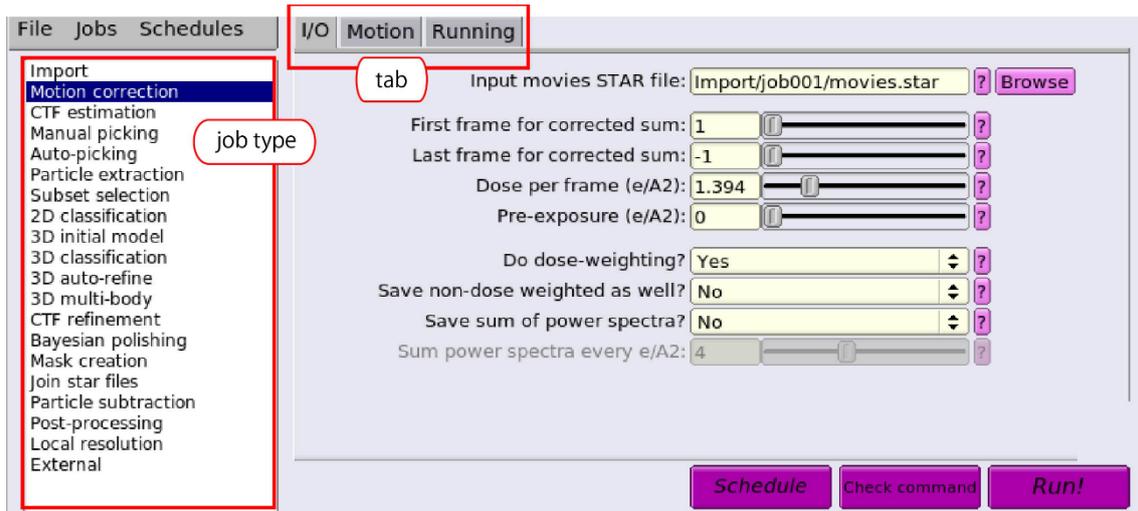


図 1.2 : Motion correction

左側の job-type ブラウザーの **Motion correction** をクリックすると、図 1.2 の画面へ変化します。図 1.2 の I/O タブ中を次のように変更します。

・ Input movies STAR file: **Import/job001/movies.star**

⇒1.1 でインポートした電子顕微鏡画像ファイルリスト (**job+3ケタ数字は適宜変更**)

注意：以降の入力部分において job001 のように job+3ケタ数字がでてくる表記に関しては、ご自分が行われた操作手順によって数字部分は異なります。適宜適切なものを選んでください。

・ First frame for corrected sum: **1**

⇒使用するムービー画像の開始位置を指定します。1 なら最初からだが、ビームブランキングの影響が大きいため、2 に変更する場合があります。最終的に Bayesian polishing で1番目から使用することを指定できるので、あまり気にしなくても大丈夫です。

・ Last frame for corrected sum: **-1**

⇒使用するムービー画像の最後位置を指定します。-1 なら最後まで。こちらも最終的に Bayesian polishing で後ろのフレームを削ることができるので、あまり気にしない。

・ Dose per frame (e/A2): **1.394**

⇒撮影条件ごとに異なります。

・ Pre-exposure (e/A2): **0**

⇒撮影条件ごとに異なりますが、今回は検出器で記録する前に試料への照射は行ってい

ませんので、0 になります。

• Do dose-weighting?: Yes

⇒照射電子線量に依存してフレームに重みをつけます。見た目 S/N 比は悪くなりますが、高分解能情報が保存されますので、後の処理で高分解を目指せます。

• Save no-dose weighted as well?: No

⇒上記の dose-weight を選択すると、次のステップ CTF 評価時に若干悪影響を及ぼすことが知られています。そのため Save:Yes を選べば dose-weight していない画像も保存され、そちらを使用することもできます。ただし、通常はほとんど影響がないことが分かっていますし、無駄にディスク使用量や計算時間を増やすことにつながるため、No を選択します。

Motion タブ中を次のように変更します。

• Bfactor: 150

⇒各フレーム画像に温度因子を適用してフレーム間のずれ位置を決定します。デフォルト値のままです。

• Number of patches X, Y: 5

⇒一旦各フレーム画像のずれ位置を計算したのち、フレーム画像を指定する数値に分割して、パッチ状の領域定義を行ったのち、更に各領域でのずれ位置を計算する。1 の場合には、パッチ状分割計算はしません。Bayesian polishing をすれば同じだと主張する人もいますが、経験上 3~5 分割しておくのがよいようです。

• Group frames: 1

⇒各フレーム間のずれ位置を計算する際に、1 フレームでは SN 比が低くて計算結果が芳しくないこともあります。その際、数値を増やして複数フレームを足しこんだ中間画像同士で計算することでうまくいくことがあります。今回もその一例です。

• Binning factor: 1

⇒出力時に binning するかどうか。1 ならそのまま。数値に依存した binning を行う。その際、フーリエ空間で情報を切り出すため(fourier cropping)、実空間で行う binning のようなエリアジグは生じない。あとで Bayesian polishing を行う予定なら 1 にする。

• Gain-reference image:

⇒次の 4 項目を含めた項目は、規格化が行われていない画像を使用する場合のオプションになります。今回の画像は、規格化された画像としてデータ収集されていますので、次の 4 項目を含めて必要ありません。

• Gain rotation: No rotation (0)

• Gain flip: No flipping (0)

• Defect file:

• Use RELION's own implementation? Yes

⇒UCSF の MOTIONCOR2 を使用する場合には、No になります。CPU の処理能力が

高い場合には RELION に実装されている MOTIONCOR2 を使ったほうが並列化されているため処理時間も短くなりお勧めです。ただし、UCSF のものと少し計算方法が異なる点にご留意ください。

- MOTIONCOR2 executable: `/usr/local/bin/motioncor2`
⇒MOTIONCOR2 バイナリーファイル名を絶対パスで指定します。
- Which GPUs to use:
⇒MOTIONCOR2 を計算する GPU 番号を選びます。空欄のときは自動でアサインされます。
- Other MOTIONCOR2 arguments
⇒MOTIONCOR2 用のパラメータを入れたいときに使用します。

Running タブ中を次のように変更します。

- Number of MPI procs: `4`
⇒使用するコンピューターの CPU/GPU 数に応じて変更。CPU の場合 $\text{num CPUs} = \text{procs} \times \text{threads}$ を目安に。
- Number of threads: `8`
- Submit to queue?: `Yes`

に変えます。

続いて、タブ右下の **Run!** ボタンを押して計算を開始します。

1.3 CTF の評価

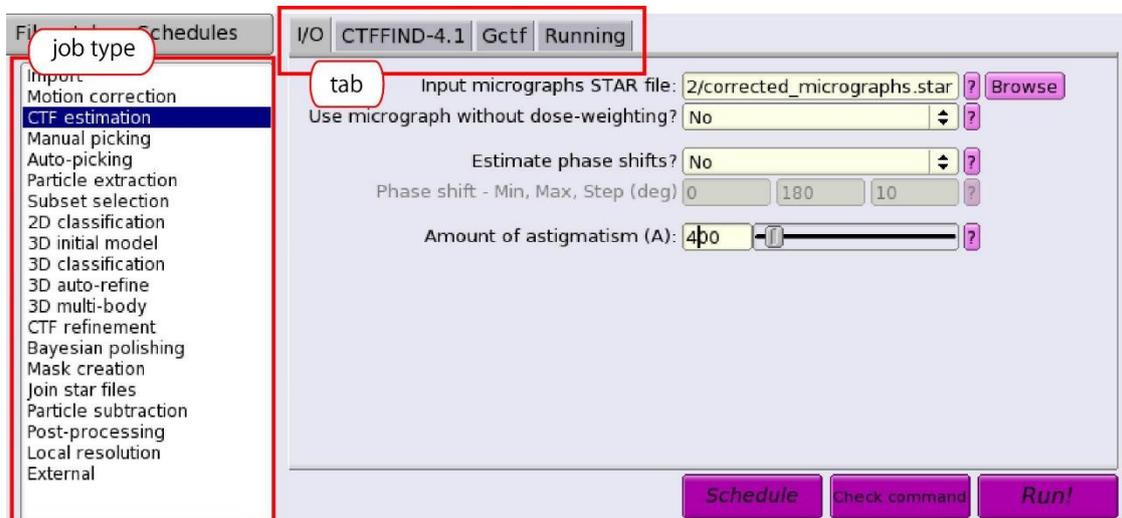


図 1.3 : CTF estimation

左側の job-type ブラウザーの **CTF estimation** をクリックすると、図 1.3 の画面へ変化します。図 1.3 の **I/O** タブ中を次のように変更します。

- Input micrographs STAR file: `Import/job002/corrected_micrographs.star`
⇒1.2 で出力された電子顕微鏡画像ファイルリスト (**job+3ケタ数字は適宜変更**)

- Use micrograph without dose-weighting?: No
⇒dose weight した画像は Thon ring がぼやけるので C T F 評価には適さない。
- Estimate phase shifts?: No
⇒Phase plate を用いて測定した際のみ、Yes になります。
- Amount of astigmatism (A): 400
⇒電子顕微鏡の設定状況に依存して異なりますが、100~500 程度です。

CTFFIND-4.1 タブ中を次のように変更します。GCTF を使用する場合には、ここに記載されている項目は無視します。

- Use CTFFIND-4.1: Yes
- CTFFIND-4.1 executable: /usr/local/bin/ctffind
- Use exhaustive search?: Yes
- Estimate CTF on window size (pix): -1
- FFT box size (pix): 512
- Minimum resolution (A): 30
- Maximum resolution (A): 5
- Minimum defocus value (A): 5000
- Maximum defocus value (A): 50000
- Defocus step size (A): 500

今回の実習では、CTFFIND を使用するので、Gctf タブ中は次のデフォルトのままで構いません。GCTF を使用する場合には、ここに記載されている項目の変更が必要になります。

Running タブ中を次のように変更します。

- Number of MPI procs: 28
⇒使用するコンピューターの core 数に応じて変更。
- Submit to queue?: Yes

に変えます。

続いて、タブ右下の Run! ボタンを押して計算を開始します。

計算後、図 1.4(A)のようにタブ右下の Display: out:micrographs_ctf.star を選択すると、Relion Display GUI 画面 (図 1.4(B)) がポップアップしますので、下記の設定を行ってください。

- Scale: 0.3
⇒使用するコンピューター画面に応じて変更。
- Sort images on : rlnCtfMaxResolution
- Reverse sort?
⇒上記 2 箇所を ON にすると、CTFFIND で評価された分解能順 (低いものから) に表示。

入力後、**Display** を押すと、図 1.4 (C) のように各画像の FFT と CTFFIND で決定した各画像に対応する C T F 理論関数の結果 (第 2 象限) が表示されます。



図 1.4 : CTF estimation—結果

図 1.4(A)のようにタブ右下の **Display: logfile.pdf** を選択すると、evince で logfile.pdf (CTFFIND で決定した C T F の情報や統計値、C T F の減衰から推定される分解能等) が表示されます。

1.4 CTF 評価を参照した画像選択

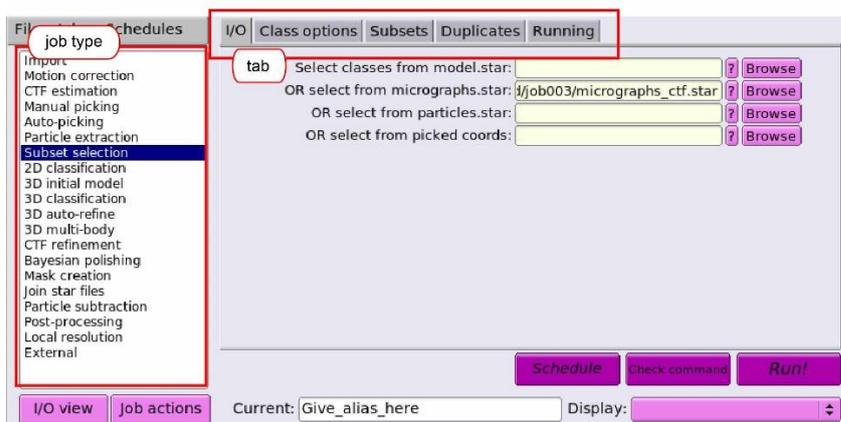


図 1.5 : 画像選択 (1)

左側の job-type ブラウザーの **Subset selection** をクリックすると、図 1.5 の画面へ変化します。図 1.5 の I/O タブ中を次のように変更します。

- OR select from micrographs.star: **CtfFind/job003/micrographs_ctf.star**

⇒1.3 で出力された C T F 評価後の電子顕微鏡画像リスト (**job+ 3 ケタ数字は適宜変更**)

Class options, Subsets, Duplicates タブ中は、すべて “No”であることを確認します。その後、Running タブ右下の **Run!** ボタンを押してマニュアルでの画像選択を開始します。

図 1.6(A)のような Relion Display GUI 画面がポップアップしますので、下記の設定を行ってください。

・ Scale: **0.3**

⇒使用するコンピューター画面に応じて変更。

・ Sort images on : **rlnCtfMaxResolution**

・ Reverse sort?

⇒上記 2 箇所を ON にすると、CTFFIND で評価された分解能順（低いものから）に表示。

入力後、**Display** を押すと、図 1.6 (B) のように各画像の FFT と CTFFIND で決定した各画像に対応する C T F 理論関数の結果（第 2 象限）が表示されます。左クリックで選択、右クリックで “Clear selection”, “Invert selection” など各種メニューが表示されます。分解能に関しては、このメニュー中の “Show metadata” を選べばマウスで現在クリックしている画像情報のみ表示されます。

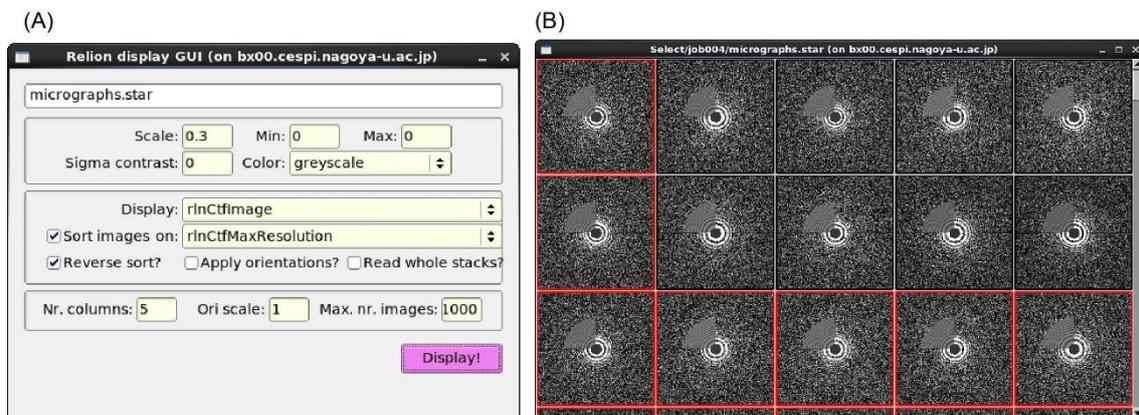


図 1.6 : 画像選択 (2)

2. 粒子のピックアップ

粒子のピックアップとして、relion 中では次の3つ ①マニュアル、②LoG-based を利用した自動、③2Dクラス平均（場合によっては3Dモデル）をテンプレートとして自動、の方法が準備されています。本実習ではオーソドックに、まずマニュアルで500～1000粒子ピックアップして、それらの粒子画像からテンプレートのための2Dクラス平均を計算します。続いて、テンプレートを利用し自動で粒子のピックアップを試みます。そのため今回②に関しては省略します。

2.1 マニュアルでの粒子ピックアップ

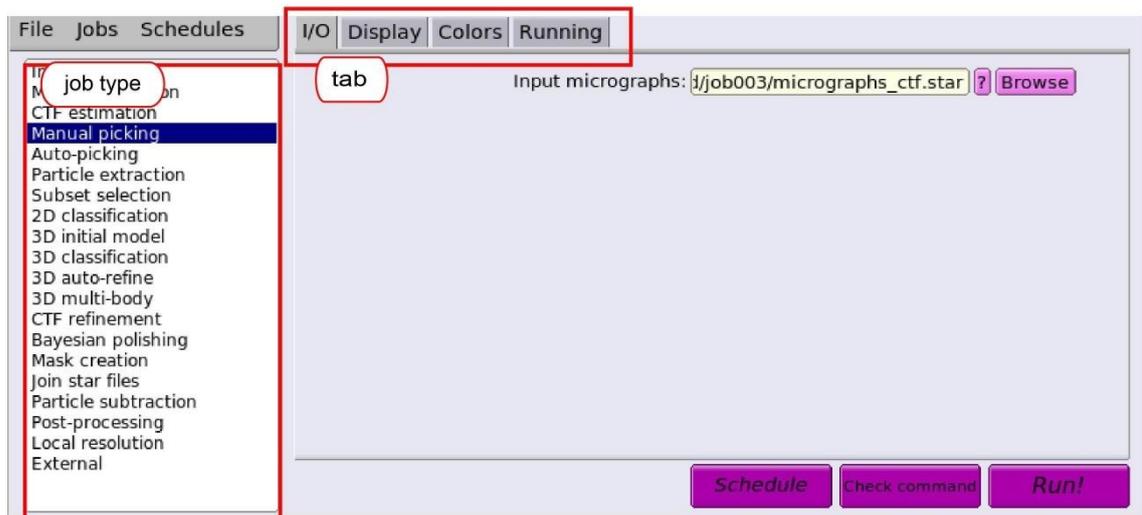


図 2.1 : Manual picking

左側の job-type ブラウザーの **Manual picking** をクリックすると、図 2.1 の画面へ変化します。I/O タブ中を次のように変更します。

・ Input micrographs: **CtfFind/job003/micrographs_ctf.star**

⇒1.3 で C T F の方法を付与した電子顕微鏡画像ファイルリスト (**job+3ケタ数字は適宜変更**) あるいは、1.4 で選択した電子顕微鏡画像ファイルリスト。

Display タブ中を押し次のように変更します。Particle diameter 以外デフォルト設定値。

・ Particle diameter (A): **200**

⇒単に画面上での円の直径表示用としてのみ使用。

・ Scale for micrographs: **0.2**

⇒画面サイズに応じて変更。

・ Sigma contrast: **3**

・ White value: **0**

・ Black value: **0**

・ Lowpass filter (A) **20**

・ Highpass filter (A) **-1**

・ Pixel size (A): -1

⇒粒子サイズの計算やフィルターに影響する。

・ Pick start-end coordinates helices? No

・ Scale for CTF image: 1

その他、Colors タブ中は、デフォルトのまま。続いて Running タブを押して、タブ右下の Run! ボタンを押します。

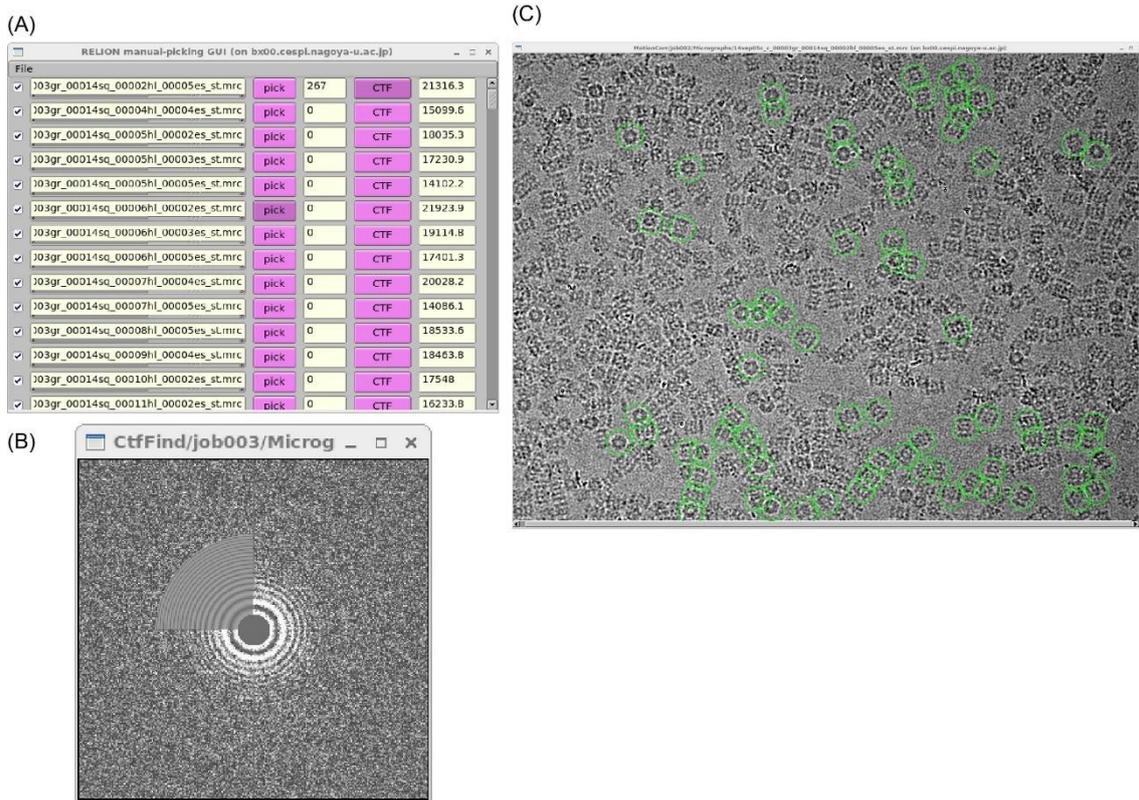


図 2.2 : Manual picking --- Display

すると、図 2.2(A)のような画面がポップアップしてきますので、比較的デフォーカス量（一番右端に表示されている数字）が異なる電子顕微鏡画像を選ぶと粒子が見やすくてお勧めです。図 2.2(A)中の CTF を押すと(B)のように Window が pop-up するので、Thon ring の様子を確認できます。図 2.2(A)の pick を押すと、図 2.2(C)のような顕微鏡画像がでてきますので、粒子をピックアップします。

その際、画面を左クリックすると粒子のピックアップ(緑色の円で表示：図 2.2(C))で、マウスのホイール部分でクリックと選択した粒子のピックアップが解除されます。右クリックによって”Save STAR with coordinates (Ctrl-s)”, “Clear coordinates”, “Quit (Ctrl-q)”などのメニューがでますので、これまでピックアップした粒子の座標の保存などができます。保存を行うと pick の隣に粒子数が表示されます (図 2.2(A))。

2.2 拾った粒子の抽出

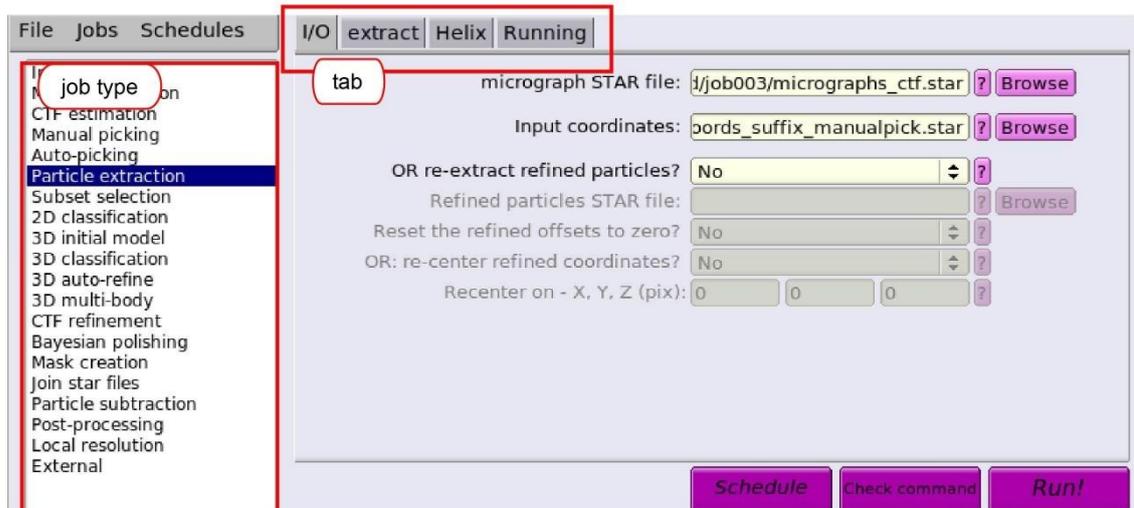


図 2.3 : Manual picking --- Extract

続いてマニュアルでピックアップした粒子画像の抽出を行います。左側の job-type ブラウザーの **Particle extraction** をクリックすると、図 2.3 の画面が表示されます。

I/O タブ中を次のように変更します。

- micrograph STAR file: **CtfFind/job003/micrographs_ctf.star**
⇒1.2 で C T F の方法を付与した電子顕微鏡画像ファイルリスト (**job+ 3 ケタ数字は適宜変更**)。あるいは、1.4 で選択した電子顕微鏡画像ファイルリスト。
- Input coordinates: **ManualPick/job004/cords_suffix_manualpick.star**
⇒2.1 でピックアップした粒子画像のファイルリスト (**job+ 3 ケタ数字は適宜変更**)
- OR re-extract refined particles? **No**

extract タブ中を押して、次のように変更します。

- Particle box size (pix): **360**
⇒サイズは必ず偶数にする必要があります。
- Invert contrast? **Yes**
⇒粒子が白に見える必要があります。
- Normalize particles? **Yes**
- Diameter background circle (pix): **-1**
⇒-1:自動的に計算。通常は、Particle box size の 75%程度。この円より外の範囲をバックグラウンドとする。
- Stddev for white dust removal: **-1**
- Stddev for black dust removal: **-1**
- Rescale particles?: **Yes**
⇒当初はスケールダウンしたほうが計算速度も速いのでお勧め。
- Rescaled size (pixels): **90**

⇒テンプレートとして使用するだけなので、低分解能情報のみでよいため今回は binning 4 として Rescaled size 90 pix = Particle Box size 360 pix / 4 で設定しました。サイ

ズは必ず偶数にする必要があります^{注)}。

^{注)} 多くの計算過程で高速フーリエ変換(FFT)を繰り返し利用しています。そのためサイズ数は、素因数分解した際に、できるだけ小さな素数(基数)の累乗で表現できるようにすると効率がよくなります。2以外の素数の累乗を使用するときは、偶数にするため2の累乗も構成要素にすることを忘れずに。

Helix タブ中は、変更なし。

Running タブ中は、次のようにします。

- Number of MPI procs: 1
- Submit to queue: no

続いてタブ右下の **Run!** ボタンを押すと粒子の抽出が始まります。

2.3 粒子画像のテンプレート作成

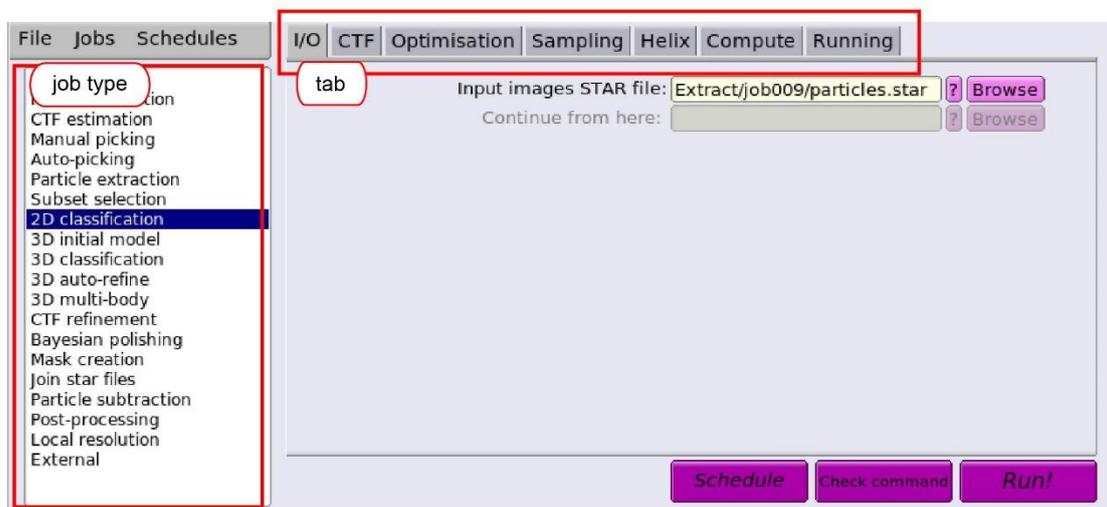


図 2.4 : Manual picking --- making template

ここでは、2D クラス平均を利用して自動ピックアップ用のテンプレートを作成します。左側の job-type ブラウザーの **2D classification** をクリックすると、図 2.4 の画面へ変化します。

I/O タブ中を次のように変更します。

- Input images STAR file: **Extract/job009/particles.star**
⇒ 2.2 で抽出後のマニュアルピックアップした粒子像リスト (**job+3ケタ数字は適宜変更**)

CTF タブ中を次のように変更します。

- Do CTF-correction?: **Yes**
⇒ 計算中に CTF 補正(位相+振幅)を行います。通常は、Yes。
- Ignore CTFs until first peak?: **No**
⇒ CTF の第一ピークまでの情報を無視するかどうか? 通常は、No。但し、CTF の振幅情報の補正によって、低分解能が強調されすぎて 2D クラス分類がうまくいかないときには、Yes を選択するほうがよいでしょう。

Optimization タブ中を次のように変更します。

• Number of classes: 16

⇒拾った粒子の数によって変更する必要があります。粒子の数に比較して、クラスの数が多いと、クラス平均は細かく分類されます。通常は、100にしますが、今回はピックアップ用のテンプレートとして使用するため、細かい特徴まで分類する必要がありません。その上、粒子の数も少ないので16にしています。

• Regularization parameter T: 1

⇒クライオ電子顕微鏡由来のデータの場合、2D クラス分類では T=2-3、3D クラス分類では T=3-4 で計算を行います。また、T 値が小さいほど低分解能の特徴が見え、逆に T 値が大きいと高分解能の特徴が見えてきます。T 値が大きい場合には、オーバーフィッティングになることもあるため結果には気を付ける必要があります。今回はテンプレート作製のため T=1 と大まかな分類で収束するようにしています。

• Number of iterations: 25

⇒通常は、25 を使用します。

• Use fast subsets for large data sets? No

⇒粒子が1万個程度のときは、No を使用します。データ数が数十万以上の時には、Yes にすると計算が早く終わります。

• Mask diameter (A): 200

⇒タンパク質の大きさに応じて変更。溶媒領域や隣の分子をマスクアウトすることができます。ノイズの多い画像や粒子が込み合っている場合には、できるだけ小さくするほうが効果的ですが、分子の最も長い方向がカバーされる程度にしないと一部が欠けたクラス平均像を得ることになってしまいます。

• Mask individual particles with zeros? Yes

⇒マスクアウトした領域を0値で埋めるかどうか？

• Limit resolution E-step to (A): -1

⇒正の数値をセットすると、その数値内までの周波数情報を使用して画像のアラインメントを行います。通常は、-1 でよいのですが、オーバーフィッティングが見られる際に使用すること効果的です。

Sampling タブ中に変更しませんので、以下のデフォルトであることを確認。

• Perform image alignment? Yes

• In-plane angular sampling: 6

⇒正二十面体のウイルスのように対称性が高い場合には、小さい値を使用することもあります。

• Offset search range (pix): 5

• Offset search step (pix): 1

Hexlix タブ中に変更しませんので、以下のデフォルトであることを確認。

• Classify 2D helical segments? No

Compute タブ中では、次のように変更します。

- Use parallel disc I/O: Yes
- Number of pooled particles: 30
- Skip padding?: No
- Pre-read all particles into RAM?: No
- Copy particles to scratch directory: /ssd
⇒実習で使用する計算機では、計算用スラッチディスクとして/ssd を使用しています。用意していない場合には、ここは空欄にすること。
- Combine iterations through disk?: No
- Use GPU acceleration?: Yes
⇒Yes:GPU を使った計算を行う場合。No:CPU のみの計算を行う場合
- Which GPU to use:
⇒空欄の場合、デフォルトとして使用できるGPU全部をアサインしてくれます。

Running タブ中で、次のように変更します。

- Number of MPI procs: 5
- Number of threads: 4
- Submit to queue?: Yes
- Queue name: openmpi
- Queue submit command: qsub
- Standard submission script: /usr/local/relion/bin/qsub.csh
- Minimum dedicated cores per node: 1
- Additional arguments:

⇒上記の変数は、今回の実習環境になっています。計算機環境に応じて変更してください。

続いて、タブ右下の Run ボタンを押すと計算が始まります。

計算後、2Dクラス平均を確認するために、図 2.5(A)のようにタブ右下の Display: out:run_it015_model.star を選択すると、図 2.5(B)の画面がポップアップします。

- Scale: 0.5
⇒使用するコンピューター画面に応じて変更。
- Sort images on : rlnClassDistribution
- Reverse sort?

図 2.5(B)への入力後、Display を押すと、図 2.5(C)のようにクラス平均画像が、クラスに含まれる粒子数の多いものが上に現れるよう表示されます。

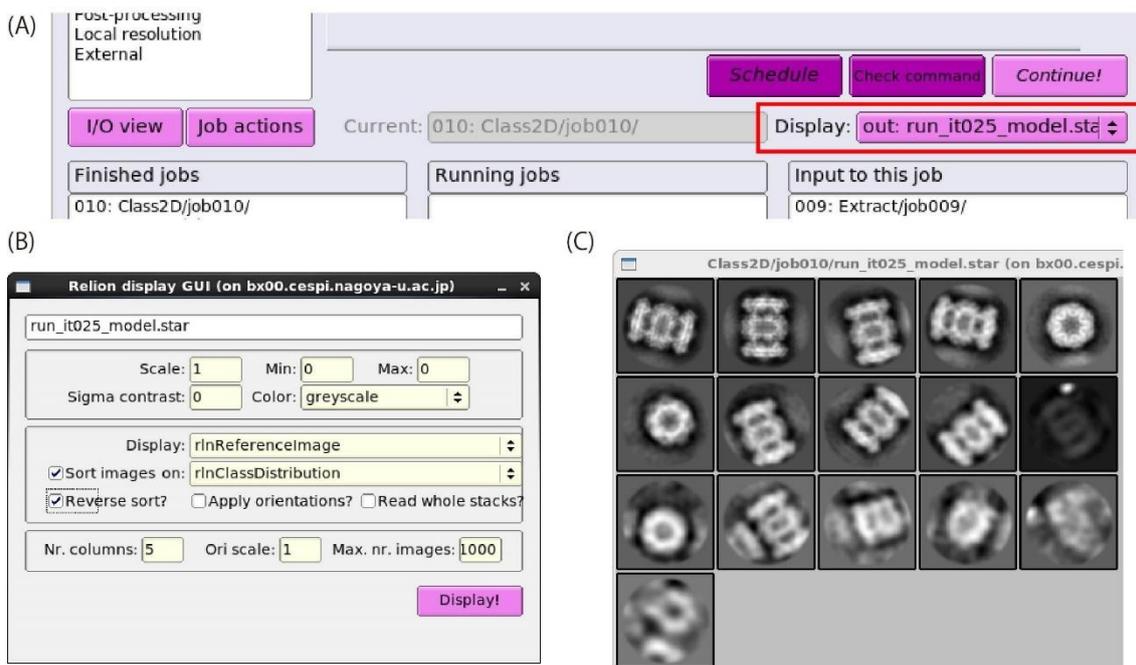


図 2.5 : Manual picking --- display template

2.4 粒子画像のテンプレート選択

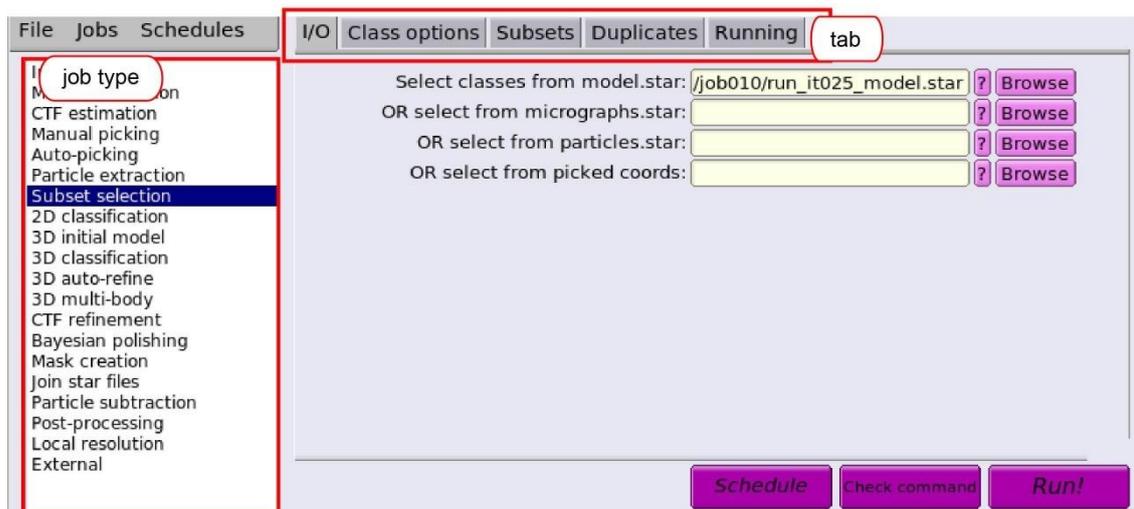


図 2.6 : Subset selection

2.3 で計算した自動ピックアップ用のテンプレートですが、実際に確認してもらった通り様々なクラス平均像が混じっています。そこでテンプレートとして適切な画像のみ選択する操作が必要になります。左側の job-type ブラウザーの **Subset selection** をクリックすると、画面が変化します (図 2.6)。

I/O タブ中を次のように変更します。

- Input images STAR file: **Class2D/job010/run_it025_model.star**

⇒2.3 で計算した 2D クラス平均のモデルリストファイル (**job+3ケタ数字は適宜変更**)

Class options タブ中で、次のように変更します。

- Re-center the class averages?: Yes
⇒ 2Dクラスで計算した画像の中心と重心位置を一致させてくれる。
- Regroup the particles: No

Subsets タブ中で、次のように変更します。

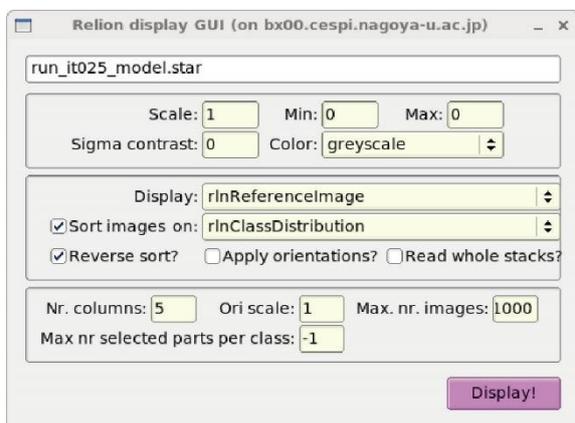
- Select based on metadata values?: No
- OR: select on image statistics?: No
- OR: split into subsets?: No

Duplicates タブ中で、次のように変更します。

- OR: remove duplicates?: No
⇒ 粒子リストファイル particles.star のみ使用できる。

続いて Running タブを押して、タブ右下の Run! ボタンを押します。

(A)



(B)

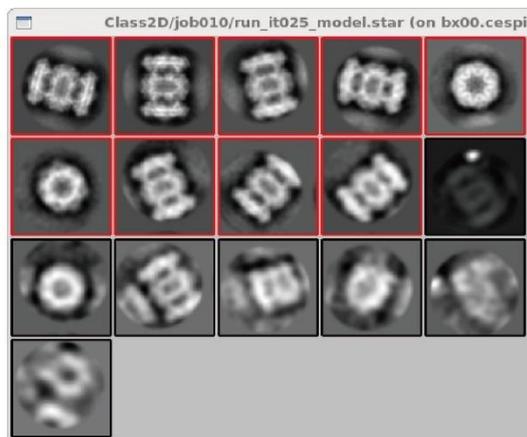


図 2.7 : Subset selection –Running

すると、図 2.7(A)の画面がポップアップします。

- Scale: 0.5
⇒使用するコンピューター画面に応じて変更。
- Sort images on : rlnClassDistribution
- Reverse sort?

図 2.7(A)への入力後、Display を押すと、図 2.7(B)のようにクラス平均画像が、クラスに含まれる粒子数の多いものが上に現れるよう表示されます。画面上で左クリックすると、マウスがある位置にあるクラスを選択することができ、クラス画像の周囲に赤い囲み線が表示されます。同じクラス画像を再度左クリックすると、選択を外すことができ、赤い囲み線が消えます。最終的にきれいなクラス平均だけを選択し、右クリックし、“Save selected classes”を選ぶと選択したクラスに含まれる画像全部がファイルリストして保存されます。

2.5 テンプレートを利用した自動粒子検出

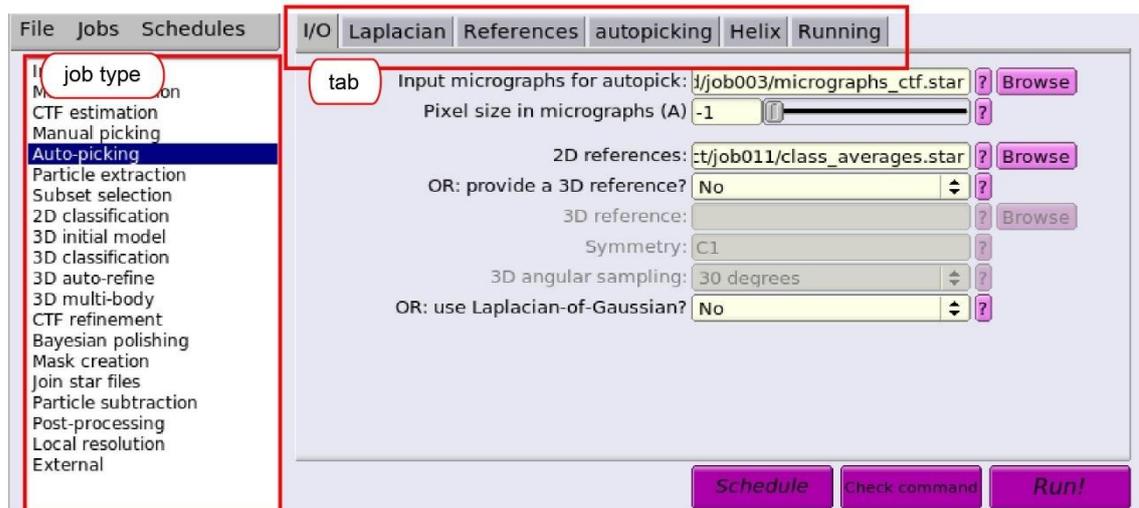


図 2.8 : Auto-picking

さきほど作製したテンプレートを利用して、電子顕微鏡画像から自動で粒子をピックアップする計算を行います。左側の job-type ブラウザーの **Auto-picking** をクリックすると、図 2.8 の画面へ変化します。

I/O タブ中を次のように変更します。

- Input micrographs for autopick: **CtfFind/job003/micrographs_ctf.star**
⇒1.2 で C T F の方法を付与した電子顕微鏡画像ファイルリスト (**job+3ケタ数字は適宜変更**)。あるいは、1.4 で選択した電子顕微鏡画像ファイルリスト。
- Pixel size in micrographs (A): **-1**
⇒-1 のときは、上記リスト中の情報から読み込みます。
- 2D References: **Select/job011/class_averages.star**
⇒2.4 で選択した 2D クラス平均のファイルリスト (**job+3ケタ数字は適宜変更**)
- OR: provide a 3D reference: **No**
⇒今回 2D クラス平均を使っているため。
- OR: use a laplacian-of-Gaussian?: **No**
⇒今回 2D クラス平均を使っているため。

Laplacian タブ中は、今回使用しないので変更なし。

References タブ中は、

- Lowpass filter references (A) **20**
- Highpass filter (A) **-1**
- Pixel size in references (A) **-1**
- Mask diameter (A) **200**
⇒サンプルごとに変更してください。
- In-plane angular sampling (deg) **5**
⇒ほとんどのケースにおいて、デフォルト値でうまく計算できます。
- References have inverted contrast ? **Yes**

⇒元の顕微鏡画像では、粒子が黒色に見えますが、参照とするテンプレートは粒子が白色になるよう補正されているため。

・ Are references CTF corrected? **Yes**

⇒参照とするテンプレートは2Dクラス平均計算時にCTF補正されているため。

・ Ignore CTFs until first peak? **No**

⇒テンプレートの2Dクラス平均計算時に使用したオプションと同じものを使用します。

Autopicking タブ中は、

・ Picking threshold: **0.8**

⇒FOM map 中で、この FOM 値を下回るものはピックアップされません。デフォルト 0.05。サンプルごとに変わりますが、後で述べるように数回繰り返して決めます。

・ Minimum inter-particle distance (A): **100**

⇒ピックアップされる粒子間で許される最小近接距離。通常は、粒子直径の 50-60%程度が適切。

・ Maximum stddev noise: **1.1**

⇒カーボン支持膜やコンタミネーションの場合、stddev が大きいことを利用してピックアップしないようにできます。通常 1.0-1.2 を使用する。-1 を入力すれば、この機能を OFF にすることができます。

・ Minimum avg noise: **-999**

⇒カーボン支持膜やコンタミネーションの場合、この値を利用することでピックアップしないようにすることができます。通常-0.5-0.0 を使用します。-999 を入力すれば、この機能を OFF にすることができます。

・ Write FOM maps? **Yes**

⇒最初に FOM maps を計算する必要があります。2回目以降は No にします。

・ Read FOM maps? **No**

⇒一度計算した FOM maps を使ってピックアップ時の適切な変数を探索する際に使用します。今回は、最初の FOM 計算なので、No になります。2回目以降は Yes にします。

・ Shrink factor: **0**

⇒縮小率 0~1 の数値。1 の場合、そのまま。0 の場合、Reference 画像に合わせて縮小してくれます。今回は、Reference 画像を縮小していないので、0 でも 1 でも同じ処理になります。

・ Use GPU acceleration? **Yes**

⇒Yes:GPU を使った計算を行う場合。No:CPU のみの計算を行う場合

・ Which GPU to use:

⇒空欄の場合、デフォルトとして使用できる GPU 全部をアサインしてくれます。

Helix タブ中は、今回使用しないので変更なし。

Running タブ中で、次のように変更します。

・ Number of MPI procs: **4**

- Submit to queue?: Yes
- Queue name: openmpi
- Queue submit command: qsub
- Standard submission script: /usr/local/relion/bin/qsub.csh
- Additional arguments: --no_fom_limit

⇒上記の変数は、今回の実習環境になっています。計算機環境に応じて変更してください。

続いて Running タブを押して、タブ右下の Run! ボタンを押すと計算が始まります。

(A)



(B)

File	pick	CTF	21316.3
✓ J03gr_00014sq_00002h_00005es_st.mrc	pick	647	CTF
✓ J03gr_00014sq_00004h_00004cs_st.mrc	pick	648	CTF
✓ J03gr_00014sq_00005h_00002es_st.mrc	pick	581	CTF
✓ J03gr_00014sq_00005h_00003cs_st.mrc	pick	660	CTF
✓ J03gr_00014sq_00005h_00005es_st.mrc	pick	642	CTF
✓ J03gr_00014sq_00006h_00002es_st.mrc	pick	695	CTF
✓ J03gr_00014sq_00006h_00003es_st.mrc	pick	670	CTF
✓ J03gr_00014sq_00006h_00005es_st.mrc	pick	650	CTF
✓ J03gr_00014sq_00007h_00004es_st.mrc	pick	615	CTF
✓ J03gr_00014sq_00007h_00005es_st.mrc	pick	690	CTF
✓ J03gr_00014sq_00008h_00005es_st.mrc	pick	589	CTF
✓ J03gr_00014sq_00009h_00004es_st.mrc	pick	664	CTF
✓ J03gr_00014sq_00010h_00002es_st.mrc	pick	660	CTF
✓ J03gr_00014sq_00011h_00002cs_st.mrc	pick	577	CTF

(C)

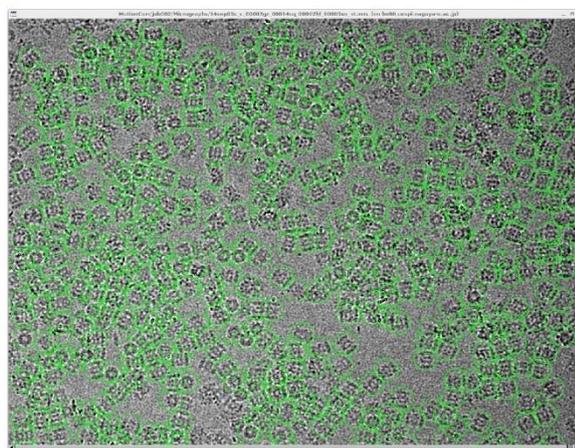


図 2.9 : Auto-picking ---Display

計算終了後に、どのような粒子をピックアップしているかを確認します。図 2.9(A)のようにタブ右下の Display: out:coords_suffix_autopick.star を選択すると、図 2.9(B)の画面がポップアップします。確認したい画像リストの右側にある pick を押すと、図 2.9(C)のような顕微鏡画像がでてきます。ここでは確認のためですが、実際にマニュアルでピックアップ操作も行えます。その際、画面を左クリックすると粒子のピックアップ(緑色の円で表示：図 2.9(B))で、マウスのホイールでクリックすると粒子の選択が解除されます。右クリックで、“Save STAR with coordinates (Ctrl-s)”, “Clear coordinates”, “Quit (Ctrl-q)”などのメニューがでますので、変更した粒子情報の保存などができます。保存を行うと pick の隣に粒子数が表示されます (図 2.9(B))。

粒子以外のものが多い時には、Autopicking タブ中の Picking threshold: 0.8 を大きい値へ、拾えている粒子が少ない場合には、Picking threshold: 0.8 を小さい値に変更したうえで、

Write FOM maps? No Read FOM maps? Yes にして、Running タブ右下の Continue! ボタンを押して再計算してみてください。

2.6 自動検出した粒子の抽出

続いて 2.2 の操作と同様に自動ピックアップした粒子画像の抽出を行います。

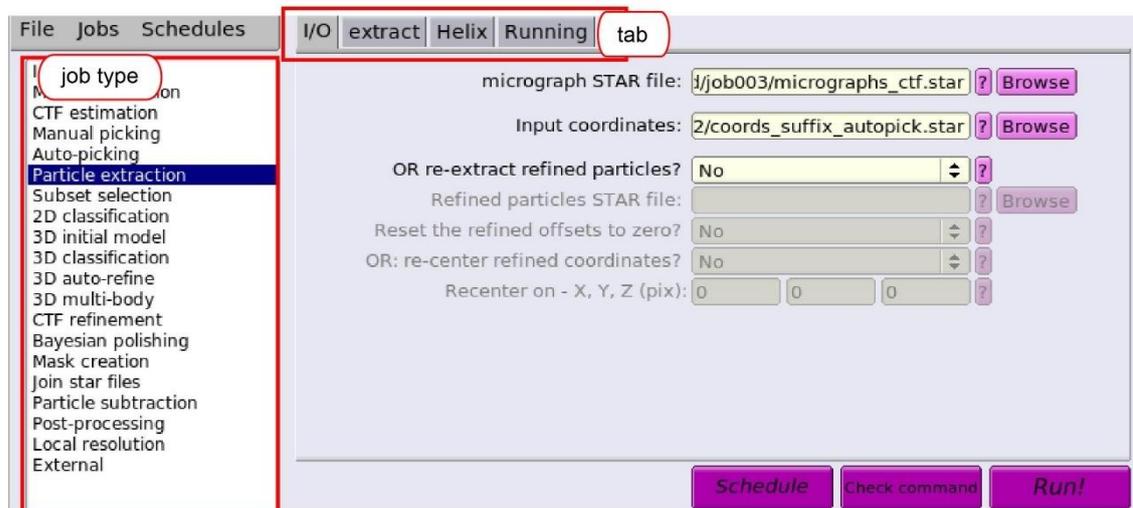


図 2.10 : Extract ----Auto picking

左側の job-type ブラウザーの **Particle extraction** をクリックすると、図 2.10 の画面が表示されます。

I/O タブ中を次のように変更します。

- micrograph STAR file: **CtfFind/job003/micrographs_ctf.star**
⇒ 1.2 で CTF の方法を付与した電子顕微鏡画像ファイルリスト (**job+3ケタ数字は適宜変更**)。あるいは、1.4 で選択した電子顕微鏡画像ファイルリスト。

- Input coordinates: **AutoPick/job012/coords_suffix_autopick.star**
⇒ 2.8 でピックアップした粒子画像のファイルリスト (**job+3ケタ数字は適宜変更**)

- OR re-extract refined particles? **No**
- Manually set pixel size? **No**

extract タブ中を押して、次のように変更します。

- Particle box size (pix): **512**
⇒ サイズは必ず偶数にする必要があります。
- Invert contrast? **Yes**
⇒ 粒子が白に見える必要があります。
- Normalize particles? **Yes**
- Diameter background circle (pix): **-1**
⇒ -1:自動的に計算。Particle box size の 75%程度が望ましく、この円より外の範囲をバックグラウンドとします。
- Stddev for white dust removal: **-1**
- Stddev for black dust removal: **-1**
- Rescale particles?: **Yes**

⇒当初はスケールダウンしたほうが計算速度も速いのでお勧め。

・ Rescaled size (pixels): 128

⇒ 2 Dクラス平均や3 Dクラス分類など粒子の区別をただけなので、低分解能情報のみでよいため今回は $512/128=4 \times 4$ binning で設定しました。サイズは必ず偶数にする必要があります。

Helix タブ中は、変更なし。

Running タブ中は、次のようにします。

・ Number of MPI procs: 1

・ Submit to queue: no

続いてタブ右下の Run! ボタンを押すと粒子の抽出が始まります。

3. 2D クラス平均

レファレンスを必要としない2Dクラス平均を利用することで、ノイズ・氷・壊れた粒子といった不適切な粒子画像を除くことができます。この方法は、2.3の自動で粒子の位置を検出する際のテンプレート作成にも使用しました。1つのクラスに属す粒子数が少ないときには、平均化によってバックグラウンドが平坦になることはないため、汚いクラス平均が生み出され簡単に見分けがつかず。そういったクラスに属する粒子像を捨てていけば、クオリティの高い粒子画像データにすることができます。

3.1 2Dクラス平均の作成

左側の job-type ブラウザーの **2D classification** をクリックすると、図 3.1 の画面へ変化します。

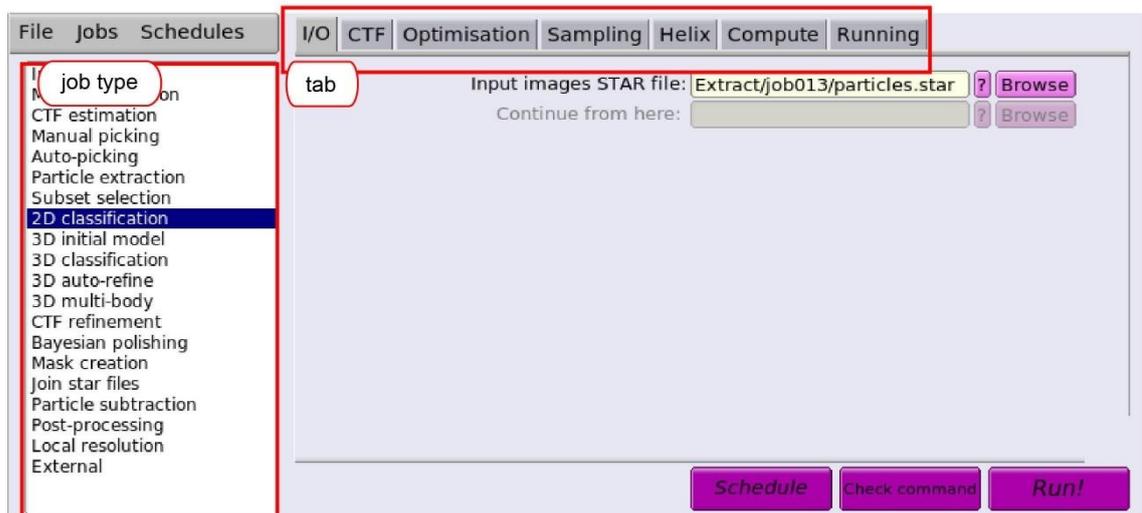


図 3.1 : 2D Classification

I/O タブ中を次のように変更します。

- Input images STAR file: **Extract/job013/particles.star**
⇒2.8 でソート後選択した画像リストファイル (**job+ 3 ケタ数字は適宜変更**)

CTF タブ中は、デフォルトである次の状態を確認すること。

- Do CTF-correction: **Yes**
- Do CTF-correction: **No**

Optimization タブ中は、次のように変更すること。

- Number of classes: **100**
⇒多くの場合、100 ぐらいが適切です。但し、トータルの粒子数が少ないときは、減らしたほうがよい。
- Regulation parameter T: **2**
⇒クライオ画像の場合、T=2~3 が適切。数が大きいほど、詳細な平均像計算になります。
- Number of iterations: **25**

⇒計算を繰り返すサイクル数。

• Use fast subsets: Yes

⇒粒子数が数万個程度のとき、計算の高速化が見込める。

• Mask diameter (A): 200

⇒分子の大きさに応じて。

• Mask individual particles with zeros?: Yes

• Limit resolution E-step to (A): -1

⇒正の数値をセットすると、その数値内までの周波数情報を使用して画像のアラインメントを行います。通常は、-1 でよいのですが、オーバーフィッティングが見られる際に使用すること効果的です。

Sampling タブ中では変更しませんので、以下のデフォルトであることを確認。

• Perform image alignment ? Yes

• In-plane angular sampling: 6

⇒正二十面体のウィルスのように対称性が高い場合には、小さい値を使用することもあります。

• Offset search range (pix) : 5

• Offset search step (pix) : 1

Hexlix タブ中では変更しませんので、以下のデフォルトであることを確認。

• Classify 2D helical segments ? No

Compute タブ中では、次のように変更します。

• Use parallel disc I/O: Yes

• Number of pooled particles: 30

• Skip padding?: No

• Pre-read all particles into RAM?: No

• Copy particles to scratch directory: /ssd

⇒実習で使用する計算機では、計算用スラッチディスクとして/ssd を使用していません。用意していない場合には、ここは空欄にすること。

• Combine iterations through disk?: No

• Use GPU acceleration?: Yes

⇒Yes:GPU を使った計算を行う場合。No:CPU のみの計算を行う場合

• Which GPU to use:

⇒空欄の場合、デフォルトとして使用できるGPU全部をアサインしてくれます。

Running タブ中で、次のように変更します。

• Number of MPI procs: 5

• Number of threads: 4

• Submit to queue?: Yes

• Queue name: openmpi

- Queue submit command: qsub
- Standard submission script: /usr/local/relion/bin/qsub.csh
- Minimum dedicated cores per node: 1
- Additional arguments:

⇒上記の変数は、今回の実習環境になっています。計算機環境に応じて変更してください。

続いて、タブ右下の **Run** ボタンを押すと計算が始まります。

計算後、2Dクラス平均を確認するために、図 3.2(A)のようにタブ右下の **Display:** **out:run_it025_model.star** を選択すると、図 3.2(B)の画面がポップアップします。

- Scale: 1
⇒使用するコンピューター画面に応じて変更。
- Sort images on : rlnClassDistribution
- Reverse sort?

図 3.2(B)への入力後、**Display** を押すと、図 3.2(C)のようにクラス平均画像が、クラスに含まれる粒子数の多いものが上に現れるよう表示されます。

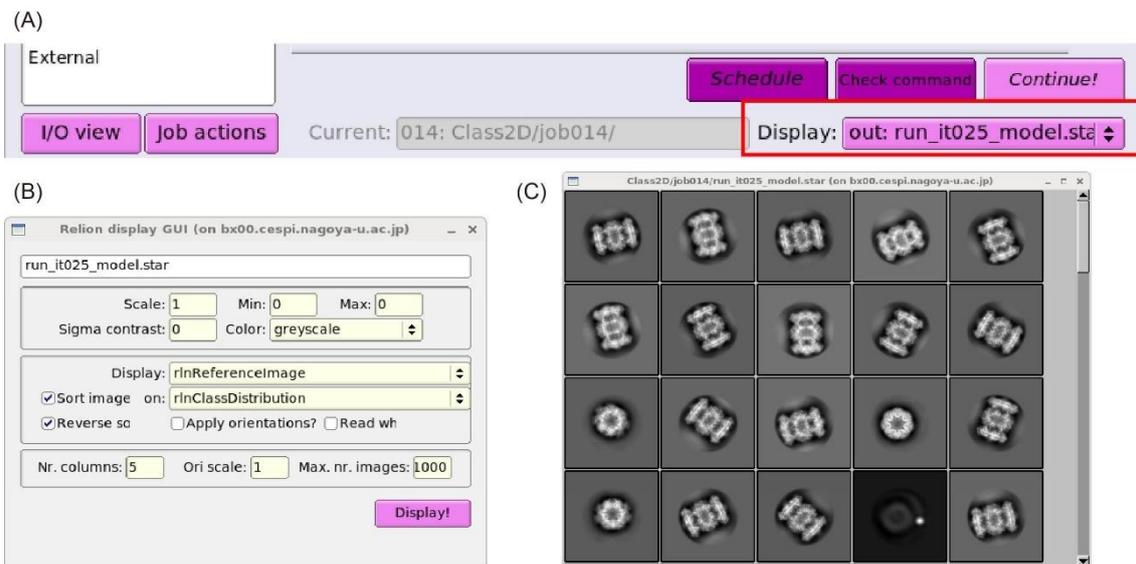


図 3.2 : 2D Classification --- Display

3.2 2Dクラス平均の選択

計算した2Dクラス平均の中から適切な2Dクラス平均のみを選択します。左側の job-type の **Subset selection** をクリックすると、図 3.3 の画面へ変化します。

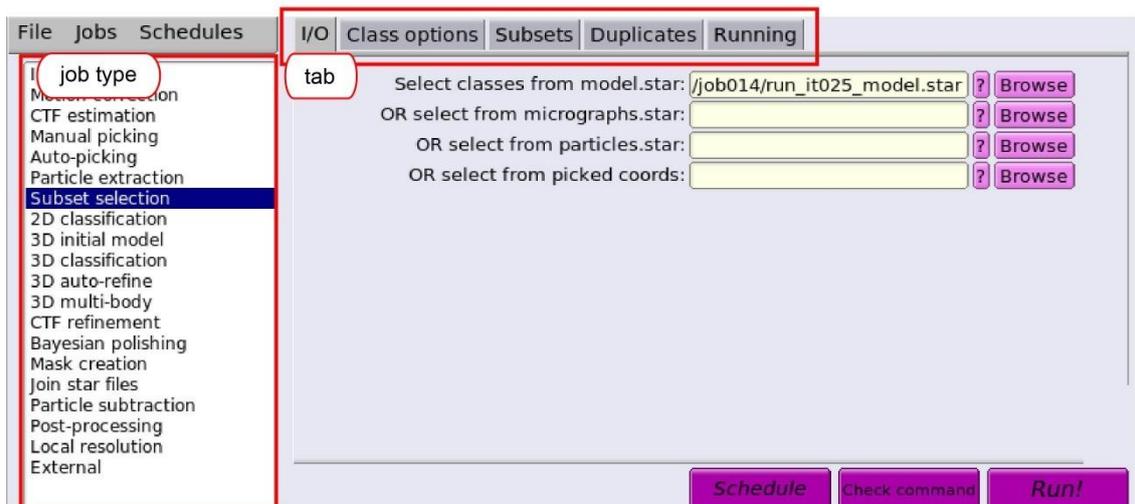


図 3.3 : Subset selection --- 2D Classification

I/O タブ中を次のように変更します。

- Input images STAR file: **Class2D/job014/run_it025_model.star**
⇒3.1 で計算したクラス平均のモデルリストファイル (**job+3ケタ数字は適宜変更**)

Class options タブ中で、次のように変更します。

- Re-center the class averages?: No
- Regroup the particles: No

Subsets タブ中で、次のように変更します。

- Select based on metadata values?: No
- OR: select on image statistics?: No
- OR: split into subsets?: No

Duplicates タブ中で、次のように変更します。

- OR: remove duplicates?: No
⇒粒子リストファイル particles.star のみ使用できる。

Running タブを押して、タブ右下の **Run!** ボタンを押します。

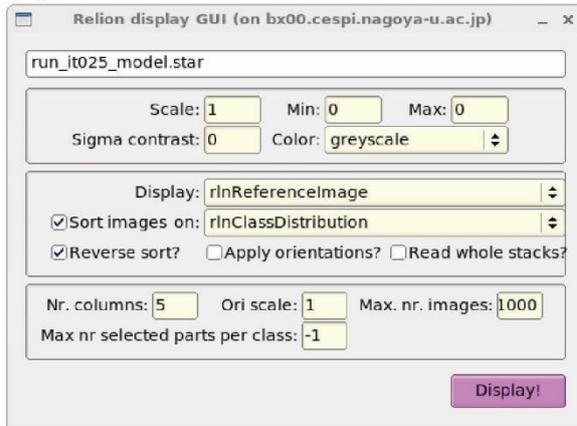
すると、図 3.4(A)の画面がポップアップします。

- Scale: 1
⇒使用するコンピューター画面に応じて変更。
- Sort images on : **rlnClassDistribution**
- Reverse sort?

図 3.4(A)への入力後、**Display** を押すと、図 3.4(B)のようにクラス平均画像が、クラスに含まれる粒子数の多いものが上に現れるよう表示されます。画面上で左クリックすると、マウスがある位置にあるクラスを選択することができ、クラス画像の周囲に赤い囲み線が表示されます。同じクラス画像を再度左クリックすると、選択を外すことができ、赤い囲み線が消えます。最終的にきれいなクラス平均だけを選択し、右クリックし、“Save selected classes”を選ぶと選択したクラスに含まれる画像全部がファイルリストして保存さ

れます。

(A)



(B)

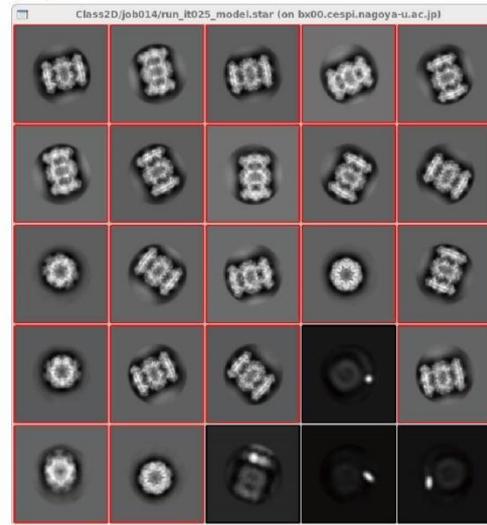


図 3.4 : Subset selection ---GUI

4. *de novo* 3D モデル生成

Relion では、確率的勾配降下法(SGD: Stochastic Gradient Descent)のアルゴリズムを採用することで、3Dモデルを初期モデルの仮定なしに二次元投影像情報だけから生成することができます。この3Dモデルを使って、3Dクラス分類や3D自動精密化を進めることができます。

4.1 新規3Dモデルの作成

左側の job-type ブラウザーの **3D initial model** をクリックすると、図 4.1 の画面へ変化します。

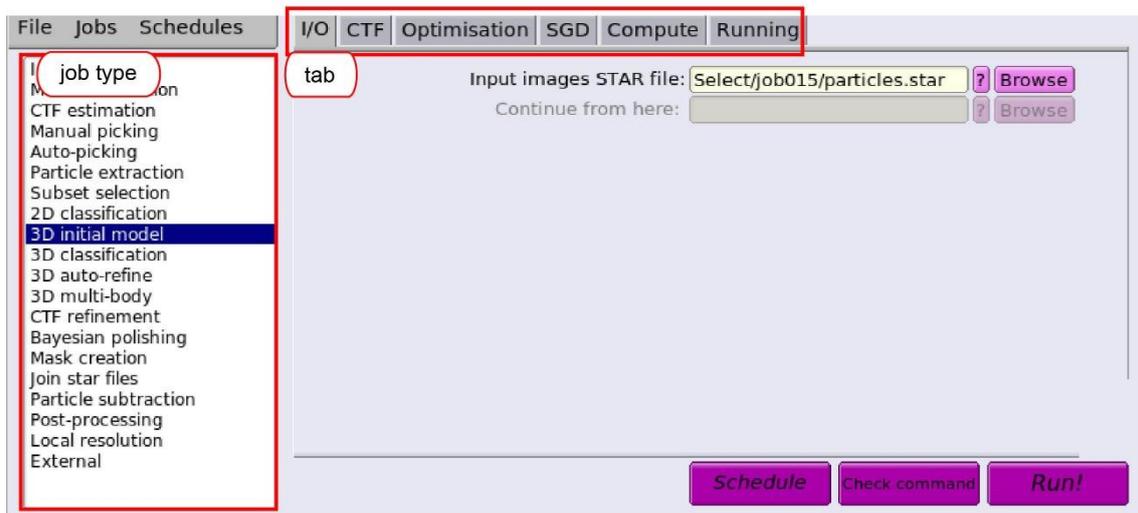


図 4.1 : 3D initial model

I/O タブ中を次のように変更します。

- Input images STAR file: **Select/job015/particles.star**
⇒3.2 で選択した画像リストファイル (**job+3ケタ数字は適宜変更**)

CTF タブ中は、デフォルトである次の状態を確認すること。

- Do CTF-correction: **Yes**
- Ignore CTFs until first peak?: **No**

Optimization タブ中は、次の状態になるよう変更すること。

- Number of classes: **1**
- Mask diameter (A): **200**
⇒タンパク質の大きさに応じて変更。
- Flatten and enforce no-negative solvent?: **Yes**
- Symmetry: **D7**

⇒C1 にすると全てをカバーできるが、うまく収束しないこともあるので、分子に対称性がある場合には試してみるのもよい。その際 Initial angular sampling は小さめに設定する。対称性を選択すると、対称性にあったマップの配向性を提示してくれるので、そのまま初期モデルとして使用できるため便利です。

あるいは対称性 C1 を選択計算後に、対称性 D7 にしたいときには、次のコマンドを使用します。このコマンドはどの対称性にでも対応します。

```
>relion_align_symmetry --i InitialModel/job016/run_it300_class001.mrc ¥  
--o InitialModel/job016/run_it300_class001_alignD7.mrc --sym D7↵
```

- Initial angular sampling: 7.5 degrees
- Offset search range (pix): 6
- Offset search step (pix): 2

SGD タブ中は、次の状態になるよう一部変更すること。(黄色地部分)

- Number of initial iterations: 50
- Number of in-between iterations: 200
- Number of final iterations: 50
- Write-out frequency (iter): 10
- Initial resolution (A): 35
- Final resolution (A): 10
- Initial mini-batch size: 100
- Final mini-batch size: 500
- SGD increased noise variance half-life: -1

Compute タブ中では、次のように変更します。

- Use parallel disc I/O: Yes
- Number of pooled particles: 30
- Skip padding?: No
- Pre-read all particles into RAM?: No
- Copy particles to scratch directory: /ssd

⇒実習で使用する計算機では、計算用スラッチディスクとして/ssd を使用しています。用意していない場合には、ここは空欄にすること。

- Combine iterations through disk?: No
- Use GPU acceleration?: Yes

⇒Yes:GPU を使った計算を行う場合。No:CPU のみの計算を行う場合

- Which GPU to use:

⇒空欄の場合、デフォルトとして使用できるGPU全部をアサインしてくれます。

Running タブ中で、次のように変更します。

- Number of MPI procs: 5
- Number of threads: 4
- Submit to queue?: Yes
- Queue name: openmpi
- Queue submit command: qsub
- Standard submission script: /usr/local/relion/bin/qsub.csh

・ Minimum dedicated cores per node: 1

・ Additional arguments:

⇒上記の変数は、今回の実習環境になっています。計算機環境に応じて変更してください。

最後にタブ右下の **Run!** ボタンを押して計算を開始します。

正しく計算が終了すると、top_directory/InitialModel/jobXXX (XXX は、それぞれ異なりますので、Finished job のところに出ている InitialModel の番号を参照)に、

収束した初期モデル： run_it300_class001.mrc が生成されます。

マップや構造は Chimera (※下記参照) で確認してみてください。図 4.2 のようにサイクル数が増加するとともに、3Dモデルが収束する様子が確認できます。うまくいかないときは、SGD タブ中の変数を変えて再度計算することも良いですが、大半は選んだ粒子画像にコンタミの氷やゴミが混ざっていることが原因です。再度、3.2 2Dクラス平均の選択をやり直して、きれいな2Dクラス由来の粒子だけになるようにしてみてください。

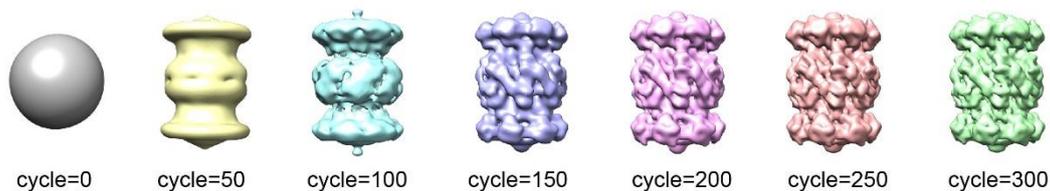


図 4.2：SGDによる初期モデルと収束状況

※Chimera

電子顕微鏡由来の立体構造・マップを可視化する際に便利なソフトウェアです。インストール・使用方法などは次の URL を参照してみてください。

<https://www.cgl.ucsf.edu/chimera/>

5. 3D クラス分類

5.1 3D クラス分類計算

左側の job-type ブラウザーの **3D classification** をクリックすると、図 5.1 の画面へ変化します。

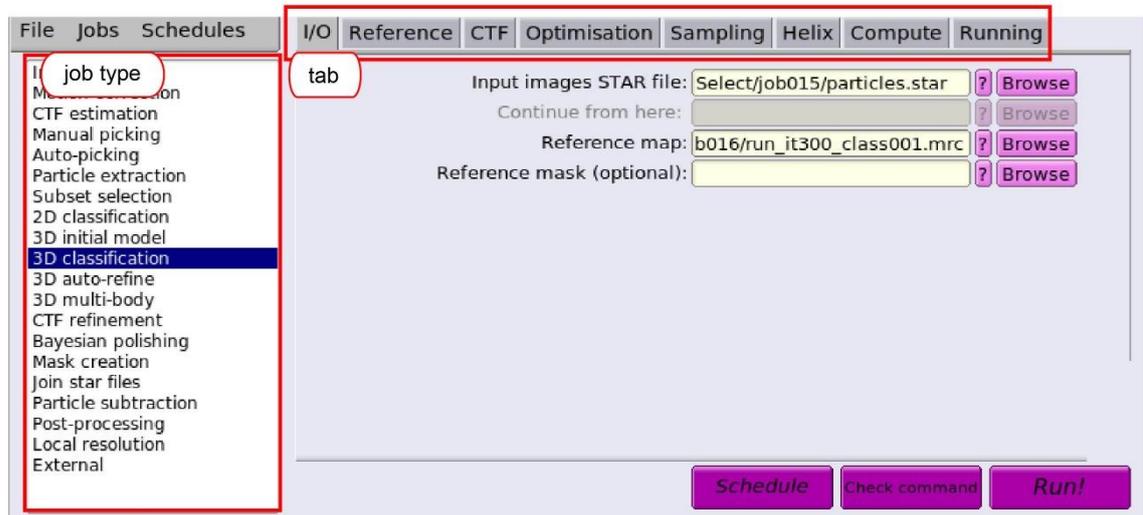


図 5.1 : 3D Classification

I/O タブ中を次のように変更します。

- Input images STAR file: **Select/job015/particles.star**

⇒3.2 で選択した画像リストファイル (**job+ 3 ケタ数字は適宜変更**)

- Reference map: **InitialModel/job016/run_it300_class001.mrc**

⇒4.1 で計算した初期 3D モデル (**job+ 3 ケタ数字は適宜変更**)。Symmetry が決まっている場合には、正しく適用したものを使用するほうがよい。今回の実習では InitialModel 計算時に対称性 D7 を使っているのので、結果をそのまま使用しています。対称性なしの C1 で計算した後、4.1 中の説明にある操作で対称性をコマンド `relion_align_symmetry` により適用した結果をお持ちの際はそちらをご利用ください。

Reference タブ中を次のように変更します。

- Ref. map is on absolute greyscale? **Yes**

⇒同じデータセットから計算した場合には、Yes。異なるデータセットや原子モデルから計算したものをを使用する場合には、再度 `relion` のなかで grey スケールを計算する必要があるため No。

- Initial low-pass filter (A): **60**

⇒モデルバイアスを減らすため 40-60 Å 程度の値を使用します。Ribosome のような巨大分子の場合は、70 Å。

- Symmetry: **C1**

⇒最初の 3D クラス分類では、`symmetry` を適用しないほうが良い場合 (壊れたりノイズ粒子由来のクラスを除ける) が多い。

CTF タブ中を次のように変更します。

- Do CTF-correction? Yes
- Has reference been CTF-corrected? Yes
⇒4.1 SGD で計算したモデルを使う場合、Yes。その他、No。
- Ignore CTFs until first peak? No
⇒3.1 2D Classification で使用したオプションと同じにすること。

Optimization タブ中は、次のように変更すること。

- Number of classes: 4
⇒数が増えるほど、より多くの3Dクラスモデルができます。それに比例して計算量も増えるため結果を得るまでの時間も増えていきます。
- Regulation parameter T: 2
⇒クライオ画像の場合、T=2~4 が適切です。数が大きいほど、詳細な構造情報での分類がなされます。
- Number of iterations: 25
⇒計算を繰り返すサイクル数。
- Use fast subsets: Yes
⇒粒子数が数万個程度のとき、計算の高速化が見込める。
- Mask diameter (A): 200
⇒分子の大きさに応じて。
- Mask individual particles with zeros?: Yes
- Limit resolution E-step to (A): -1
⇒正の数値をセットすると、その数値内までの周波数情報を使用して画像のアライメントを行います。通常は、-1 でよいのですが、オーバーフィッティングが見られる際に使用すること効果的です。

Sampling タブ中は変更しませんので、以下のデフォルトであることを確認。

- Perform image alignment? Yes
- Angular sampling interval: 7.5 degree
- Offset search range (pix) : 5
- Offset search step (pix) : 1
- Perform local angular searches? No

Hexlix タブ中は変更しませんので、以下のデフォルトであることを確認。

- Classify 2D helical segments? No

Compute タブ中では、次のように変更します。

- Use parallel disc I/O: Yes
- Number of pooled particles: 30
- Skip padding?: No
- Pre-read all particles into RAM?: No
- Copy particles to scratch directory: /ssd

⇒実習で使用する計算機では、計算用スラッシュディスクとして/ssd を使用しています。用意していない場合には、ここは空欄にすること。

・ Combine iterations through disk?: No

・ Use GPU acceleration?: Yes

⇒Yes:GPU を使った計算を行う場合。No:CPU のみの計算を行う場合

・ Which GPU to use:

⇒空欄の場合、デフォルトとして使用できるGPU全部をアサインしてくれます。

Running タブ中で、次のように変更します。

・ Number of MPI procs: 5

・ Number of threads: 4

・ Submit to queue?: Yes

・ Queue name: openmpi

・ Queue submit command: qsub

・ Standard submission script: /usr/local/relion/bin/qsub.csh

・ Minimum dedicated cores per node: 1

・ Additional arguments:

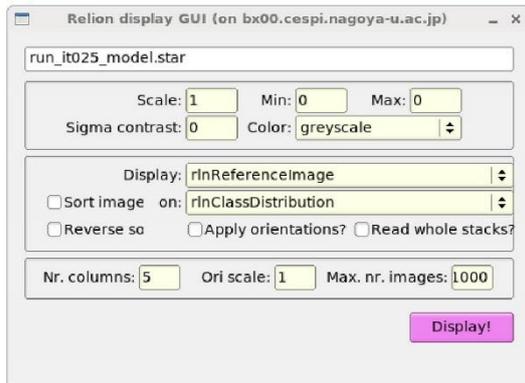
⇒上記の変数は、今回の実習環境になっています。計算機環境に応じて変更してください。

続いて、タブ右下の Run ボタンを押すと計算が始まります。

(A)



(B)



(C)



図 5.2 : 3D Classification --- Display

計算後、3Dクラス平均を確認するために、図 5.2(A)のようにタブ右下の Display: out:run_it025_model.star を選択すると、図 5.2(B)の画面がポップアップします。

・ Scale: 1

⇒使用するコンピューター画面に応じて変更。

図 5.2(B)への入力後、**Display** を押し、図 5.2(C)のようにクラス平均画像が、左から順に Class1,2,...となります (ソートをおこなっていないため)。

計算結果は、top_directory/Class3D/**job017** (黄色地の番号は、それぞれ異なります。Finished job のところに出ている Class3D の番号を参照)に、

収束した 4 種類の 3 Dモデル: run_it025_**class001**.mrc, ..., run_it025_**class004**.mrc が生成されます。各ファイル名の **class001** が計算上付けられたクラス番号です。

Chimera(p 3 1 参照)で確認してみてください。比較的きれいな 3 Dクラス番号を書き留めておいて次の操作に備えてください。

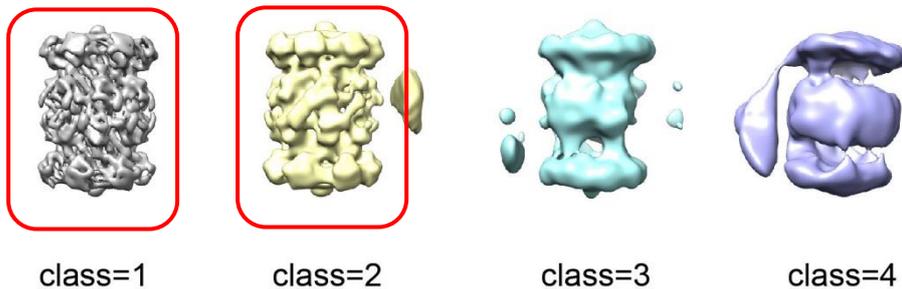


図 5.3 : 3D クラス分類の結果例

5.2 3 Dクラスの選択

計算した 3 Dクラスの中から適切な 3 Dクラスのみを選択します。左側の job-type の **Subset selection** をクリックすると、図 5.4 の画面へ変化します。

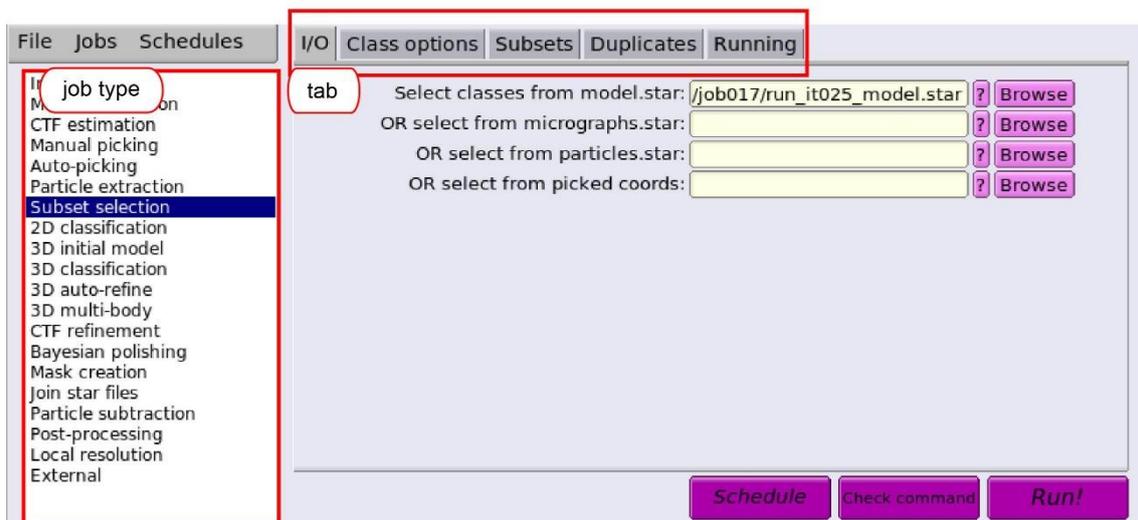


図 5.4 : Subset selection --- 3D Classification

I/O タブ中を次のように変更します。

- Input images STAR file: **Class3D/job017/run_it025_model.star**
⇒5.1 で計算した 3D クラスのモデルリストファイル (**job+ 3 ケタ数字は適宜変更**)

Class options タブ中で、次のように変更します。

- Re-center the class averages? No
- Regroup the particles? No

Subsets タブ中で、次のように変更します。

- Select based on metadata values?: No
- OR: select on image statistics?: No
- OR: split into subsets?: No

Duplicates タブ中で、次のように変更します。

- OR: remove duplicates?: No

⇒粒子リストファイル particles.star のみ使用できる。

Running タブを押して、タブ右下の Run! ボタンを押します。

(A)

(B)

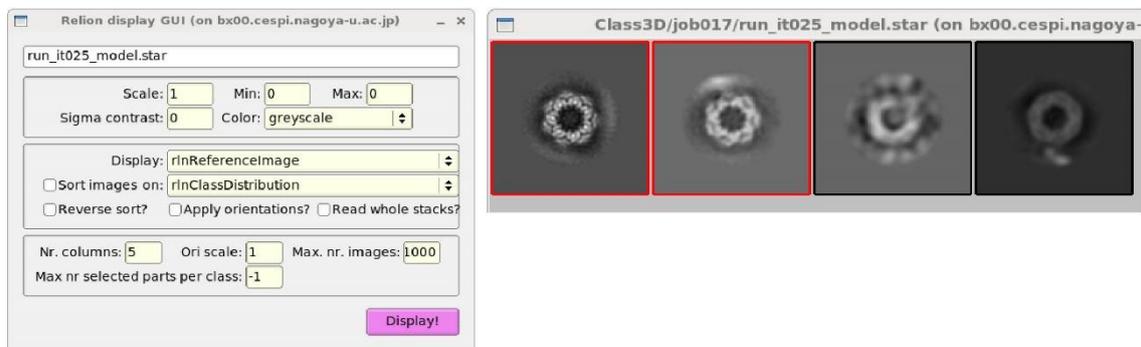


図 5.5 : Subset selection ---GUI

すると、図 5.5(A)の画面がポップアップします。

- Scale: 1

⇒使用するコンピューター画面に応じて変更。

図 5.5(A)への入力後、Display を押すと、図 5.5(B)のようにクラス平均画像が、クラスの番号順に上左側から順に表示されます。画面上で左クリックすると、マウスがある位置にあるクラスを選択することができ、クラス画像の周囲に赤い囲み線が表示されます。同じクラス画像を再度左クリックすると、選択を外すことができ、赤い囲み線が消えます。最終的にきれいな 3D クラスだけを選択し（決める際は、5.1 で述べたように Chimera で確認してください）、右クリックし、“Save selected classes”を選ぶと選択したクラスに含まれる画像全部がファイルリストして保存されます。

6. 3D 精密化

前章の3Dクラス分類によって単一なコンフォメーションを選び出せた後は（言い換えるとマイナーなコンフォメーションやノイズ由来の画像を除けた場合）、Relionの3D auto-refine機能によって自動的に高分解能のマップを計算することができます。その際、gold-standard Fourier Shell Correlation (FSC)という分解能定義を利用します。この定義は、解析のために選び出した粒子をランダムに2つのグループにあらかじめ分け、それぞれのグループで独立に立体構造を解析します。得られた独立な立体構造をフーリエ変換し、計算点とする任意の分解能ごとに2つのグループ間の相関を計算します。Relionでは、相関値であるFSCが、0.143に初めて到達した分解能位置を解析結果の分解能として定義しています。このように独立に立体構造を決定することで変数のオーバーフィッティングを避け、分解能の過大評価につながらないように考慮されています。

6.0 画像のスケール変更

2.の粒子ピックアップ時にダウンスケールしている場合に、3D auto-refine前にスケールを変更する際は、再度粒子の抽出を行います。但し、今回はダウンスケールのまま計算しますので、この操作はスキップします。2.2あるいは2.6の粒子の抽出を参考にしてください。

6.1 3D 自動精密化計算

左側のjob-typeブラウザーの3D auto-refineをクリックすると、図6.1の画面へ変化します。

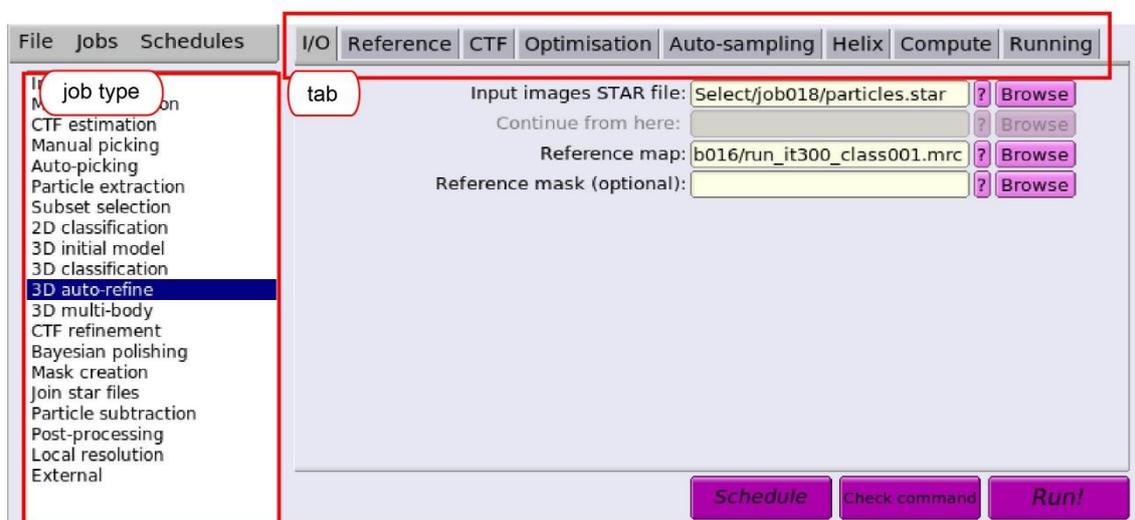


図 6.1 : 3D auto-refine

I/O タブ中を次のように変更します。

・ Input images STAR file: `Select/job018/particles.star`

⇒5.2 で選択した 3D クラス由来の粒子画像リストファイル (**job+3ケタ数字は適宜変更**)

・ Reference map: `InitialModel/job016/run_it300_class001.mrc`

⇒4 で生成した初期 3D モデルあるいは、5 で出力された 3D クラス。

• Reference mask (optional):

⇒通常は空欄のまま。

Reference タブ中で、次のように変更します。

• Ref. map is on absolute greyscale? Yes

⇒同じデータセットから計算した場合には、Yes。異なるデータセットや原子モデルから計算したものをを使用する場合には、再度 relion のなかで grey スケールを計算する必要があるため No。

• Initial low-pass filter (A): 60

⇒モデルバイアスを減らすため 40-60 Å 程度の値を使用します。Ribosome のような巨大分子の場合は、70 Å。

• Symmetry: D7

⇒分子の対称性を入力。

CTF タブ中を次のように変更します。

• Do CTF-correction? Yes

• Has reference been CTF-corrected? Yes

⇒4.1 SGD で計算したモデルや 3D class のマップを使う場合、Yes。その他、No。

• Ignore CTFs until first peak? No

⇒3.1 2D Classification で使用したオプションと同じにすること。

Optimization タブ中は、次のように変更します。

• Mask diameter (A): 200

⇒分子の大きさに応じて。

• Mask individual particles with zeros? Yes

• Use solvent-flattened FSCs? No

⇒通常は、No でよい。

Auto-sampling タブ中は変更しませんので、以下のデフォルトであることを確認。

• Initial angular sampling: 7.5 degree

• Initial offset range (pix) : 5

• Initial offset step (pix) : 1

• Local searches from auto-sampling: 1.8 degrees

Hexlix タブ中は変更しませんので、以下のデフォルトであることを確認。

• Do helical reconstruction? No

Compute タブ中では、次のように変更します。

• Use parallel disc I/O: Yes

• Number of pooled particles: 30

• Skip padding?: No

• Pre-read all particles into RAM?: No

- Copy particles to scratch directory: /ssd
⇒実習で使用する計算機では、計算用スラッチディスクとして/ssd を使用しています。用意していない場合には、ここは空欄にすること。
- Combine iterations through disk?: No
- Use GPU acceleration?: Yes
⇒Yes:GPU を使った計算を行う場合。No:CPU のみの計算を行う場合
- Which GPU to use:
⇒空欄の場合、デフォルトとして使用できるGPU全部をアサインしてくれます。

Running タブ中で、次のように変更します。

- Number of MPI procs: 5
- Number of threads: 4
- Submit to queue?: Yes
- Queue name: openmpi
- Queue submit command: qsub
- Standard submission script: /usr/local/relion/bin/qsub.csh
- Minimum dedicated cores per node: 1
- Additional arguments:

⇒上記の変数は、今回の実習環境になっています。計算機環境に応じて変更してください。

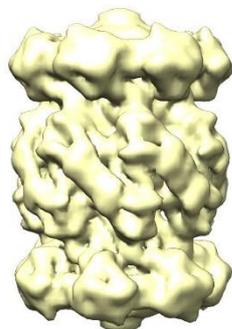
続いて、タブ右下の Run ボタンを押すと計算が始まります。

```

(A) message
.....~(,">
Auto-refine: Refinement has converged, stopping now...
Auto-refine: + Final reconstruction from all particles is saved as: Refine3D/job019/run_class001.mrc
Auto-refine: + Final model parameters are stored in: Refine3D/job019/run_model.star
Auto-refine: + Final data parameters are stored in: Refine3D/job019/run_data.star
Auto-refine: + Final resolution (without masking) is: 5.36381
Auto-refine: + Please remember to replace the unfiled mrc maps and calculate a higher resolution FSC
stderr will go here: double-click this window to open stderr in a separate window

```

(B) Initial Model



(C) Converged Model

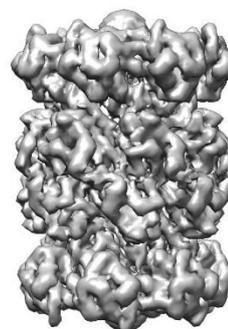


図 6.2 : 3D auto-refine --display

正しく計算が終了すると、message ウィンドウ中(図 6.2(A))に分解能として Final resolution (without masking) is : と表示されます。この分解能は、マスクをかける前の溶

媒領域を含んだ状態であるため、通常低く見積られることとなりますので、次の7章での操作によって適切な分解能評価を行う必要があります。

top_directory/Refine3D/job019 (job+3ケタ数字は、それぞれ異なります。Finished job のところに出ている番号を参照)に、

収束した 3D モデル(図 6.2(C)) : run_class001.mrc

angular distribution : run_class001_angdist.bild

が生成されます。Chimera(p 3 1 参照)で初期モデル(図 6.2(B))と収束したモデル(図 6.2(C))を比較確認してみてください。また、収束したモデルに関しては contour level をぎりぎりまで落としてノイズがでないようなデンシティ値を書き留めておいてください。画像のサンプリングサイズが変わると大きく変わることがありますので、毎回確認してください。この時のデンシティ値は、次章の 7.1 mask 作成時に使用します。

6.0 画像のスケール変更に戻って、マップのスケールを変更した状態で粒子の抽出を行います。その後、**6.1 3D 自動精密化計算**を行います。その際、スケール変更に伴い参照すべきマップのスケール変更も行います。その際、次のようなコマンド入力になります。例、現在のマップは 1 pixel = 2.64 Å で、スケール変更後は粒子サイズ(Rescaled size = 280) 注:p14)、1 pixel = 1.2068 Å として計算する場合。入力デンシティマップ;

Refine3D/job019/run_class001.mrc 出力デンシティマップ:

run_class001_job019_280pix.mrc とする。

```
>relion_image_handler -i Refine3D/job019/run_class001.mrc --angpix 2.64 ¥  
--rescale_angpix 1.2068 --new_box 280 --o run_class001_job019_280pix.mrc↵
```

画像のスケール変更は **2.2** あるいは **2.6** の**粒子の抽出**を参考にします。粒子画像の抽出として、左側の job-type ブラウザーの **Particle extraction** をクリックすると、図 2.3 の画面が表示されます。その中で注意すべき点は、次の通りです。

extract タブ中を押して、次のように変更します。

・ Particle box size (pix): **512**

・ Rescale particles?: **Yes**

・ Rescaled size (pixels): **280**

⇒ Bayesian Polishing の中で、必ず元画像での粒子サイズ(Particle box size = 512)、スケール変更後の粒子サイズ(Rescaled size = 280)を入力値として使用します。これらの数値は、ともに偶数の必要があります。入力時には気を付けてください。

⇒ 前もって Rescaled size の適切な値を決めるために、1 pixel の試料上での距離を計算しておいたほうがよいです。もちろん **Particle extraction** 操作後に表示されます。元画像において 1 pixel = 0.66 Å のため、スケール変更後 1 pixel = $512/280 * 0.66 \text{ \AA} = 1.12068 \text{ \AA}$ に相当します。

7. マスク作成とポスト処理

前章の 3D 自動精密化後のマップ (図 6.2(C)) を見て、3.4 Å 分解能のマップであると言われても、それほど細かい特徴があるようには見えません。クライオ電子顕微鏡の画像は、さまざまな理由から構造因子の振幅が弱められています。そのため 3D 自動精密化の後には、マップの B-factor シャープニングを行うことではじめて原子分解能の特徴である側鎖の様子が見えてきます。更に、Refine3D で計算された gold-standard FSC は、溶媒領域を考慮したものではありませんでしたので、本来の分解能よりも低く見積もられています。そこでポスト処理と呼ばれる B-factor シャープニングとマスクを考慮した FSC 計算を行う必要があります。

7.1 マスク作成

ポスト処理を行う前に蛋白質と溶媒領域を適切に分けるためのマスクを作成します。左側の job-type ブラウザーの **Mask creation** をクリックすると、図 7.1 の画面へ変化します。



図 7.1 : Mask creation

I/O タブ中を次のように変更します。

- Input 3D map: **Refine3D/job039/run_class001.mrc**
⇒ 6.1 で計算した収束後のマップファイル (**job+3ケタ数字は適宜変更**)

Mask タブ中で、次のように変更します。

- Lowpass filter map (A) **20**
⇒ 溶媒領域のためのマスクとして 15-20 Å が最適。
- Pixel size (A) **-1**
⇒ 計算時の画像の画素サイズ。-1: 指定した Input 3D map から値を読み込み使用。
- Initial binarisation threshold: **0.0045**
⇒ 6.2 のように chimera を使って確認したノイズがぎりぎりでないような threshold。
- Extend binary map this many pixels: **0**

⇒lowpass フィルターで決めたタンパク質領域のマスクを、上記ピクセル分だけ全方向に拡張します。

- ・ Add a soft-edge of this many pixels: 6

⇒lowpass フィルターで決めたタンパク質領域に対して全方向に拡張したマスクに対して、cosine-shaped のソフトエッジマスクを適用します。

Helix タブ中は、確認のみで変更する必要はありません。

- ・ Mask a 3D helix?: No

Running タブ中は、確認のみで変更する必要はありません。

- ・ Number of threads: 1
- ・ Submit to queue? No

最後にタブ右下の **Run!** ボタンを押すと計算が始まります。

7.2 ポスト処理

左側の job-type ブラウザーの **Post-processing** をクリックすると、図 7.2 の画面へ変化します。

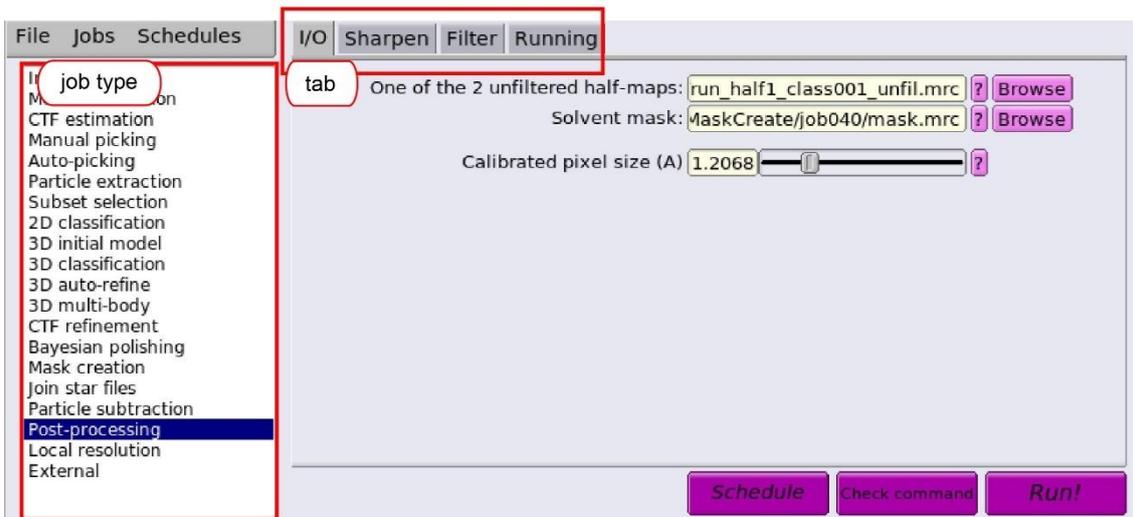


図 7.2 : Post-processing

I/O タブ中を次のように変更します。

- ・ One of the 2 unfiltered half-maps: Refine3D/job039/run_half1_class001_unfil.mrc

⇒6.1 で計算した収束際の独立データセットで決定した2つの unfiltered のマップファイルのうち1つ (**job+3ケタ数字は適宜変更**)

- ・ Solvent mask: MaskCreate/job040/mask.mrc

⇒7.1 で計算した溶媒マスク領域 (**job+3ケタ数字は適宜変更**)

- ・ Calibrated pixel size: 1.2068

⇒unfiltered half-map の画素サイズ。しばしば、モデル構築時に数%スケールが異なることに気付くことがあります。その際は、この数値を変更することで、正しい分解能の評価を行うことができます。Mask 計算時はそのままオリジナルを使用して問題ありませ

ん。

Sharpen タブ中で、次のように変更します。

- Estimate B-factor automatically? Yes

⇒10Å分解能を超えているときは、自動評価で問題ありません。それ未満の分解能あるいは、自動評価でうまく決定できないことが分かったときには、No を選びマニュアルで ad-hoc B-factor を決めていきます。

- Lowest resolution for auto-B fit (A): 10

⇒通常変更する必要がありません。ad-hoc B-factor の際には、入力できなくなります。

- Use your own B-factor? No

- MTF of the detector (STAR file)

⇒使用している検出装置ごとに MTF(Modulated Transfer Function)は異なります。

MTF は、電子顕微鏡の加速電圧にも依存します。ただし、Relion-3.1 を使用した場合 **1.1 画像のインポート** で既に MTF ファイル名を指定していますので、わざわざ入力する必要はありません。

- Original detector pixel size:

⇒元データの 1 pixel あたりのサイズ。今回は 0.66Åとなるが、同上同様 **1.1 画像のインポート** にて入力済みのため不要。

Filter タブ中は、確認のみで変更する必要はありません

- Skip FSC-weighting? No

⇒通常は、No です。Yes にすると、グローバル分解能を超えた高分解能情報をもつ領域の解析を行う際に便利です。

Running タブ中は、確認のみで変更する必要はありません。

- Submit to queue? No

最後にタブ右下の **Run!** ボタンを押すと計算が始まります。

正しく計算が終了すると、message ウィンドウ中(図 7.3(A))に分解能として FINAL RESOLUTION: と表示されます。6 章で計算した Refine3D の分解能とは、異なり大きく改善されていると思います (このテキスト例では、2.99⇒2.64Åへ向上)。計算時に使用したマスクがタイトすぎると、+fraction f (solvent mask based): inf になります。その他、phase-randomized FSC (図 7.3(B)中の赤線) が FINAL RESOLUTION の分解能付近で 0.1 より大きくなります。こういった場合には、FINAL RESOLUTION には意味がありません。マスクで使用する・Initial binarisation threshold:の値を下げて、**7.1,7.2** の操作を行ってください。

全ての計算結果は、

top_directory/PostProcess/job041 (job+ 3 ケタ数字は、それぞれ異なります。Finished job のところに出ている番号を参照)に、

FSC plot : postprocess_fsc.eps

Sharpening 後の収束した map(図 7.3(C)) : postprocess.mrc
 が生成されます。分解能 FSC のプロットは、図 7.3(A)の **Display: logfile.pdf** を選択すると、図 7.3(B)の画面がポップアップします。マップに関しては、Chimera で確認してみてください。

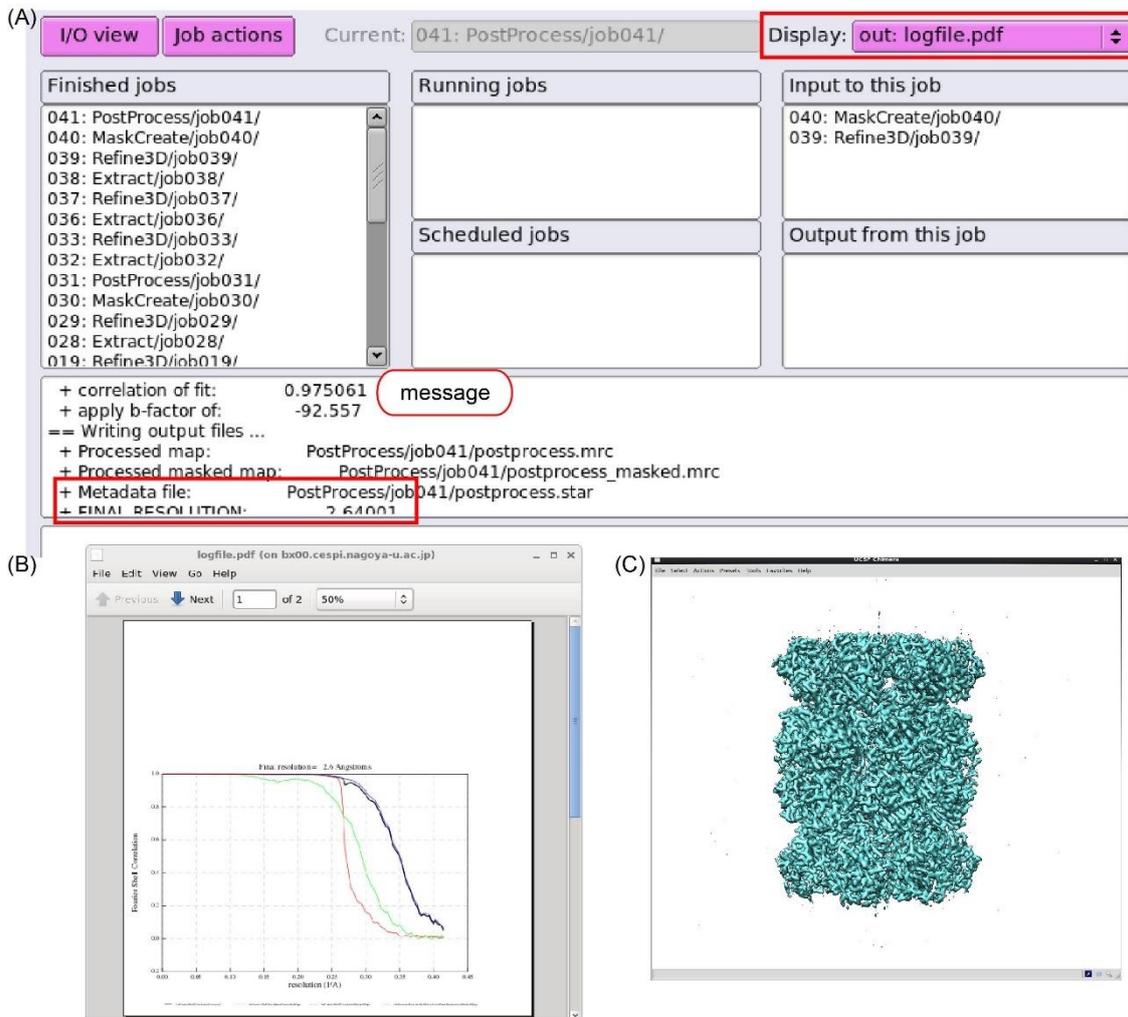


図 7.3 : Post-processing --- Results

(A)メッセージウィンドウ、(B)FSC プロット。緑：マスクのない状態での FSC。赤：phase-randomized FSC。振幅情報は観測値、位相情報のみ一定の分解能以上を乱数で置きかえているため、FSC=0.143 に対応する分解能（今回の結果は 2.64 Å 分解能）付近の値は 0 に近いことが望ましい。0.1 より大きいときは、マスクサイズに問題あり。青：マスク後の FSC。黒：phase-randomized FSC 補正後の FSC。分解能を評価する際に使う FSC 曲線。(C)chimera(p 3 1 参照)で表示したポスト処理後のマップ。

8. CTF と収差変数の精密化

ここでは、CTF、収差を表現する変数の評価を行います。変数としては、粒子ごとのデフォーカス値・非点、ビーム傾斜に伴う光軸からのズレ、光軸に対して非対称な倍率などの評価を行います。データの分解能に依存して、精密化が有効な変数は変わりますので、気を付けて精密化を行ってください。必要となる分解能に関しては、各変数の説明中で行いますが、詳細な説明は Relion 開発者らによる論文（文献3）を参照してください。

8.1 ビーム傾斜の精密化計算

左側の job-type ブラウザーの **CTF refinement** をクリックすると、図 8.1 の画面へ変化します。

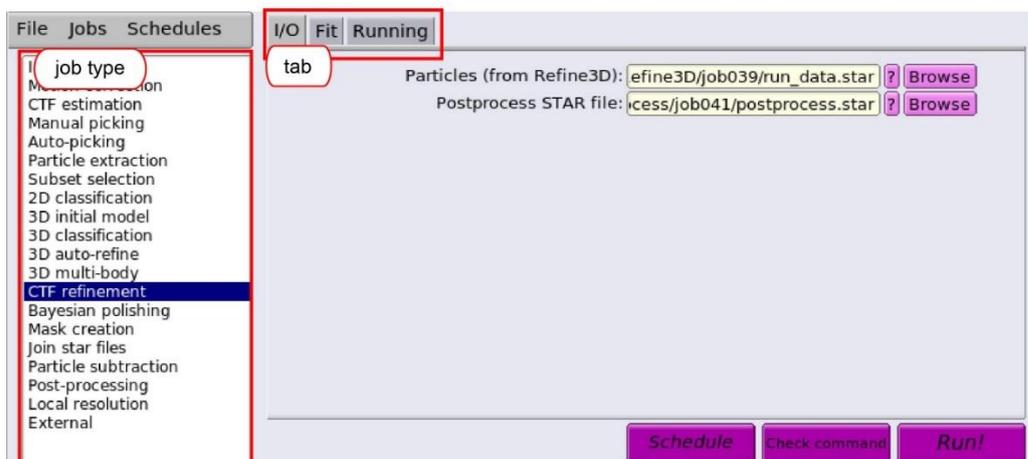


図 8.1 : CTF refinement

I/O タブ中を次のように変更します。

・ Particles (from Refine3D): **Refine3D/job039/run_data.star**

⇒6.1 の Refine3D で計算したパラメータを含む粒子リストファイル (**job+3ケタ数字は適宜変更**)

・ Postprocess STAR file: **PostProcess/job041/postprocess.star**

⇒7.2 の Postprocess で計算したパラメータを含むデータファイル (**job+3ケタ数字は適宜変更**)

Fit タブ中は、次のように変更します。

・ Estimate (anisotropic) magnification? **No**

⇒光軸に対して非対称な倍率に対する精密化は行わないので、No。

・ Perform CTF parameter fitting? **No**

⇒CTF 由来のデフォーカス値の精密化を行わないので、No。。

・ Estimate beamtilt? **Yes**

⇒ビーム傾斜の精密化するので、Yes。コマ収差フリーのアライメントを行ってデータ収集を行っていてもこの精密化は有用。データが<4.5 Å分解能のとき有効。

・ Also estimate trefoil? **Yes**

⇒三つ葉 (trefoil) 成分の収差精密化を行うので、Yes。データが $<3.5 \text{ \AA}$ 分解能で有効。

• Estimate 4th order aberrations? Yes

⇒4次までの高次収差の精密化を行うので、Yes。データが $<3.0 \text{ \AA}$ 分解能のとき有効。

3.0 \AA 分解能程度では大きな改善は期待できない。

• Minimum resolution for fits (A): 30

⇒通常は、そのまま。

Running タブ中は、次のように変更します。

• Number of MPI procs: 4

• Number of threads: 8

• Submit to queue?: Yes

• Queue name: openmpi

• Queue submit command: qsub

• Standard submission script: /usr/local/relion/bin/qsub.csh

• Minimum dedicated cores per node: 1

• Additional arguments:

⇒上記の変数は、今回の実習環境になっています。計算機環境に応じて変更してください。

最後にタブ右下の Run! ボタンを押すと計算が始まります。計算後は、図 8.2(A)の Display: logfile.pdf を選択することで、図 8.2(B,C)のグラフ (説明: 文献 3) を見るができます。

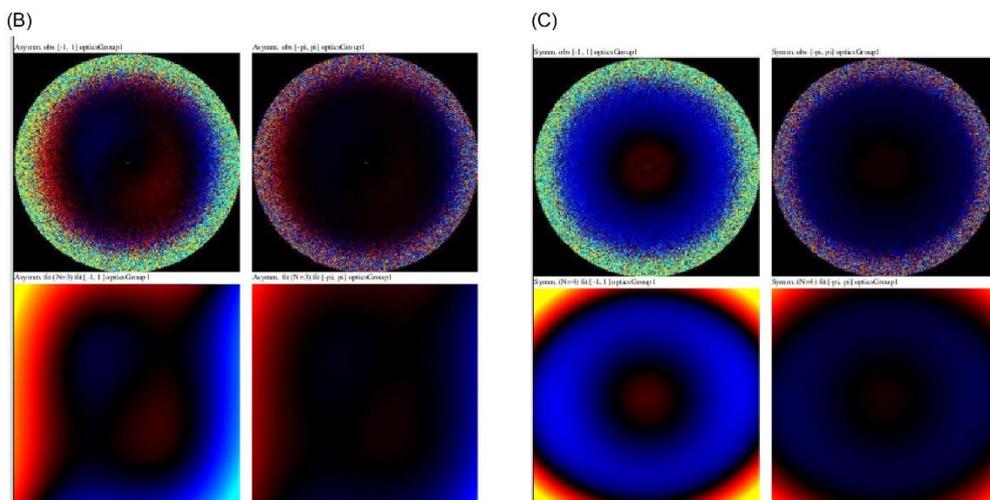


図 8.2 : CTF refinement --- Results after beam tilt refinement

全ての計算結果は、

top_directory/CtfRefine/job042 (job+3ケタ数字は、それぞれ異なります。Finished job で表示されている番号を参照してください。)

8.2 光軸に対して非対称な倍率の精密化計算

左側の job-type ブラウザーの CTF refinement をクリックすると、図 8.1 の画面へ変化します。

I/O タブ中を次のように変更します。

・ Particles (from Refine3D): CtfRefine/job042/particles_ctf_refine.star

⇒8.1 の CtfRefine で計算したパラメータを含む粒子リストファイル (job+3ケタ数字は適宜変更)

・ Postprocess STAR file: PostProcess/job041/postprocess.star

⇒7.2 の Postprocess で計算したパラメータを含むデータファイル (job+3ケタ数字は適宜変更)

Fit タブ中は、次のように変更します。

・ Estimate (anisotropic) magnification? Yes

⇒光軸に対して非対称な倍率に対する精密化のみ行なうので、No。その他のパラメータの精密化ができなくなるようになる。

・ Minimum resolution for fits (A): 30

⇒通常は、そのまま。

Running タブ中は、次のように変更します。

・ Number of MPI procs: 4

・ Number of threads: 8

・ Submit to queue?: Yes

・ Queue name: openmpi

・ Queue submit command: qsub

・ Standard submission script: /usr/local/relion/bin/qsub.csh

・ Minimum dedicated cores per node: 1

・ Additional arguments:

⇒上記の変数は、今回の実習環境になっています。計算機環境に応じて変更してください。

最後にタブ右下の Run! ボタンを押すと計算が始まります。計算後は、図 8.3(A)の Display: logfile.pdf を選択することで、図 8.3(B)のグラフ (説明: 文献3) を見るができます。

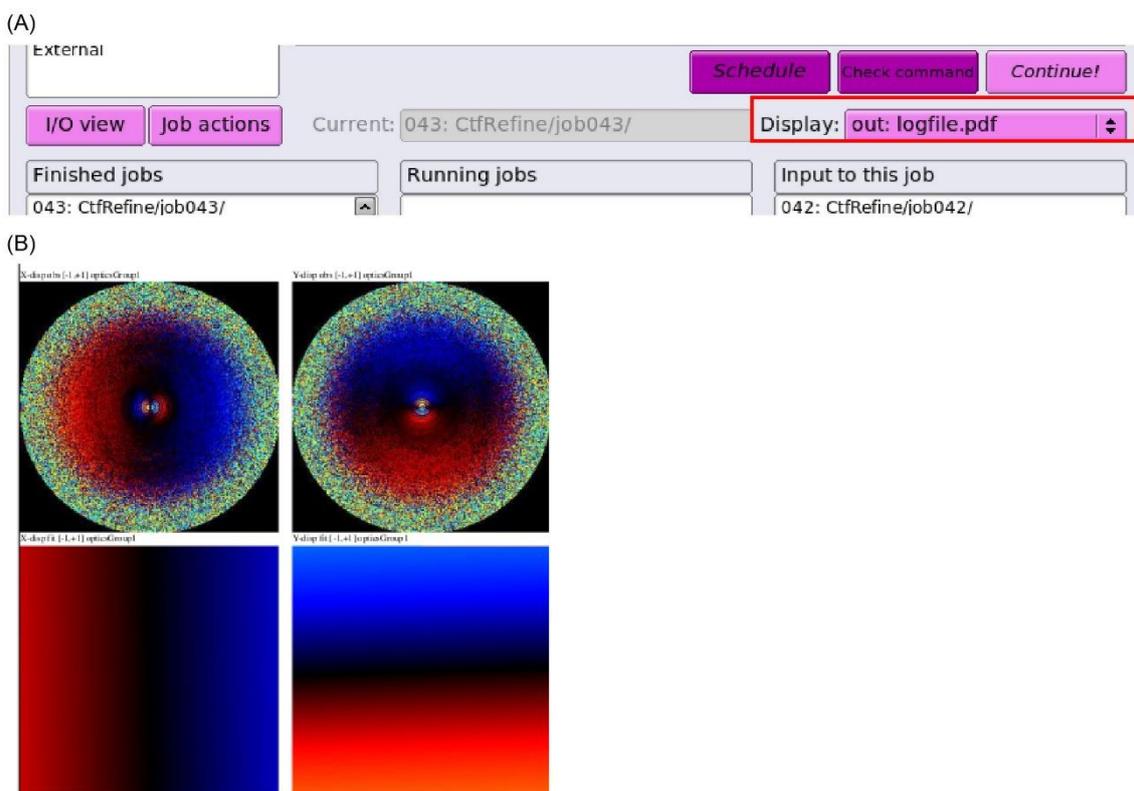


図 8.3 : CTF refinement --- Results after anisotropic magnification refinement

全ての計算結果は、

top_directory/CtfRefine/job043 (job+ 3 ケタ数字は、それぞれ異なります。Finished job に表示されている番号を参照してください。)

8.3 CTF に関する変数の精密化計算

左側の job-type ブラウザーの CTF refinement をクリックすると、図 8.1 の画面へ変化します。

I/O タブ中を次のように変更します。

・ Particles (from Refine3D): CtfRefine/job043/particles_ctf_refine.star

⇒8.2 の CtfRefine で計算したパラメータを含む粒子リストファイル (job+ 3 ケタ数字は適宜変更)

・ Postprocess STAR file: PostProcess/job041/postprocess.star

⇒7.2 の Postprocess で計算したパラメータを含むデータファイル (job+ 3 ケタ数字は適宜変更)

Fit タブ中は、次のように変更します。

・ Estimate (anisotropic) magnification? No

⇒光軸に対して非対称な倍率に対する精密化を行わないので、No。

・ Perform CTF parameter fitting? Yes

⇒CTF 由来のデフォーカス値の精密化を行わないので、No。

• Fit defocus? **Per-particle**

⇒Per-particle:粒子ごとのデフォーカス値を精密化する場合。Per-micrograph:電子顕微鏡画像ごとにデフォーカス値を精密化する場合。

• Fit astigmatism? **Per-micrograph**

⇒Per-micrograph:電子顕微鏡画像ごとにデフォーカス値を精密化する場合。Per-particle:粒子ごとのデフォーカス値を精密化する場合。

• Fit B-factor? **No**

⇒通常は、そのまま。

• Fit phase-shift? **No**

⇒通常は、そのまま。Phase plate を使用したデータの場合に使用することもあるが、結果は慎重に検討すべき。

• Estimate beamtilt? **No**

⇒ビーム傾斜の精密化は行わないので、No。

• Also estimate trefoil? **No**

⇒Estimate beamtilt:No を選ぶと自動的に No になり、変更できない。

• Estimate 4th order aberrations? **No**

⇒4次収差の精密化を行わないので、No。

• Minimum resolution for fits (A): **30**

⇒通常は、そのまま。

Running タブ中は、次のように変更します。

• Number of MPI procs: **4**

• Number of threads: **8**

• Submit to queue?: **Yes**

• Queue name: **openmpi**

• Queue submit command: **qsub**

• Standard submission script: **/usr/local/relion/bin/qsub.csh**

• Minimum dedicated cores per node: **1**

• Additional arguments:

⇒上記の変数は、今回の実習環境になっています。計算機環境に応じて変更してください。

最後にタブ右下の **Run!** ボタンを押すと計算が始まります。計算後は、図 8.4(A)の **Display: logfile.pdf** を選択することで、図 8.4(B)のグラフを見ることができます。

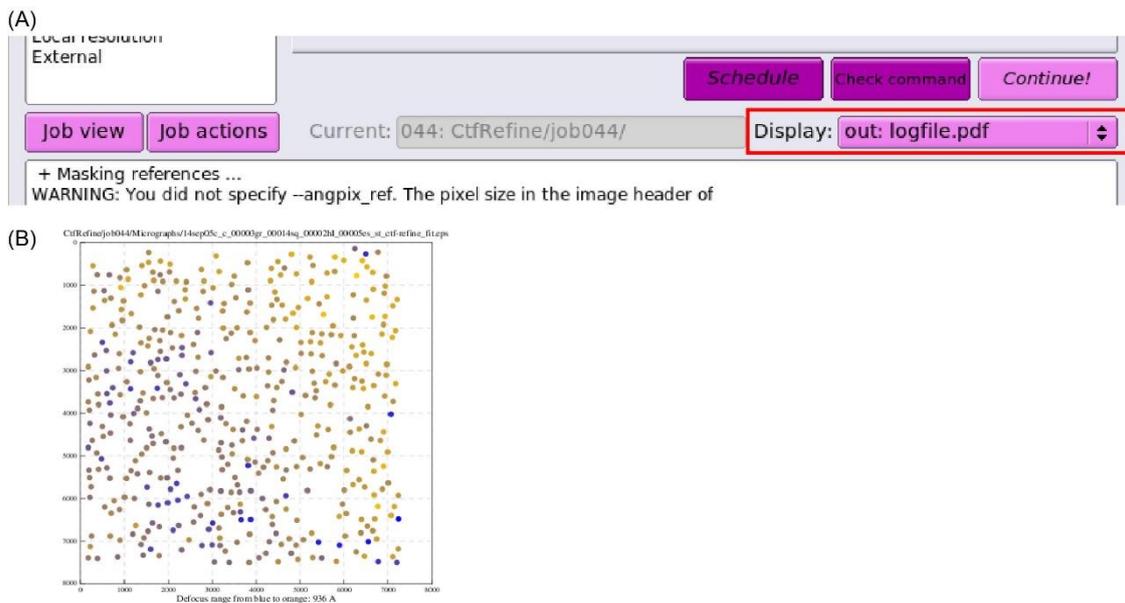


図 8.4 : CTF refinement --- Results after defocus parameters refinement

全ての計算結果は、

top_directory/CtfRefine/job044 (job+3ケタ数字は、それぞれ異なります。Finished job に表示されている番号を参照してください。)

8.1~8.3 まで順に操作した最後の粒子画像情報ファイル： particles_ctf_refine.star を使って、6章の3D精密化、7章マスク生成とポスト処理を順に行ってください。正しく計算が終了すると、Postprocess の message ウィンドウ中(図 8.2(A))に分解能として FINAL RESOLUTION: と表示されます。6や7章で計算した分解能とは、異なり大きく改善されていると思います (このテキスト例では、2.7⇒2.3Å分解能へ)。また、準備したマスクがタイトすぎると、phase-randomized マップ由来で計算した FSC 値 (図 8.2(B)中の赤線) が FINAL RESOLUTION のところで 0 近くになることも確認してください。もし 0.1 より大きい場合には、message ウィンドウの下にあるエラーウィンドウ内 (Appendix 図 A.1) に赤字で警告がでます。その場合には、マスクがタイトすぎますので、7.1 の Mask タブ中 Initial binarization threshold の値を下げてください。

9. Bayesian Polishing による粒子画質改善

各粒子画像の電子線照射に伴うモーション補正に関して、ベイズ推定に基づいた計算を行います。推定で使用する変数は、わずか3つ σ for velocity σV ($\text{\AA}/\text{dose}$), σ for divergence σD (\AA), σ for acceleration σA ($\text{\AA}/\text{dose}$)だけです。まずはトレーニングモードにて、4000~10000 個程度の粒子画像を使って変数を決定します。続いてポリッシングモードにて、決定した3変数を利用して粒子画像ごとのモーション補正を行います。また、トレーニングモードの推定 σ 値は再現性がとれないことも多々あります。再現性がそれほど高くないうえ、計算時間もかかりますので、デフォルト σ 値 ($\sigma V = 0.2$; $\sigma D = 5000$; $\sigma A = 2$) を使って直接ポリッシングモードを実行してみるのも1つのやり方です。

9.1 トレーニングモードでの計算

左側の job-type ブラウザーの **Bayesian polishing** をクリックすると、図 9.1 の画面へ変化します。

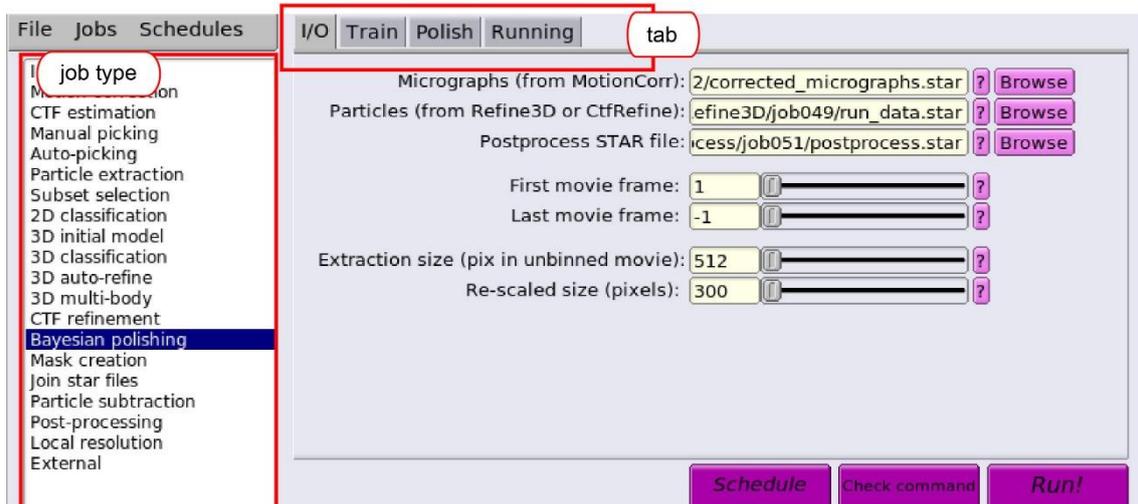


図 9.1 : Bayesian polishing

I/O タブ中を次のように変更します。

- Micrographs (from MotionCorr): **MotionCorr/job002/corrected_micrographs.star**
⇒1.2 モーション補正で計算したパラメータを含むデータファイル (**job+3ケタ数字は適宜変更**)
- Particles (from Refine3D): **Refine3D/job049/run_data.star**
⇒8.4 の Refine3D で計算したパラメータを含む粒子リストファイル (**job+3ケタ数字は適宜変更**)
- Postprocess STAR file: **PostProcess/job051/postprocess.star**
⇒8.4 の Postprocess で計算したパラメータを含むデータファイル (**job+3ケタ数字は適宜変更**)
- First movie frame: **1**
⇒補正で使用するムービー画像の開始位置を指定する。通常 1 : 最初に設定しますが、

何らかのトラブルでビームブランキングの影響が大きい場合には、2 などに変更することも。

・ Last frame for corrected sum: -1

⇒使用するムービー画像の最後位置を指定する。-1 なら最後まで。

・ Extraction size (pix in unbinned movie): 512

⇒オリジナルムービー画像上でのピクセルサイズ。直前で使用した Particle extraction で使用した extract タブ中の Particle box size (pix)を参照してください。要偶数で指定。

・ Rescaled size (pixels): 300

⇒Postprocess の際に使用した画像のピクセルサイズ。直前で使用した Particle extraction で使用した extract タブ中の Rescaled size (pixels)を参照してください。extract タブ中で Rescale particles? No を選んだ場合には、extract タブ中の Particle box size (pix)の数値を入力してください（上記 Extraction size と一致）。要偶数で指定。

Train タブ中は、次のように変更します。

・ Train optimal parameters? Yes

⇒トレーニングモードのため、Yes。

・ Fraction of Fourier pixels for testing: 0.5

⇒デフォルト値(=0.5)を推奨。

・ Use this many particles: 8000

⇒変数の推定に使用する粒子画像数。画像数が多いほど精度は上がりますが、メモリー消費量と計算時間は比例します。そのためメモリー不足で計算が正常に終了しないときは、粒子数を減らしてみてください。4000 個より少ない粒子数での計算はお勧めできません。その他、収束せずにエラーで終了するようなときは、粒子画像数を増やして計算し直してみるとうまくいくことが多いようです。

Polish タブ中は、次のように変更します。

・ Perform particle polishing? No

⇒トレーニングモードのため、No。その他の変数は設定できなくなります。

Running タブ中は、次のように変更します。

・ Number of MPI procs: 1

・ Number of threads: 28

・ Submit to queue?: Yes

・ Queue name: openmpi

・ Queue submit command: qsub

・ Standard submission script: /usr/local/relion/bin/qsub.csh

・ Minimum dedicated cores per node: 1

・ Additional arguments:

⇒上記の変数は、今回の実習環境になっています。計算機環境に応じて変更してください。

最後にタブ右下の **Run!** ボタンを押すと計算が始まります。この計算は、MPI 化されていませんので、MPI 数は 1 にする必要があります。Thread の数は core 数分増やすと計算が速く終わります。計算結果は、メッセージ画面に

Good parameters --s_vel 0.456 --s_div 2385 --s_acc 30.405

と表示され、 σ 値($\sigma V = 0.456$; $\sigma D = 2385$; $\sigma A = 30.405$)に相当します。

9.2 ポリッシングモードでの計算

左側の job-type ブラウザーの **Bayesian polishing** をクリックすると、図 9.1 の画面へ変化します。

I/O タブ中は、9.1 で指定した設定値のまま大丈夫です。

Train タブ中は、次のように変更します。

・ Train optimal parameters? **No**

⇒ポリッシングモードのため、No。その他の変数は変更できなくなります。

Polish タブ中は、次のように変更します。

・ Perform particle polishing? **Yes**

⇒ポリッシングモードのため、Yes。その他の変数は設定できなくなります。

・ Optimised parameter file: **Polish/job052/opt_params_all_groups.txt**

⇒**9.1 トレーニングモード**で計算したパラメータを含むデータファイル (**job+3ケタ数字は適宜変更**)。

・ OR use your own parameters? **No**

⇒今回は **9.1 トレーニングモード**で計算した結果を使用するため、No。その他の変数 (Sigma for velocity, Sigma for divergence, Sigma for acceleration)は設定できなくなります。

・ Minimum resolution for B-factor fit (A): **20**

・ Maximum resolution for B-factor fit (A): **-1**

⇒上記2つの変数はデフォルト値を使用してください。

Running タブ中は、次のように変更します。

・ Number of MPI procs: **4**

・ Number of threads: **28**

・ Submit to queue?: **Yes**

・ Queue name: **openmpi**

・ Queue submit command: **qsub**

・ Standard submission script: **/usr/local/relion/bin/qsub.csh**

・ Minimum dedicated cores per node: **1**

・ Additional arguments:

⇒上記の変数は、今回の実習環境になっています。計算機環境に応じて変更してください

い。

最後にタブ右下の **Run!** ボタンを押すと計算が始まります。この計算は、MPI 化されていますが、かなりメモリーを消費しますので1計算ノードあたりの MPI 数は気を付けたほうがよいです。Thread の数は core 数分増やすと計算が速く終わります。図 9.2(A)の

Display: logfile.pdf を選択することで、図 9.2(B, C, D)のグラフを見ることができます。

(B)は、参照画像に対する各フレームの温度因子。フレーム番号（横軸）が増えるほど電子線損傷が増えますので、高分解能側の情報が抑制されている（B-factor の負値が大き）ことを示しています。(C) 参照画像に対する各フレームのスケール。フレーム番号が増えるほど電子線損傷が増えますので、反比例して小さくなります。(D)画像内の粒子移動度。電子顕微鏡画像ごとに計算されます。青○；粒子のスタート位置、黒線：粒子の軌跡（x 3 0 倍に協調）。この例では、同じ画像内でも電子線照射により様々な向きで粒子が移動していたことがわかります。

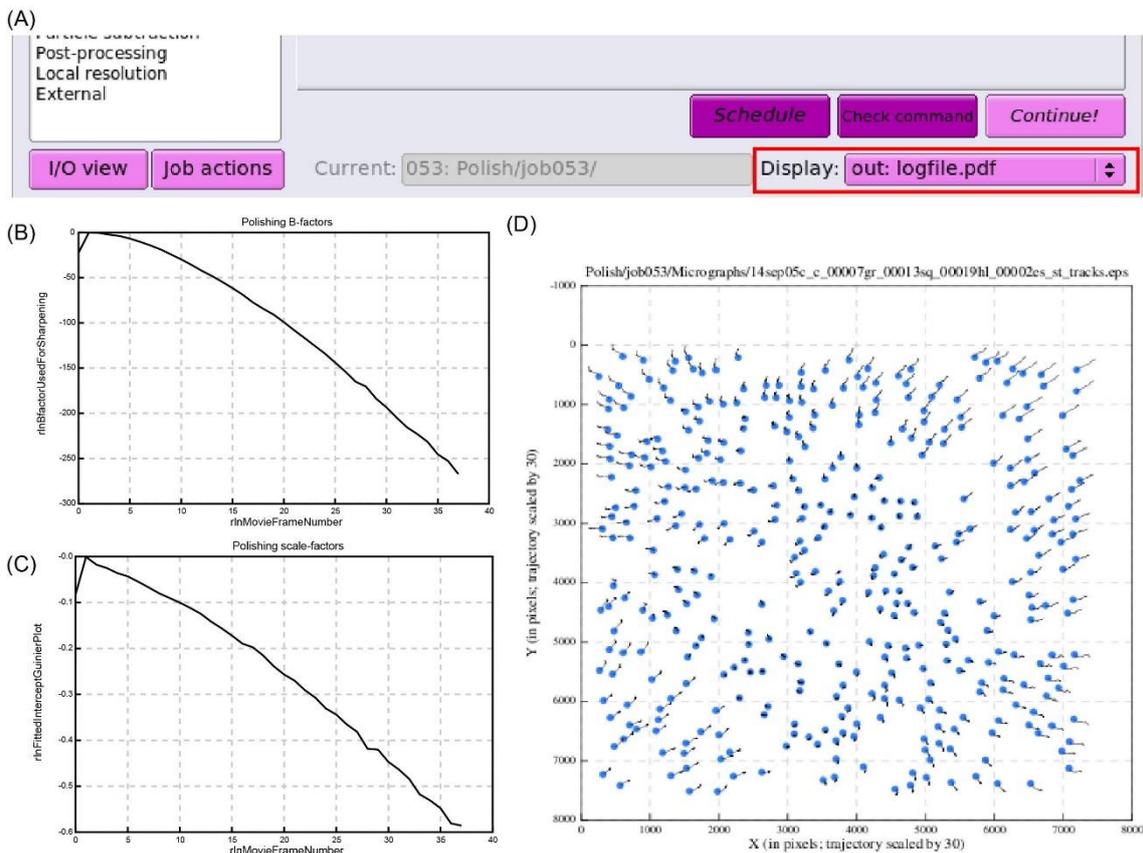


図 9.2 : Bayesian polishing --- Results after polishing

10. 局所空間における分解能評価

ポスト処理による空間分解能 FSC の評価だけでは、分子複合体といった分子量の大きな蛋白質の場合、部分的な揺らぎを適切に表すことはできません。そこで、プログラム ResMap を Relion 中から起動し、局所空間における分解能評価を行います。Relion では、Alp Kucukelbir と Hemant Tagre の開発した Resmap(文献4)を起動できますが、自前のプログラムも実装されています。文献4の方法とは異なり、Relion に実装された Resmap は、ソフトマスクを使用して各部分ごとの分解能を postprocess 同様に計算する方法です。

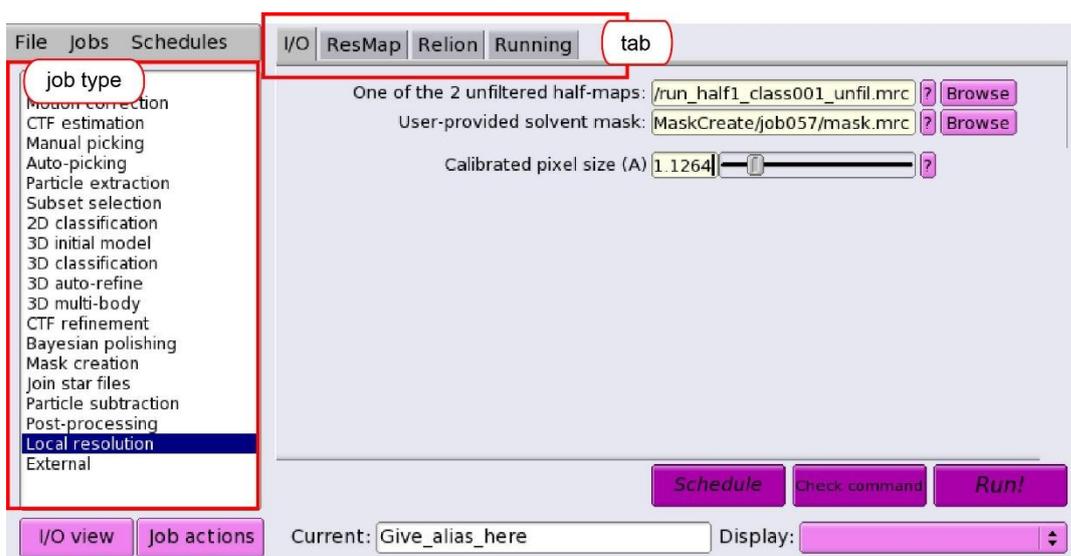


図 10.1 : Local resolution

左側の job-type ブラウザーの **Local resolution** をクリックすると、図9の画面へ変化します。図 10.1 の I/O タブ中を次のように変更します。

- One of the two unfiltered half-maps: **Refine3D/job059/run_half1_class001_unfil.mrc**
⇒7 で最終の Postprocess で計算時に利用した 3D マップ (**job+3 ケタ数字は適宜変更**)

- User-provided solvent mask: **MaskCreate/job057/mask.mrc**
⇒7 で最終の Postprocess で計算時に利用したマスク (**job+3 ケタ数字は適宜変更**)

- Calibrated pixel size (A) **1.1264**
⇒ときどきモデル構築を行っている際にスケールが合っていないことがあり、その際

Postprocess,Mask,LocalResolution の 3カ所で一致させて変更する必要がある。今回は、Relion の Resmap を使用するため、

ResMap タブ中で次のように指定します。

- Use Resmap? **No**

続いて、Relion タブ中で次のように指定します。

Relion タブ中で次のように指定します。

- Use Relion? **Yes**

- User-provided B-factor: **-60**
⇒Postprocess で使用したときと同じぐらいの値がよい。
- MTF of the detector (STAR file):
⇒Postprocess と同じ。Relion-3.1 で最初から計算していればファイル名指定の必要なし。

Running タブを押して、

- Number of MPI procs : **28**
- Number of threads : **1**
- Submit to queue : **No**

以上を入力後、タブ右下の **Run!** ボタンを押すと計算が始まります。計算後は、図 10.2(A) の **Display: logfile.pdf** を選択することで、図 10.2(B) の局所的な分解能領域に関するヒストグラムを見ることができます。おおよその分解能の範囲を確認できますので、つぎの chimera 確認時の surface color のカラーマッピングの数値範囲を決める際の目安にしてください。

Chimera(p 3 1 参照)を使って、local resolution を表示させるためには、まず top_directory/PostProess/job059/postprocess.mrc (**job+ 3 ケタ数字は適宜変更**) を読み込みます。続いて、Chimera の Tools->Volume data->Surface color を選択します。すると surface color ウィンドウ (図 10.2(C)左) がポップアップします。ウィンドウ中で、更に表示モードを by volume data value にし、使用する volume ファイルとして先の top_directory/LocalRes/job050/relion_localres.mrc (**job+ 3 ケタ数字は適宜変更**) を選びます。すると図 10.2(C) のような画面を表示できます。今回は、2.2 (青) , 2.4 (シアン) , 2.6 (緑) , 2.8 (黄) . 3.0 (赤) というカラーコードにしています。

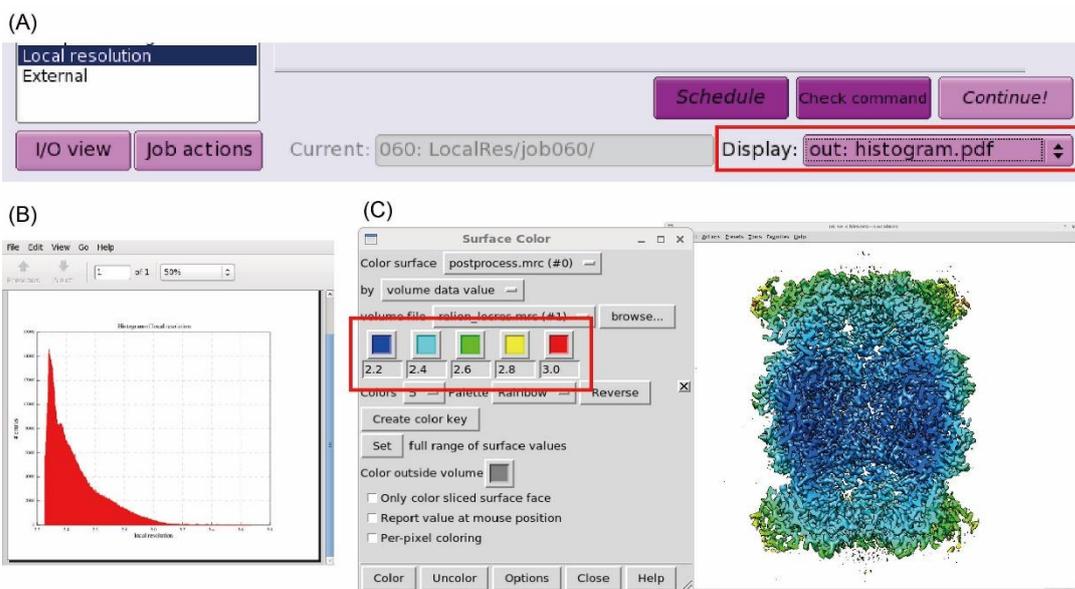


図 10.2 : Chimera-Local resolution

11. 手系の確認

電子顕微鏡の非傾斜投影像だけから構築した3Dモデルの場合、正しい手系を求めることができません。そのため、ここまでの操作で決定した3Dモデルの手系は、本来とは反対である可能性が半分あります。その場合、本来右巻きであるはずの α ヘリックスが左巻きに見える場合、あるいは、構造既知のホモログやモチーフが存在するにも関わらず、求めたデンシティマップに対して、既知構造が全くフィットできないことなどで気づくことができます。このような手系が反対の場合には、以下のコマンドで手系を戻すことができます。

```
>relion_image_handler --i postprocess.mrc --o postprocess_inv.mrc --invert_hand↵
```

relion の計算上ではどちらの手系でも構わないだけでなく、初期モデルの手系が正しい場合でも3D精密化計算の最中に反対の手系になることがあります。正しい手系はマップを解釈する上で大切ですので、最終計算結果の手系に対しては必ず確認を行うようにしてください。

12. Relion-3.1 の互換性に関する注意事項

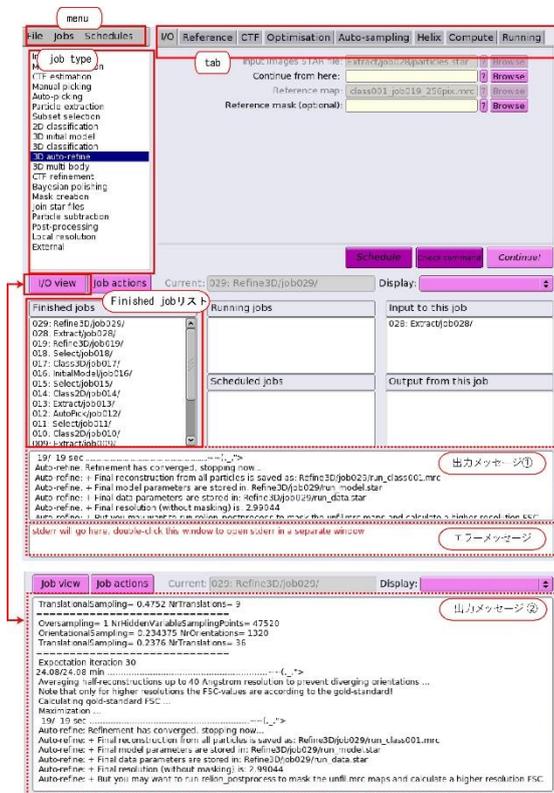
Relion-3.0 以前のプログラムで処理された解析結果を利用して、Relion-3.1 で新しく追加された高次の収差補正などを行うことはできます。

ただし、Relion-3.1 が出力する STAR ファイル（単粒子解析の結果を記述）の形式内で、高次収差の精密化をするために optics groups など新しい記述を導入しました。そのため従来のものとは異なりますが、Relion-3.1 でのプログラム実行時に自動的に変換してくれますので、概ね問題はありません。経験上、ときどき稼働しない（計算中のプログラムが途中で止まる）ことはありますが、その場合は解析のステップを1段階戻って計算し直せば正しい計算結果が得られようです。

一旦、**Relion-3.1 で処理をすると、それ以降生成された STAR ファイルを使った Relion-3.0 以前での解析はできなくなります。**この点は、留意して解析を行ってください。

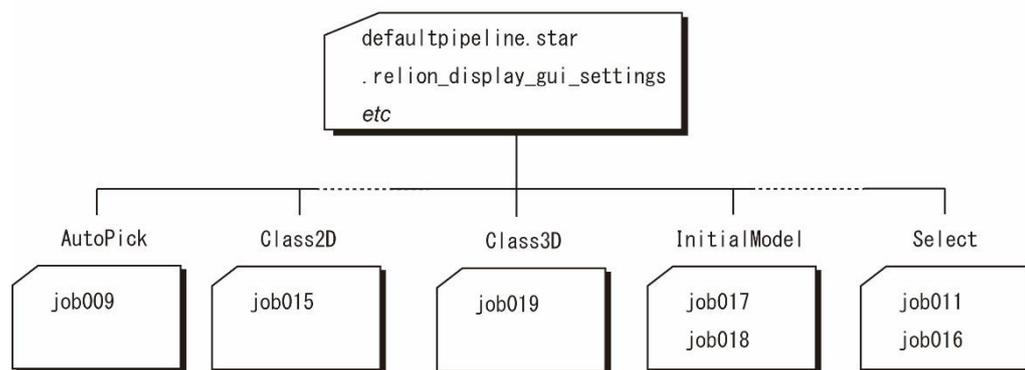
Appendix A. 計算結果とディレクトリ構造

Relionでの計算が完了するごとに、Finished job リストが増えていきます (図 A.1)。リストをクリックすると、出力メッセージ①に結果が表示されます。その際、I/O view を押すと出力メッセージ①から出力メッセージ②に変更され、より詳細な情報も確認できます。



(図 A.1) Relion ウィンドウの説明 :

top ディレクトリ (relion を起動した場所)



(図 A.2) Relion ワーキングディレクトリ

また、解析後の結果データは、すべてジョブタイプ (図 A.1) ごとに上記のようなディレクトリへ振り分けられます (図 A.2)。更に、ジョブタイプディレクトリ (AutoPick, Class2D, Class3D etc.) の下には、ジョブごとに3ケタの通し番号 (job001 etc.) が付与されたディレクトリが用意され、入出力データなどが保存されています。これらのディレクトリ構造は、relion の Window 左下にある Finished jobs リスト (図 A.1) と関連付けられていますので、

名前の変更により、これまでの計算結果が簡単に見られなくなるばかりでなく、今後の行う計算も不可能になる場合があります。一方で、いろいろな条件を試行するための計算や失敗したジョブなどは、残しておいてもスペースの無駄になりますので、GUI 上から次の操作によって消去することをお勧めします。

まず、消したいジョブを Finished jobs リスト中でクリックします。続いて、**Job actions** ボタンを押し Delete を選択します。すると消したいジョブのディレクトリが Trash というディレクトリへ移動します。最後に Menu の中から File (図 A.1 左上) を押し続けると複数の選択候補がでてくるので、その中から “Empty trash” を選ぶとハードディスク上にある Trash 以下のデータが消去されます。一旦消去するとデータを元に戻すことはできませんので、気を付けて操作してください。

参考文献

1. Sjors H.W. Scheres 博士” Single-particle processing in Relion-3.1”

(ftp://ftp.mrc-lmb.cam.ac.uk/pub/scheres/relion31_tutorial.pdf)

Relion 開発者によるオリジナルマニュアルです。今回のテキストは、このマニュアルに準拠して作成していますが、すべてを取り扱っているわけではありません。更に詳細な情報にアクセスされたい場合には、ぜひご一読ください。

2. Campbell M.G. et al., *Elife* 4 e06380 (2015), “2.8 Å resolution reconstruction of the Thermoplasma acidophilum 20S proteasome using cryo-electron microscopy”

<https://doi.org/10.7554/eLife.06380.001>

今回使用したテストデータの論文です。当時は relion-1.3 で 2.8 Å に到達しています。新しく開発された relion-3.1 で得られた結果 (テキスト内では 2.3 Å 分解能) と比較してもらうと、解析技術の進歩を実感して頂けると思います。

3. Zivanov J. et al., *IUCr* 7 253 (2020), “Estimation of high-order aberrations and anisotropic magnification from cryo-EM data sets in RELION-3.1”

<https://doi.org/10.1107/S2052252520000081>

or <https://www.biorxiv.org/content/10.1101/798066v1>

高次の収差補正に関する論文です。relion-3.1 の特徴の一つで、高分解能解析を可能にした技術です。

4. Kucukelbir F.J. et al., *Nature meth.* 11 63 (2014), “Quantifying the Local Resolution of Cryo-EM Density Maps”

<http://dx.doi.org/10.1038/nmeth.2727>

謝辞

本講習会は、AMED 創薬等先端技術支援プラットフォーム (BINDS) のサポートを受けて開催しています。課題番号 JP20am0101118j。