Beautiful crystallographic maps with FEM (Feature Enhanced Map)

Pavel Afonine

Computation Crystallography Initiative
Physical Biosciences Division
Lawrence Berkeley National Laboratory, Berkeley CA, USA

07 – May – 2014

Preamble

- Crystallographic maps are used to derive atomic model
  - **Map quality defines quality of atomic model**

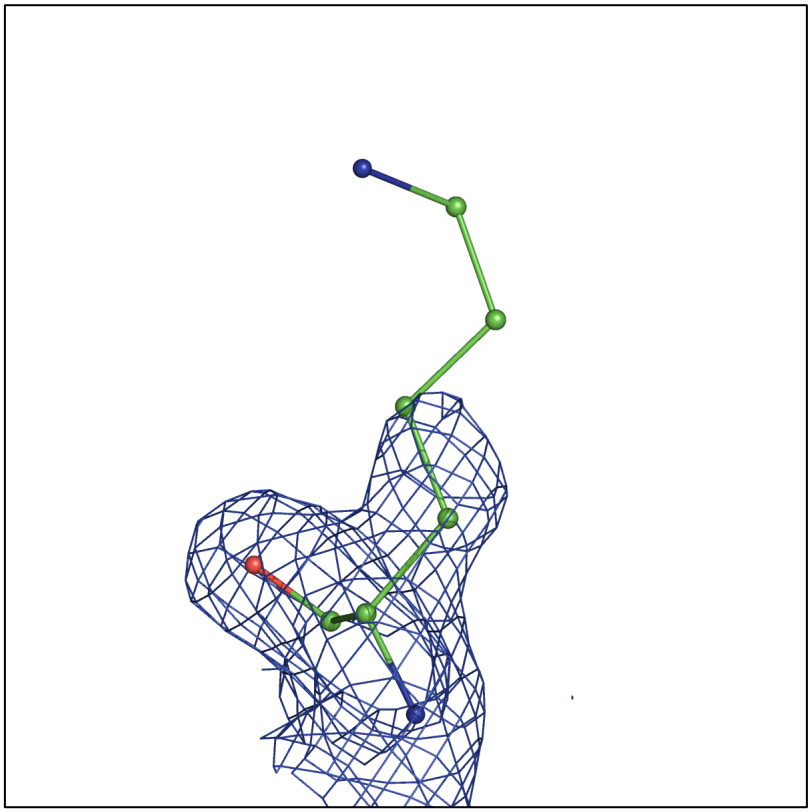
- Crystallographic maps always
  - **Have errors and noise**
  - **Model biased**

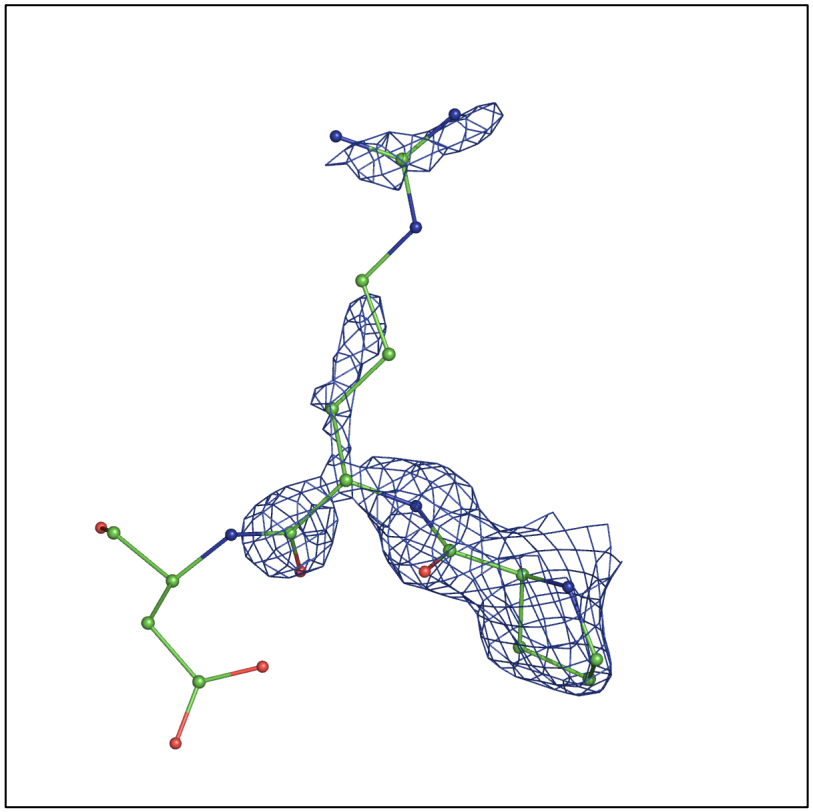
- If map is bad a result of map interpretation (atomic model) will be always questionable
  - **It is vital to obtain the best possible map to work with (before investing or wasting time interpreting it!)**

Macromolecular maps: Problem statement

Problem statement

- Macromolecular maps are never perfect
  - **Weak or missing density**
  - **Ambiguity in interpretation**

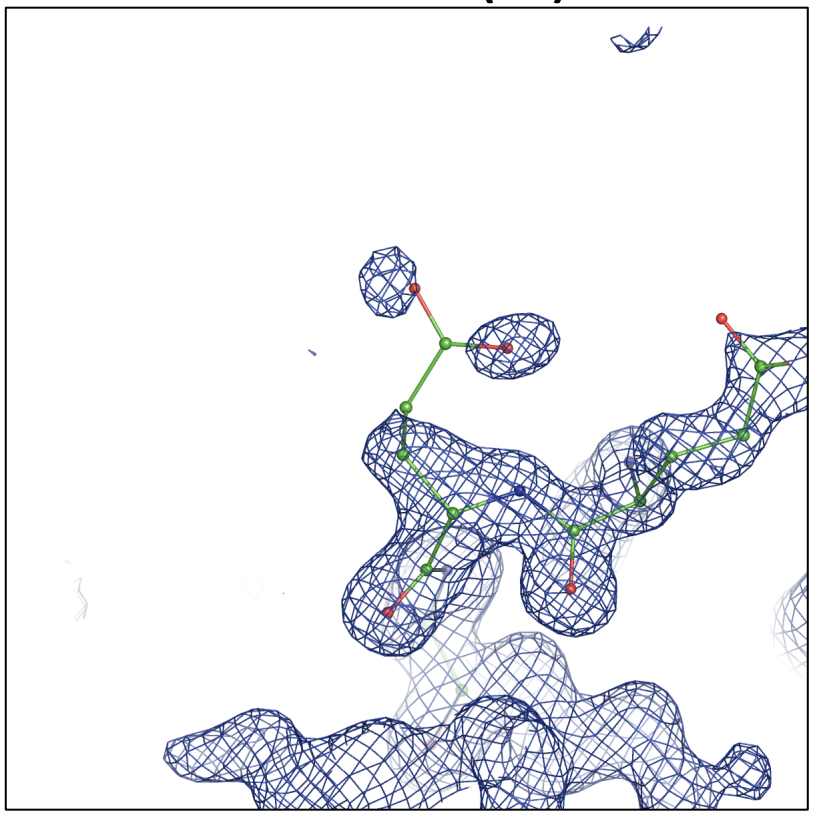




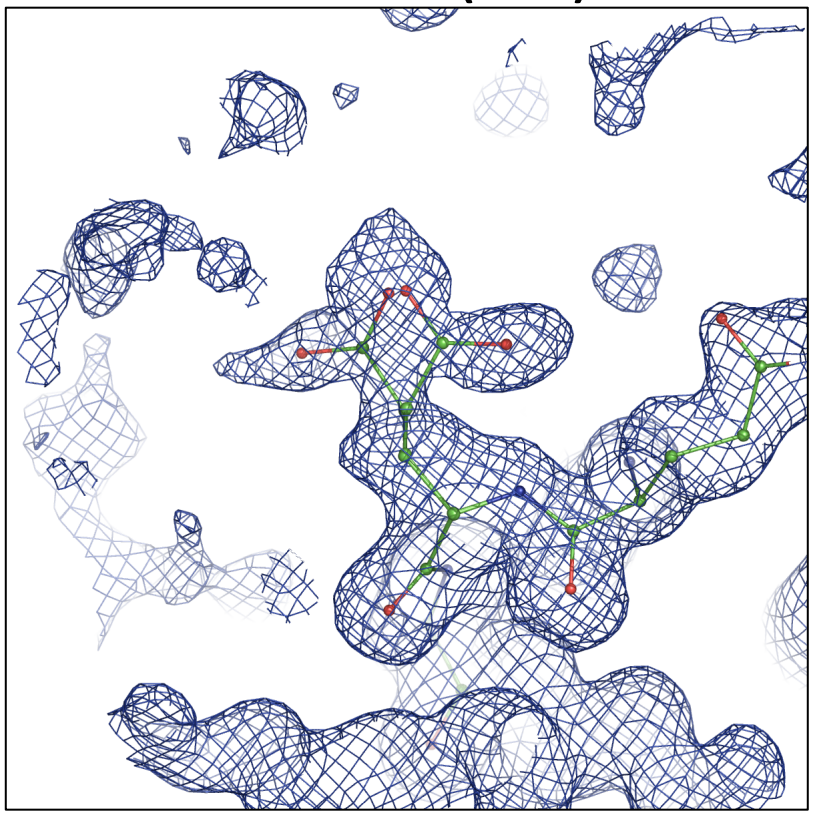
Problem statement

- Lowering map contouring levels
  - **May or may not show more features**
  - **Always shows more noise**
  - **Risk to see and interpret model bias**

2mFo-DFc (1σ)



2mFo-DFc (0.5σ)



Sources of crystallographic map imperfections

Sources of crystallographic map imperfections

1. Errors:
   - **Finite resolution of measured reflections**
   - **Incompleteness of data (missing reflections within the resolution)**
   - **Experimental errors in measured reflections**
   - **Errors in atomic model parameters**
2. Signal strength may vary drastically across unit cell volume:
   - **Ordered atoms *vs* partially occupied, highly mobile (flexible loops)**
   - **Heavy atoms *vs* typical protein (C, N, O) atoms**
3. Model bias:
   - **Crystallographic maps are calculated using model phases or a combination of model phases with experimental phase information**
     - **Errors in atomic model result in map errors that may be erroneously interpreted**

All three sources are ***always*** present all together

Decades of effort to improve maps

1. Weighting schemes:
   - **Luzzati, 1953; Woolfson, 1956; Sim, 1959; Raman, 1959; Ramachandran & Raman, 1959; Srinivasan, 1961; Ramachandran & Srinivasan, 1961, 1970; Main, 1979; Vijayan, 1980; Urzhumtsev *et al.*, 1996; Vellieux & Dijkstra 1997; Read 1997; Sheldrick, 2008**

2. Maps from ensembles of perturbed models or structure factors:
   - **Perrakis *et al.*, 1997; Rice *et al.*, 1998; Reddy *et al.*, 2003; Praznikar, 2009; Lang et al, 2014**

3. Various kinds of OMIT maps:
   - **Bhat & Cohen, 1984; Bhat, 1988; Hodel *et al.*, 1992; Guncar et al, 2000; Vellieux & Dijkstra 1997; Terwilliger et al, 2008; Cowtan, 2012; Echols & Afonine, 2014**

4. Density modification techniques:
   - **Reviews: Podjarny *et al.*, 1996; Zhang *et al.*, 2006; Cowtan, 2010**

1-4 aim to address one or a few specific problems but not all or at once

New approach: Feature Enhanced Map (FEM)

FEM (Feature Enhanced Map): facts

- FEM procedure modifies $2m\mathbf{F}_{obs}-D\mathbf{F}_{model}$ σ_A-weighted map to:
  - **Reduce noise**
  - **Retain existing features**
  - **Enhance existing weak features**
  - **Reduce model bias**
- FEM:
  - **Fast to compute: from a few to several minutes**
  - **No user adjustable parameters**
  - **Requires basic input: PDB model, Fobs or Iobs**
  - **Eliminates the need to choose (arbitrary!) map contouring cutoffs**
  - **One map to use for all purposes**
  - **Good for X-ray or neutron data**
- Availability:
  - **Phenix build 1.9-1692 and up: http://www.phenix-online.org/**
  - **GUI**
  - **Command line:**

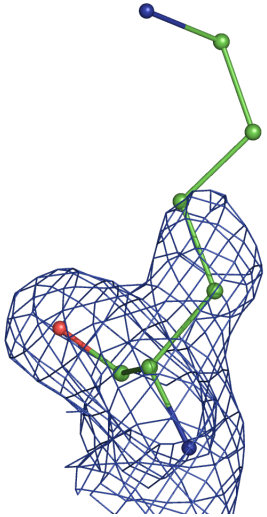
    phenix.fem model.pdb data.mtz

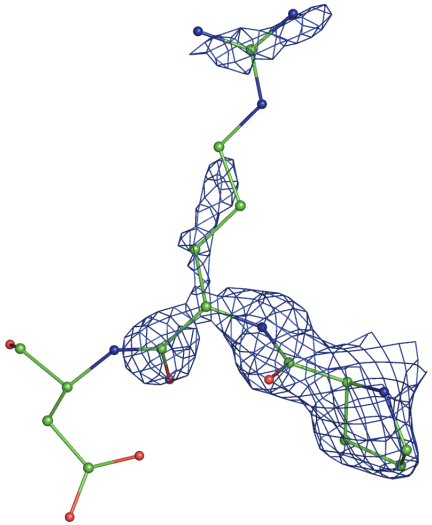
FEM (Feature Enhanced Map): limitations and warnings

- Not absolutely all noise and bias may be removed
  - **Some may still be present**
- Signal arising from features on average should be greater than noise
  - **FEM most useful for intermediate, final or next to final maps**
  - **FEM may not be useful for poor initial maps**
- Resolution limits
  - **Tested at 1..4 Å**
  - **May well work at resolution beyond the above limits (untested)**

FEM: examples of expected improvement

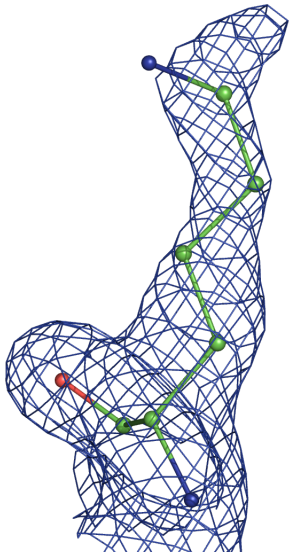
Original 2mFo-DFc maps

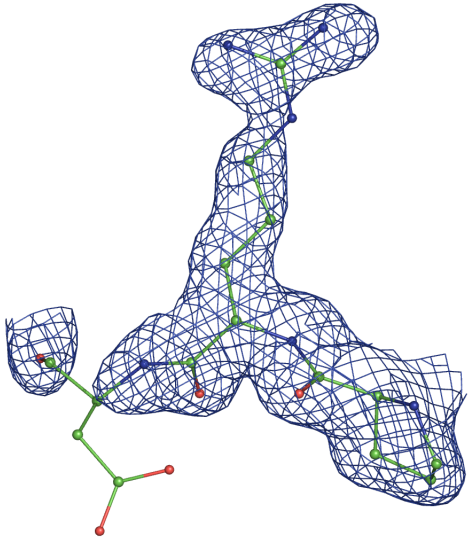




FEM

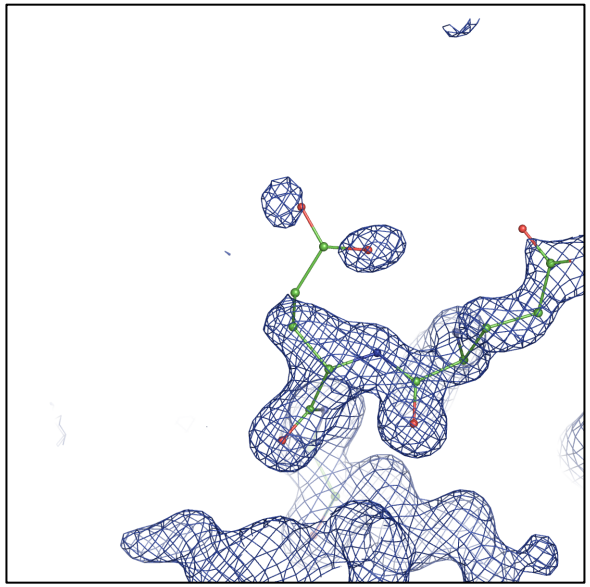
New (FEM) maps



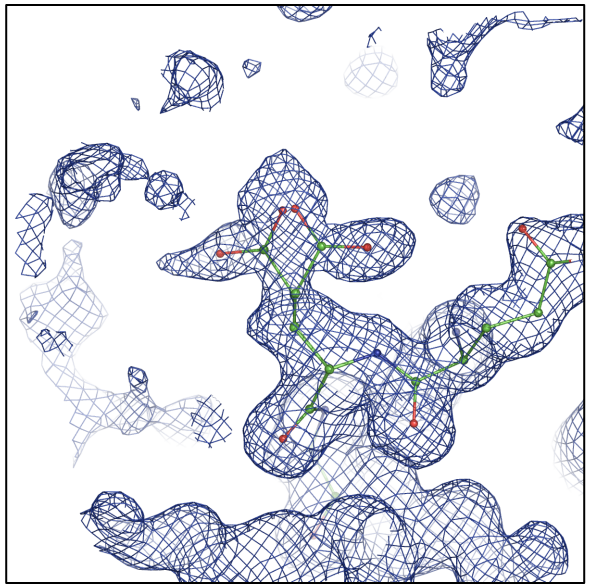


FEM: examples of expected improvement

2mFo-DFc , 1σ

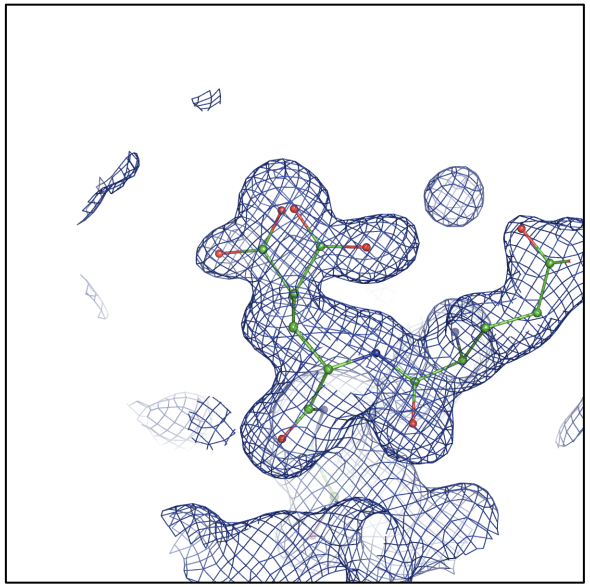


2mFo-DFc , 0.5σ



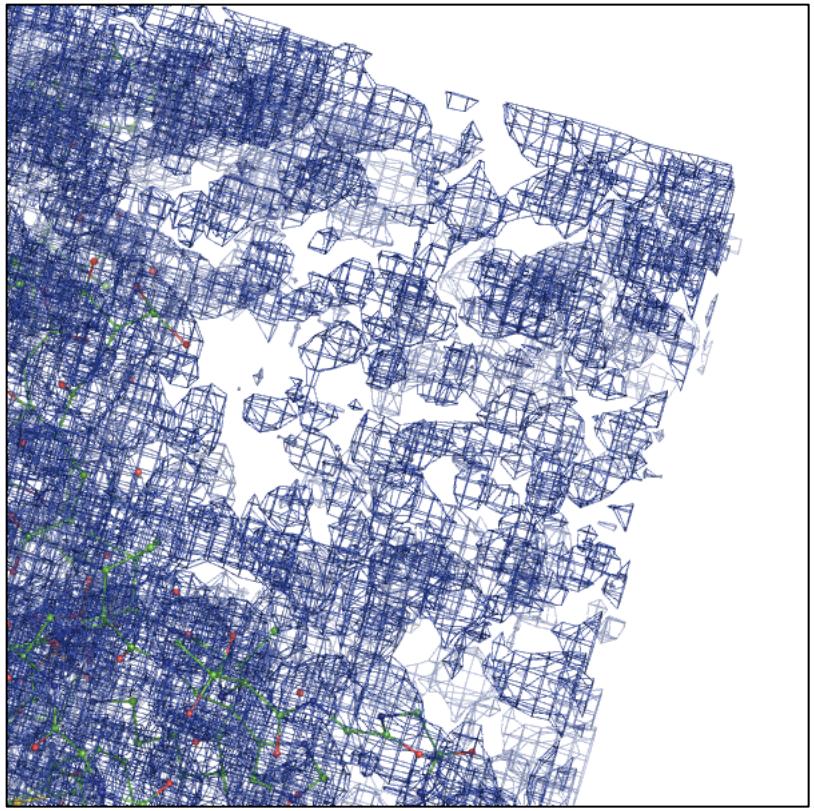
FEM

FEM, equivalent 1σ



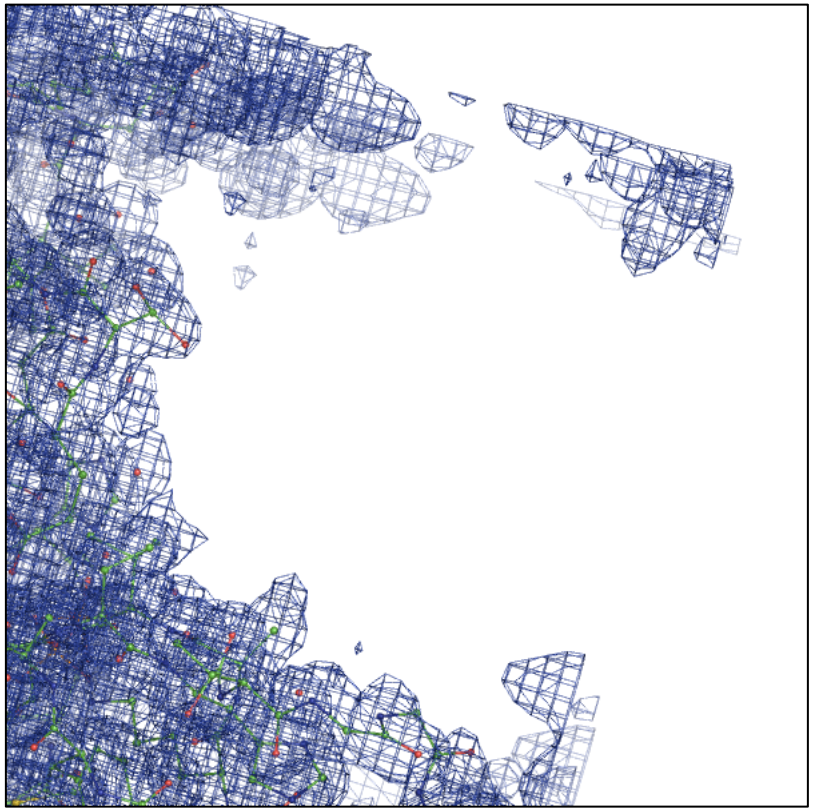
FEM: examples of expected improvement

Original 2mFo-DFc , 0.5σ



FEM

FEM, equivalent 0.5σ



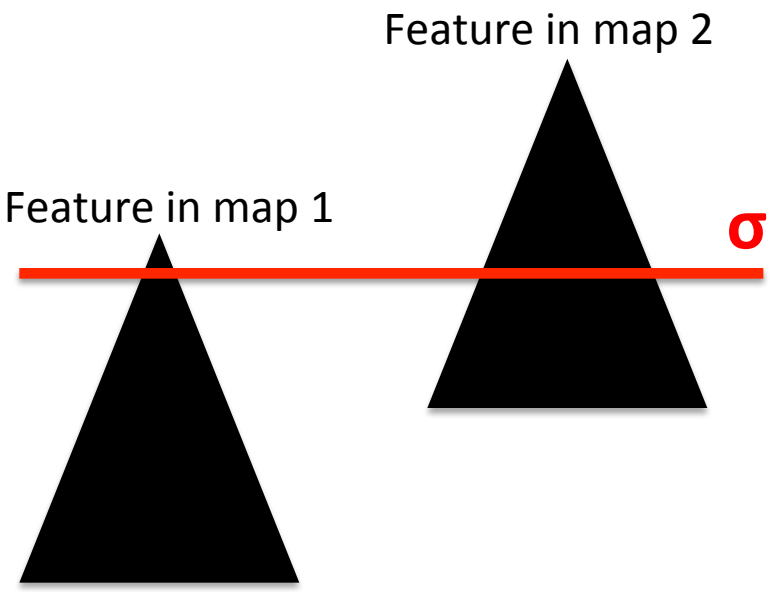
Map comparison and display

A side note about proper comparison of crystallographic maps

Map comparison and display: problem illustration

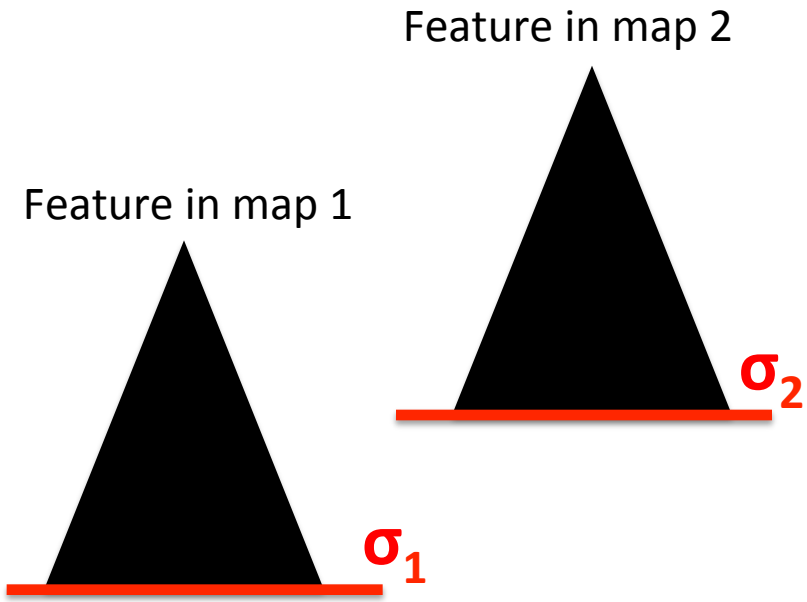
- Comparison and display of maps at identical standard deviation contouring levels (at identical “sigmas”) is incorrect and misleading
  - **A. Urzhumtsev, P.V. Afonine, V.Y. Lunin, T.C. Terwilliger & Paul D. Adams. (2014). Metrics for comparison of crystallographic maps. Submitted to Acta D**

Comparison at identical threshold



- Selected objects (above red line) are obviously different *(while we know they are identical!)*

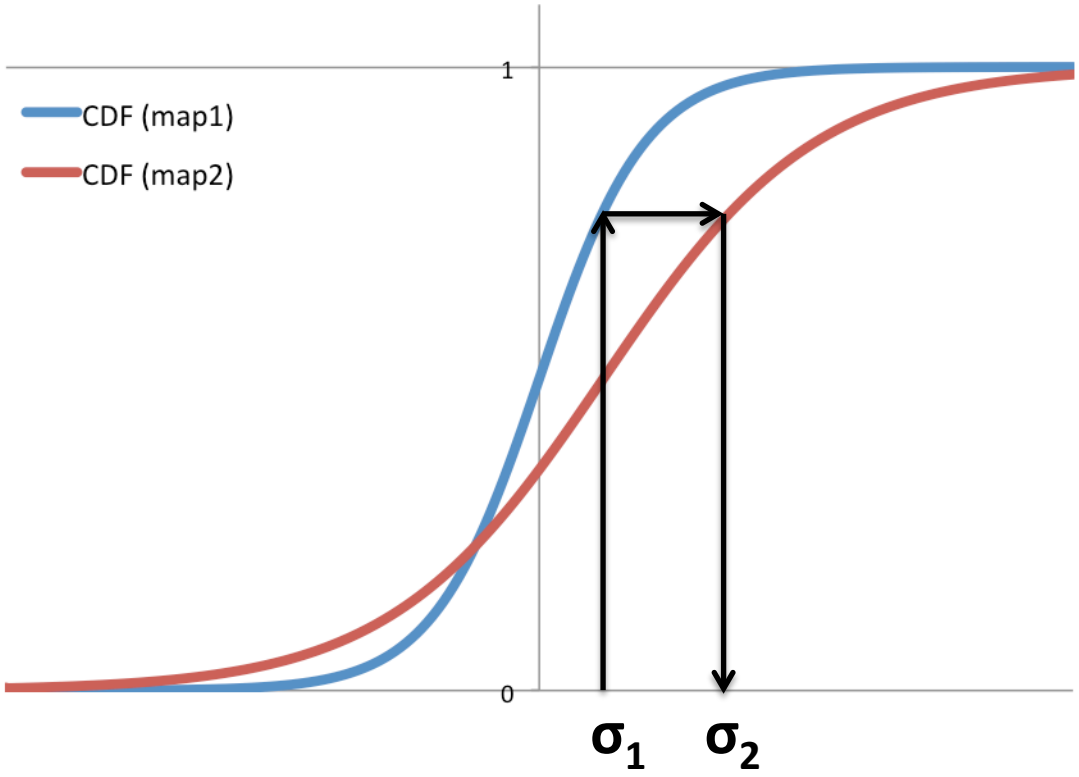
Comparison at different thresholds



- Selected objects (above red line) are identical *(as they actually are!)*

Map comparison and display: how to choose contouring levels

- Schematic illustration of choosing equivalent map contouring thresholds:



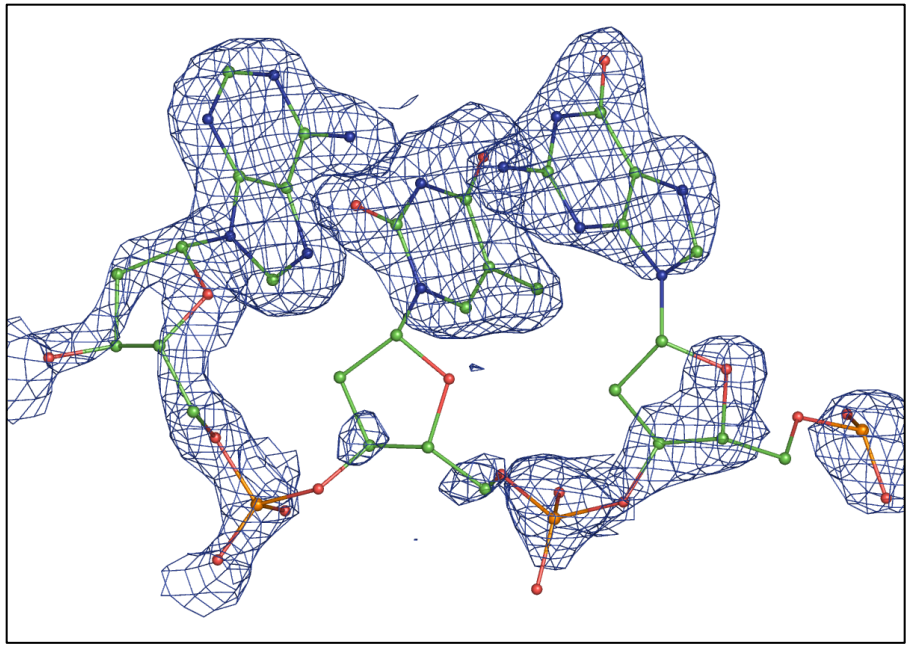
1. Compute cumulative distribution functions (CDF) for two maps in question.
2. Given contouring threshold for map 1 (σ_1), find corresponding value of CDF.
3. Lookup contouring threshold of map 2 (σ_2) that corresponds to the same value of the CDF of map 1.
4. Contouring thresholds σ_1 and σ_2 select equivalent fractions of in both maps (equal volume maps).

Using FEM (Feature Enhanced Map): real-life examples

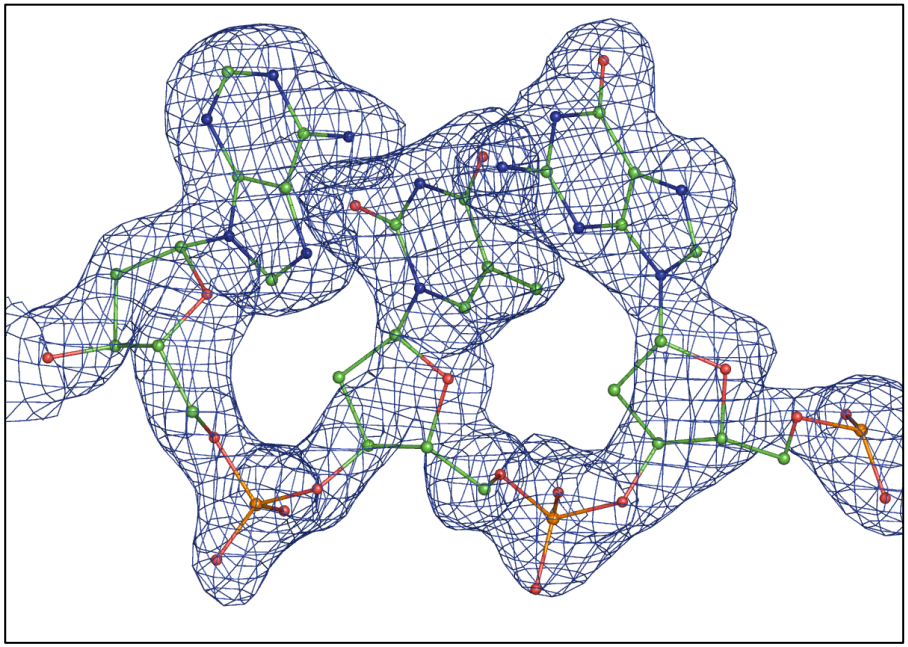
FEM: example #1

- PDB code: 1NH2, resolution 1.9Å, showing E6-E8

2mFo-DFc , 1σ



FEM, equivalent 1σ



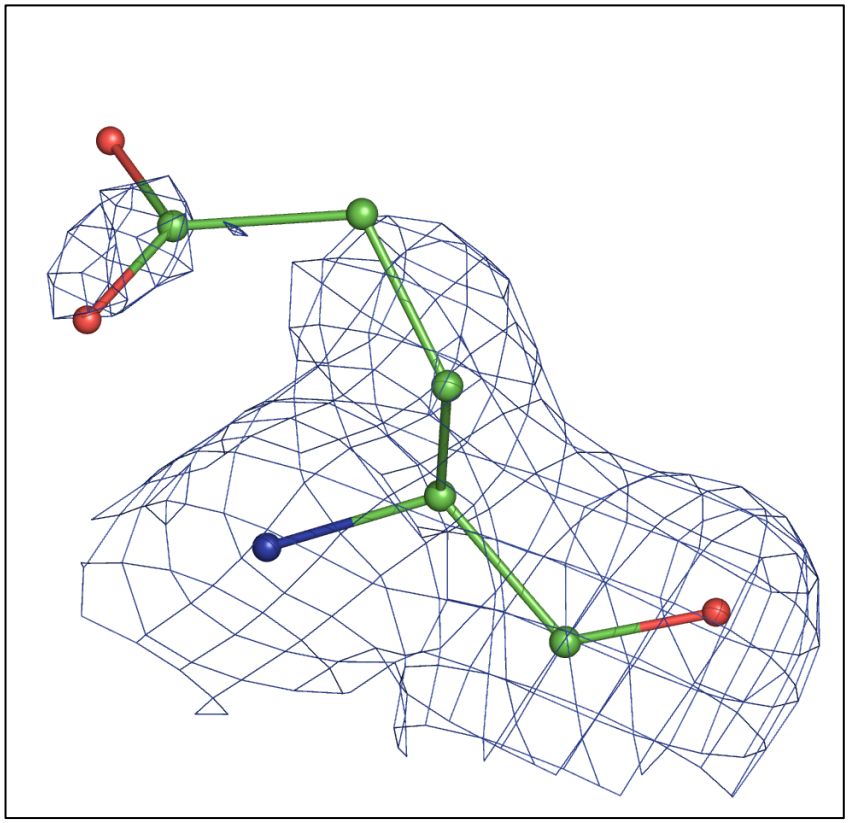
FEM

This is an example of typical improvement observed throughout the whole structure

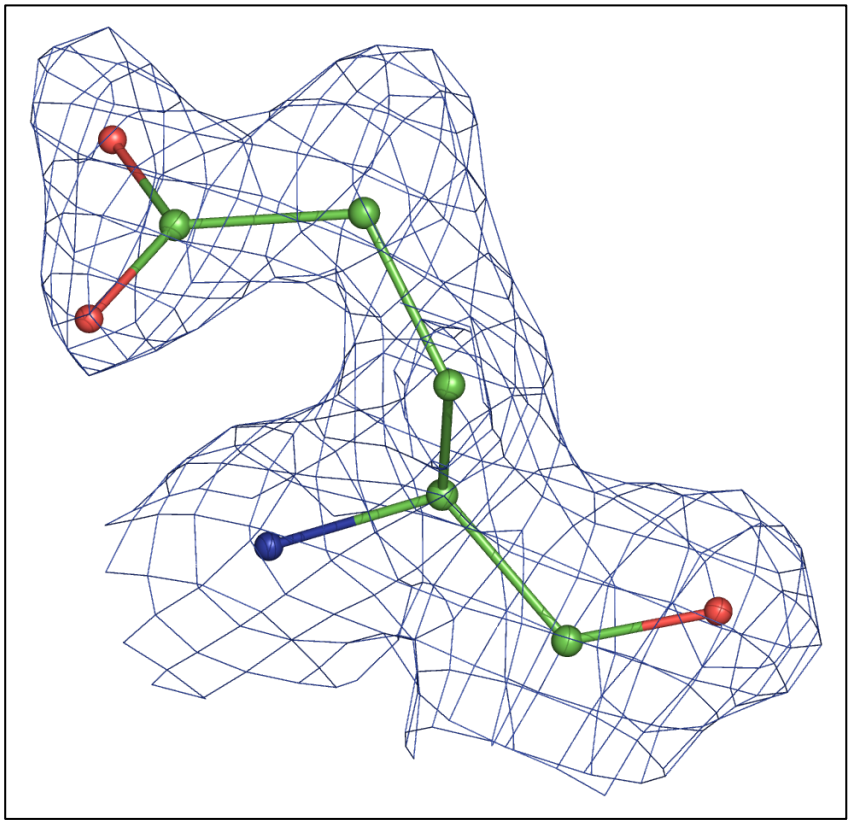
FEM: example #2

- PDB code: 1F8T, resolution 2.2Å, showing L81

2mFo-DFc , 1σ



FEM, equivalent 1σ



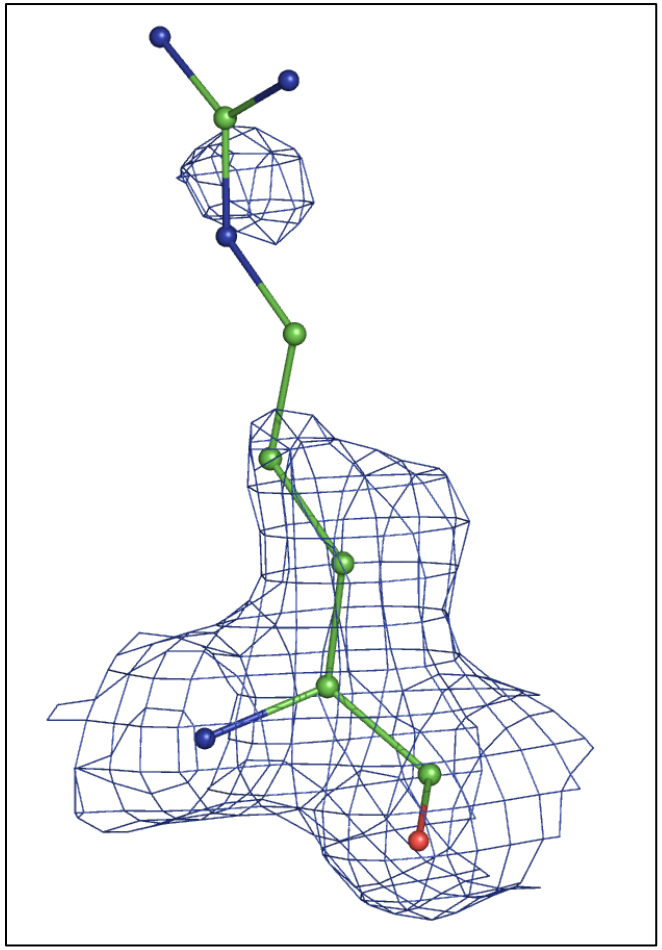
FEM

This is an example of typical improvement observed throughout the whole structure

FEM: example #3

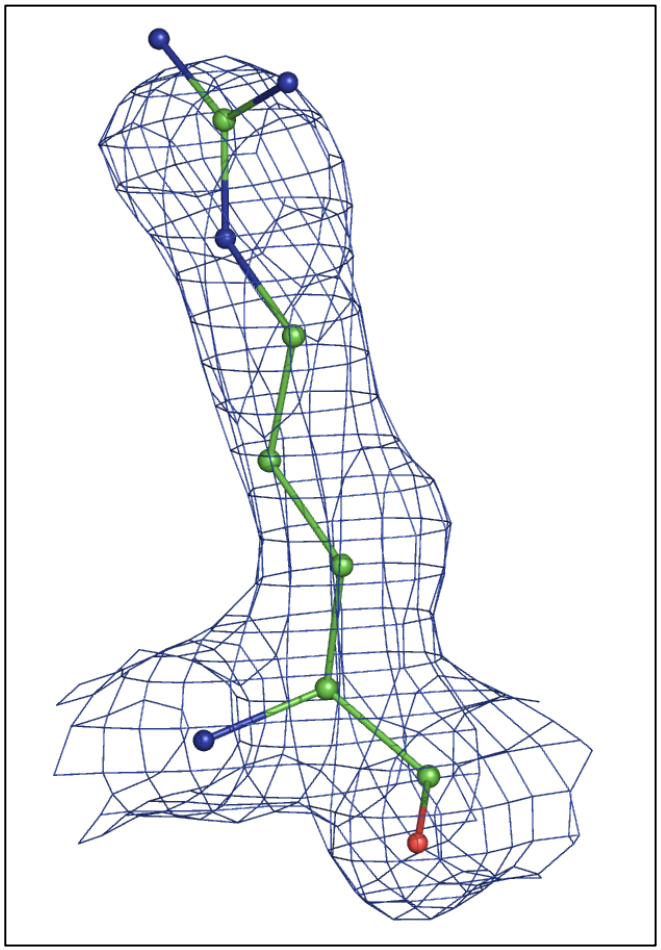
- PDB code: 1F8T, resolution 2.2Å, showing L74

2mFo-DFc , 1σ



FEM

FEM, equivalent 1σ

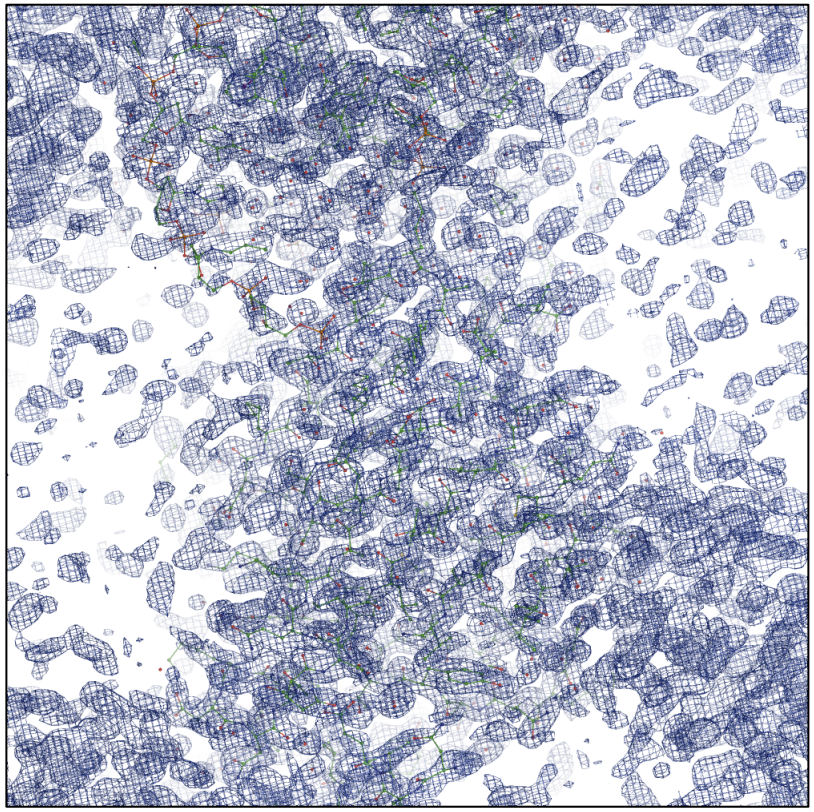


This is an example of typical improvement observed throughout the whole structure

FEM: example #4

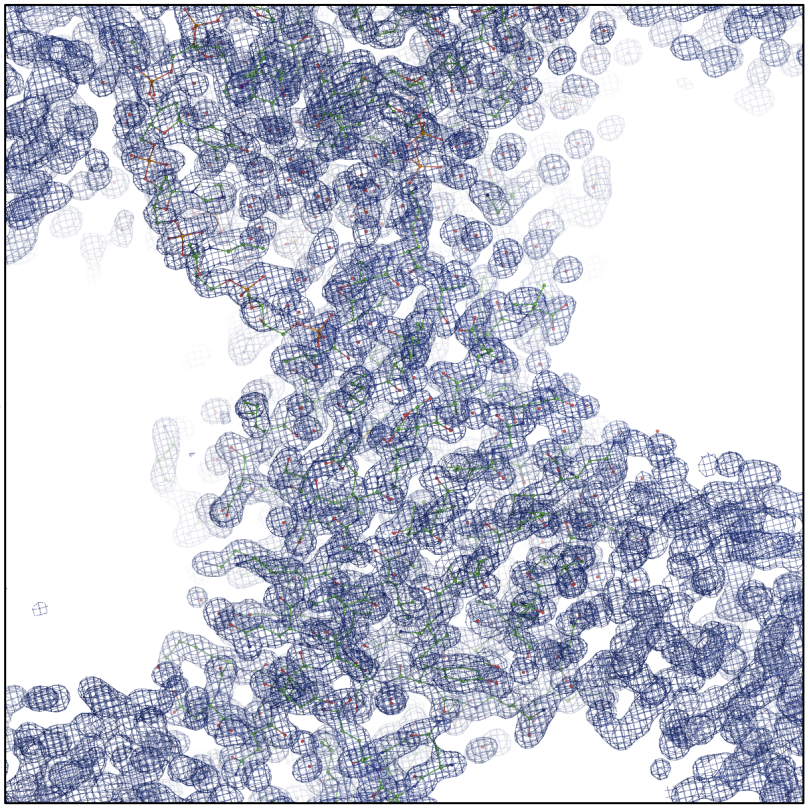
- PDB code: 1NH2, resolution 2.2Å

2mFo-DFc , 1σ



FEM

FEM, equivalent 1σ

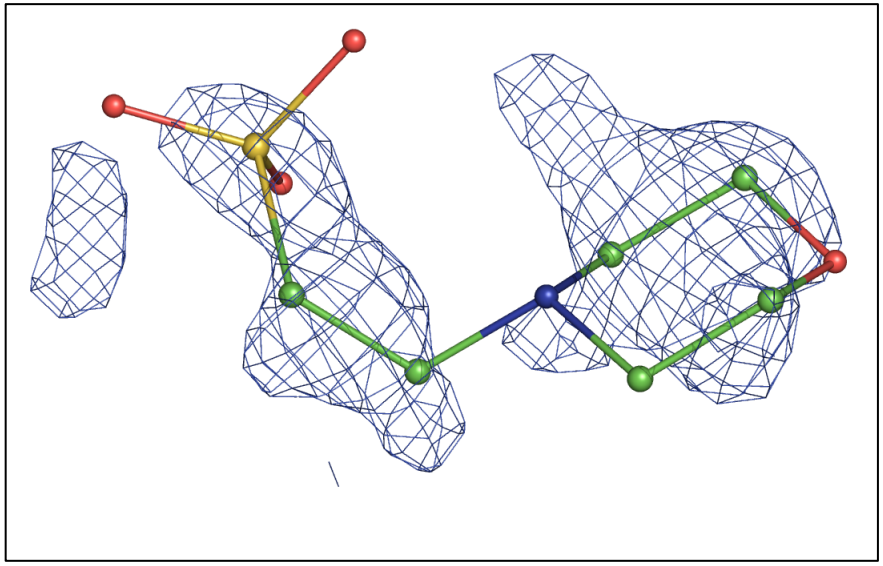


This is an example of typical improvement observed throughout the whole structure

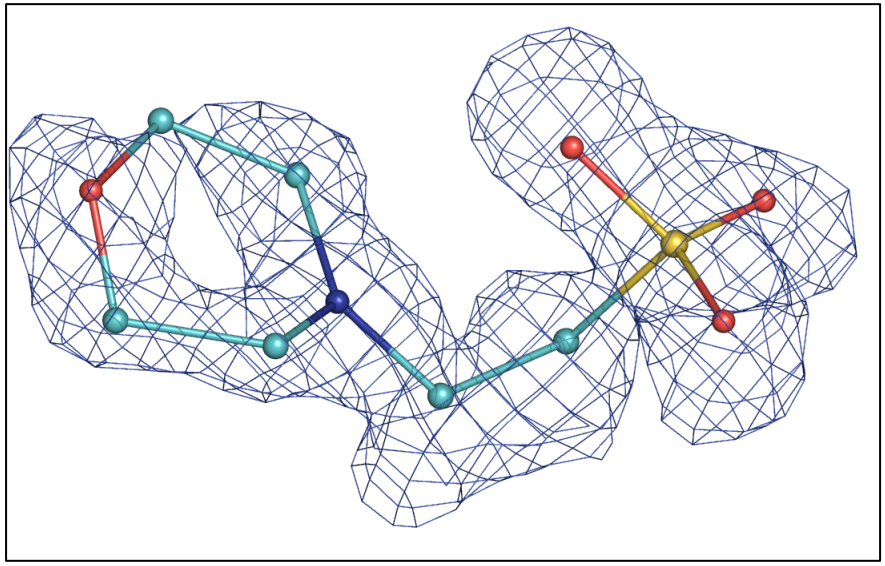
FEM: example #5

- PDB code: 1SE6, resolution 1.6Å: incorrectly built ligand into poor map
  - **Example from:**
    - **Techniques, tools and best practices for ligand electron-density analysis and results from their application to deposited crystal structures.**
      E. Pozharski, C. X. Weichenberger and B. Rupp, Acta Crystallogr D69, 150-167 (2013)

2mFo-DFc , 1σ



FEM, equivalent 1σ



FEM

FEM: methods

FEM calculation methods in a nutshell

- **Efficient map randomization and combination**
  - **Idea:**
    - Compute large ensemble of slightly perturbed maps and combine them all into one map
  - **Rationale:**
    - Map artifacts are more sensitive to randomization than the signal
  - **Challenges:**
    - Combining maps may blur the signal, so map sharpening is necessary
    - How to keep many maps in memory?
- **Signal equalization**
  - **Idea:** Make strong and weak signal similar in strength
  - **Rationale:**
    - It is not important how strong the strong signal is, as long as it is *strong enough* to be reliably distinguished from the noise and interpreted in terms of atomic model
  - **Challenges:**
    - Not selective (noise may be enhanced as well as signal) - Noise needs to be eliminated as much as possible
- **Restrict map to regions where there is convincing evidence of density**
  - Use OMIT map to reduce model bias

FEM calculation protocol

1. OMIT map filter
- Compute composite residual OMIT map: M_{cromit}
- Scale M_{cromit} by standard deviation (σ)
- Compute filter: $M_{filter}=0$ if $M_{cromit}<0.5\sigma$ else $M_{filter}=1$

2. Initialize collector of integer maps, IMC

3. For j in j=1,16:

a. Map randomization and averaging
For i, i=1,10:
- Compute 100 map coefficients and average them: $MC_{average}$
- Randomly remove 5% of terms from $MC_{average}$: $\widetilde{MC}_{average}$
- Compute Fourier map M_i from $\widetilde{MC}_{average}$
- Scale M_i by standard deviation
- Truncate low values: set $M_i=0$ if $M_i<0.5\sigma$
- Eliminate regions in M_i with small volume

b. Sharpen M_i , M_{sharp}

c. Histogram equalize M_{sharp} , M_{HE}

d. Filter M_{sharp} by OMIT map: $M_{filtered} = M_{HE} * M_{filter}$

e. Add $M_{filtered}$ to IMC

4. Compute median map M_m from 16 maps in IMC, which is resulting Feature Enhanced Map, $M_{FEM} = M_m$

Base map for FEM calculations

Methods: base map for FEM calculations

$$\mathbf{F}_{\mathrm{map}} = \begin{cases} \frac{w}{\mathrm{k_{total}}}(2\mathrm{mF_{obs}} - \mathrm{DF_{model}}), \varphi_{\mathrm{model}} & \text{acentric refletions} \\ \frac{w}{\mathrm{k_{total}}}\mathrm{mF_{obs}}, \varphi_{\mathrm{model}} & \text{centric reflections} \\ \mathrm{F_{fill}}, \varphi_{\mathrm{fill}} & \text{no } \mathrm{F_{obs}} \text{ in } [\mathrm{d_{min}}, \infty) \end{cases}$$

- Acentric and centric reflections: σ_A-weighted (Read, 1986) anisotropy removed and sharpened Fourier map coefficients corresponding to measured F_{obs}
- φ_{model} : model phases or combined with experimental phase information
- m, D calculated as described in (Urzhumtsev et al. 1996; Read, 1997)
- $F_{model} = k_{total} * (F_{calc} + F_{bulk})$ (Afonine et al., 2013)
- w=1, by default
- $B_{min}=\{B_1,\ldots,B_N\}$, N=number of atoms, is subtracted from all atomic B before F_{model} calculation so it becomes part of k_{total}
- Dividing by k_{total} sharpens the map and removes anisotropy
- $(F_{fill},\varphi_{fill})$ terms corresponding to unmeasured F_{obs}

Missing reflections

Missing reflections: facts

- Missing reflections: unmeasured Fobs in resolution range $[d_{min},\infty)$, where d_{min} is the highest resolution of dataset
- Missing terms in F_{map} (terms corresponding to unmeasured Fobs) can result in poor maps
  - **Lunin, 1988; Urzhumtsev, Lunin & Luzyanina, 1989; Lunin & Skovoroda, 1991; Tronrud, 1996; Cowtan, 1996; Lunina *et al.*, 2002; Urzhumtseva & Urzhumtsev, 2011**
- To improve map quality terms corresponding to missing F_{obs} in $[d_{min},\infty)$ are replaced with some values (F_{fill}, φ_{fill})
  - **Murshudov et al, 1997; Altomare *et al.*, 2008; Sheldrick, 2008**
- Ignoring missing reflections is equivalent to postulating that $\mathbf{F}_{map}=0$
  - **Since we have an atomic model we can get a better estimate for F_{map} than zero**

Accounting for missing reflections

- **FEM uses two approaches:**
  - **Obtain (F_{fill}, φ_{fill}) from *Resolve* density modified map (Terwilliger, 2003)**
  - **Use current model:**
    - (F_{fill}, φ_{fill}) = (F_{model}, φ_{model})
      - F_{model}, φ_{model} calculated using atomic model with unreliably placed atoms removed

Effect of missing data: example

- PDB code: 1NH2, 1.9Å

1NH2	
Resolution	% cmpl.
19.9-10.3	30. 7
10.3-8.5	37.6
8.5-7.1	59.0
7.1-5.9	79.6
5.9-4.9	77.2
4.9-4.0	71.4
4.0-3.4	86.2
3.4-2.8	98.5
2.8-2.3	99.7
2.3-1.9	99.2
∞-1.9	95.0

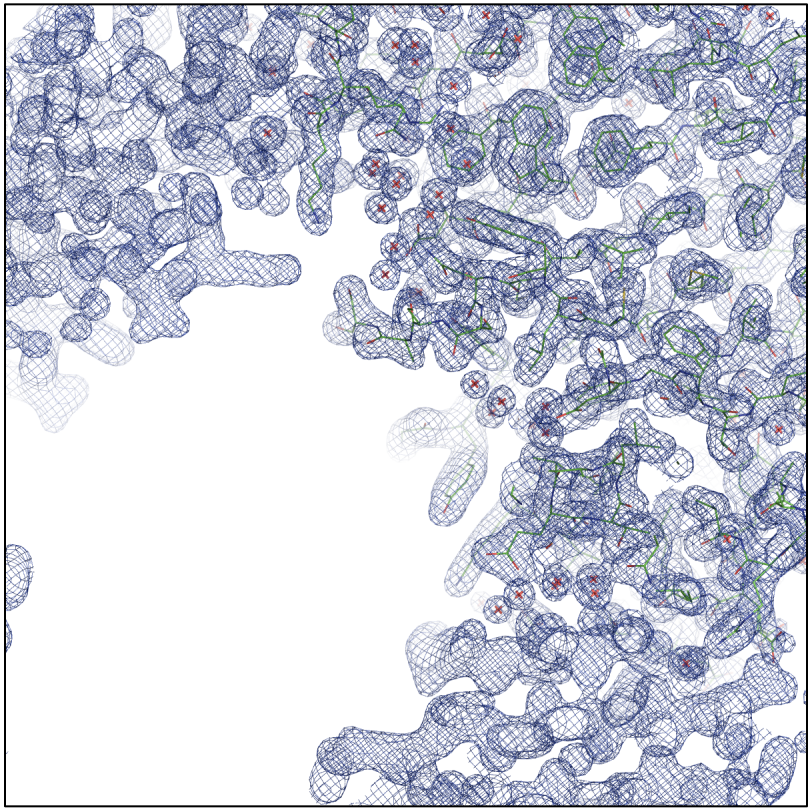
← Overall completeness

- Decent overall completeness (95%)
- Poor low-resolution completeness
- Poor low-resolution completeness is revealed by using log-based binning resolution range as described in Afonine *et al* (2013)
- Poor low-resolution completeness distorts maps

Effect of missing data: example

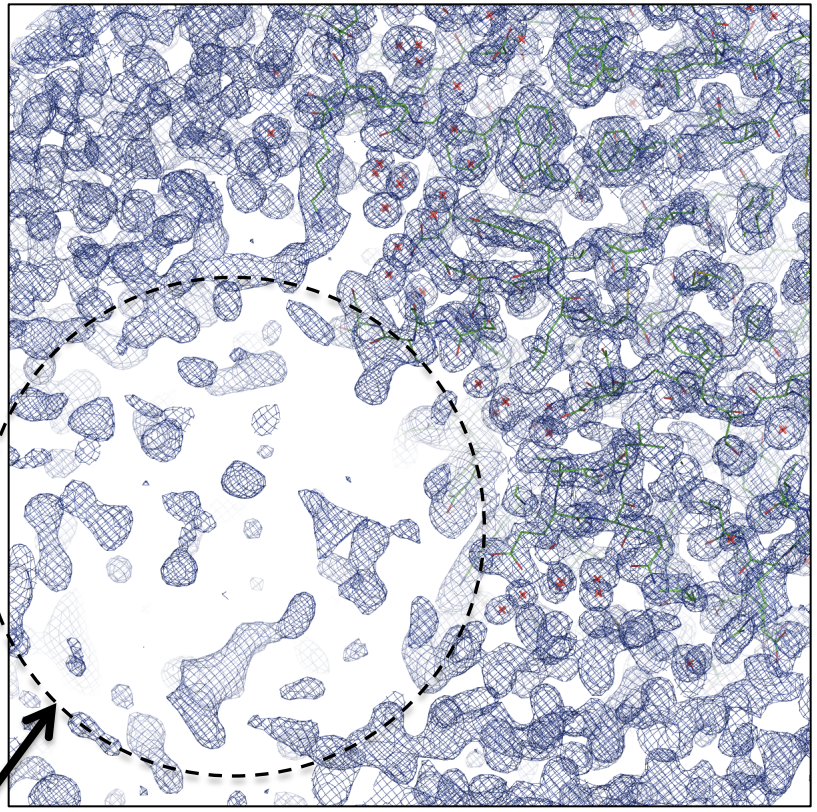
- PDB code: 1NH2, 1.9Å, **Fcalc** syntheses:

All 100% reflections in [1.9, ∞) Å



Contoured at 1σ

Reflections corresponding to measured Fobs (95% complete)



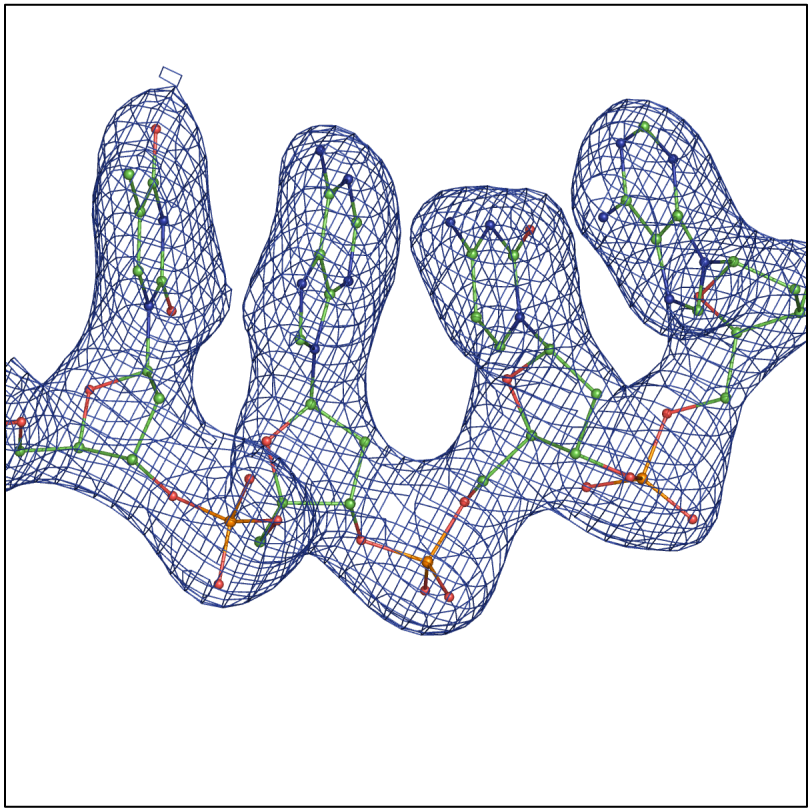
Contoured at 0.82σ

Pronounced map artifacts are solely due to incomplete data

Effect of missing data: example

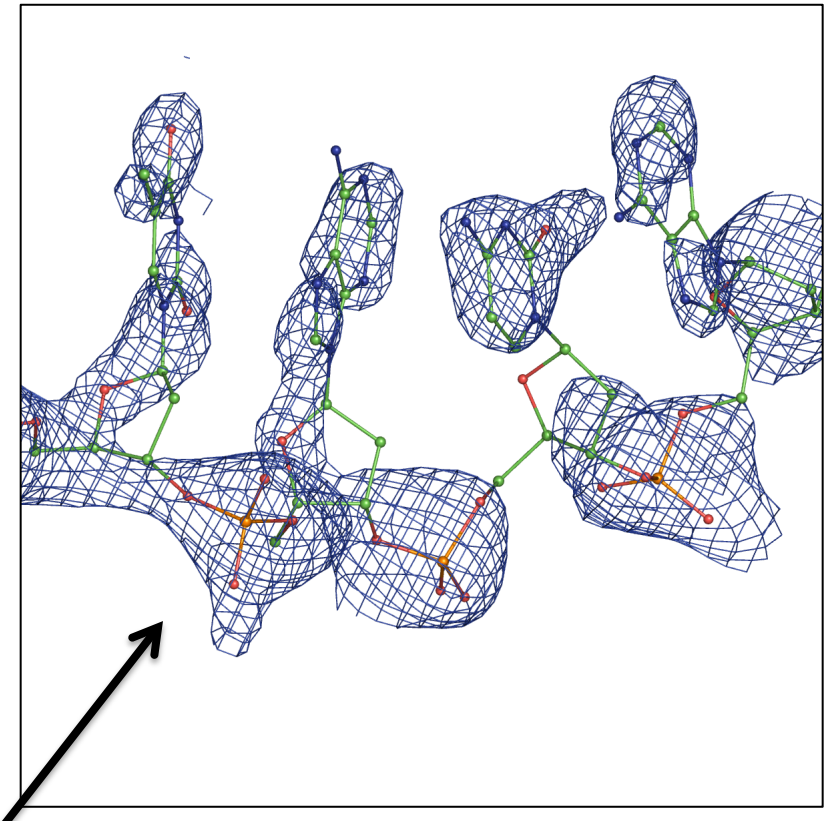
- PDB code: 1NH2, 1.9Å; **Fcalc** syntheses shown for F13-16

All 100% reflections in [1.9, ∞) Å



Contoured at 1σ

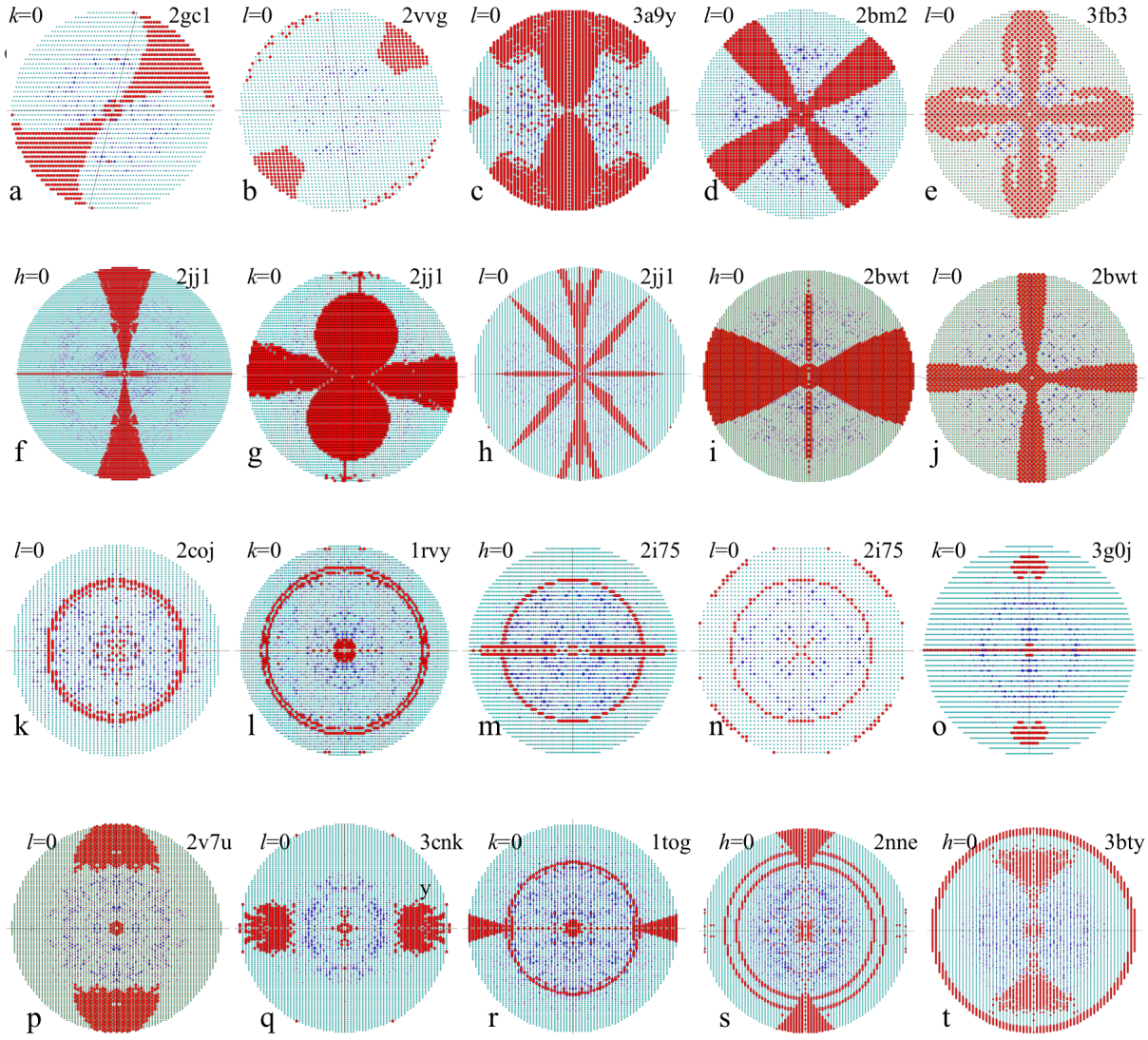
Reflections corresponding to measured Fobs (95% complete)



Contoured at 0.82σ

Severe Map distortions are solely due to incomplete data

Missing Fobs: extreme oddities or what you can find in PDB

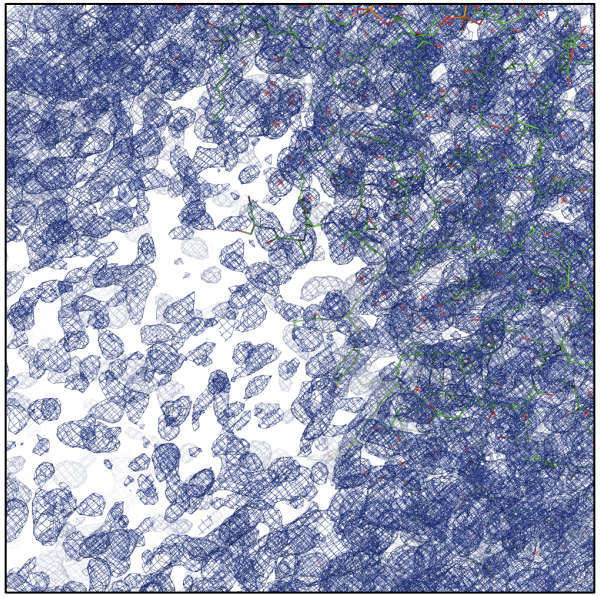


- Automatically detected and displayed using: **A program to analyze the distributions of unmeasured reflections. Urzhumtseva & Urzhumtsev.(2011). J.Appl.Cryst. 44, 865-872**
- There is a *Phenix* (GUI) tool to show this too (Nat Echols work)

Accounting for missing reflections

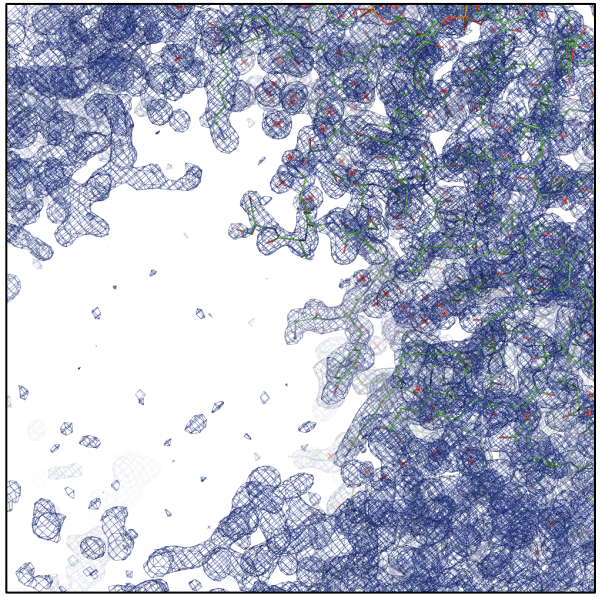
- PDB code: 1NH2, 1.9Å; **($2mF_{OBS} - DF_{MODEL}$, ϕ_{MODEL})** syntheses

F_{fill} = 0



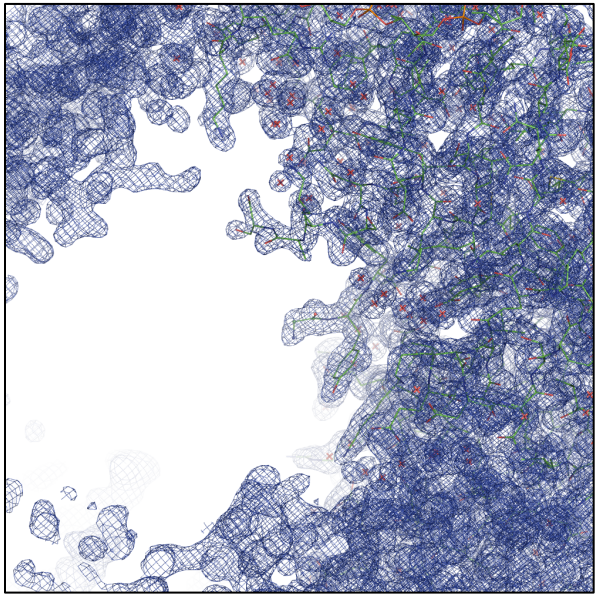
Contoured at 1σ

F_{fill} using *Resolve*



Contoured at 1.1σ

F_{fill} using F_{model}



Contoured at 1σ

Composite Residual OMIT map

1. OMIT map filter
- Compute composite residual OMIT map: M_{cromit}
- Scale M_{cromit} by standard deviation (σ)
- Compute filter: $M_{filter}=0$ if $M_{cromit}<0.5\sigma$ else $M_{filter}=1$

2. Initialize collector of integer maps, IMC

3. For j in j=1,16:

a. Map randomization and averaging
For i, i=1,10:
- Compute 100 map coefficients and average them: $MC_{average}$
- Randomly remove 5% of terms from $MC_{average}$: $\widetilde{MC}_{average}$
- Compute Fourier map M_i from $\widetilde{MC}_{average}$
- Scale M_i by standard deviation
- Truncate low values: set $M_i=0$ if $M_i<0.5\sigma$
- Eliminate regions in M_i with small volume

b. Sharpen M_i , M_{sharp}

c. Histogram equalize M_{sharp} , M_{HE}

d. Filter M_{sharp} by OMIT map: $M_{filtered} = M_{HE} * M_{filter}$

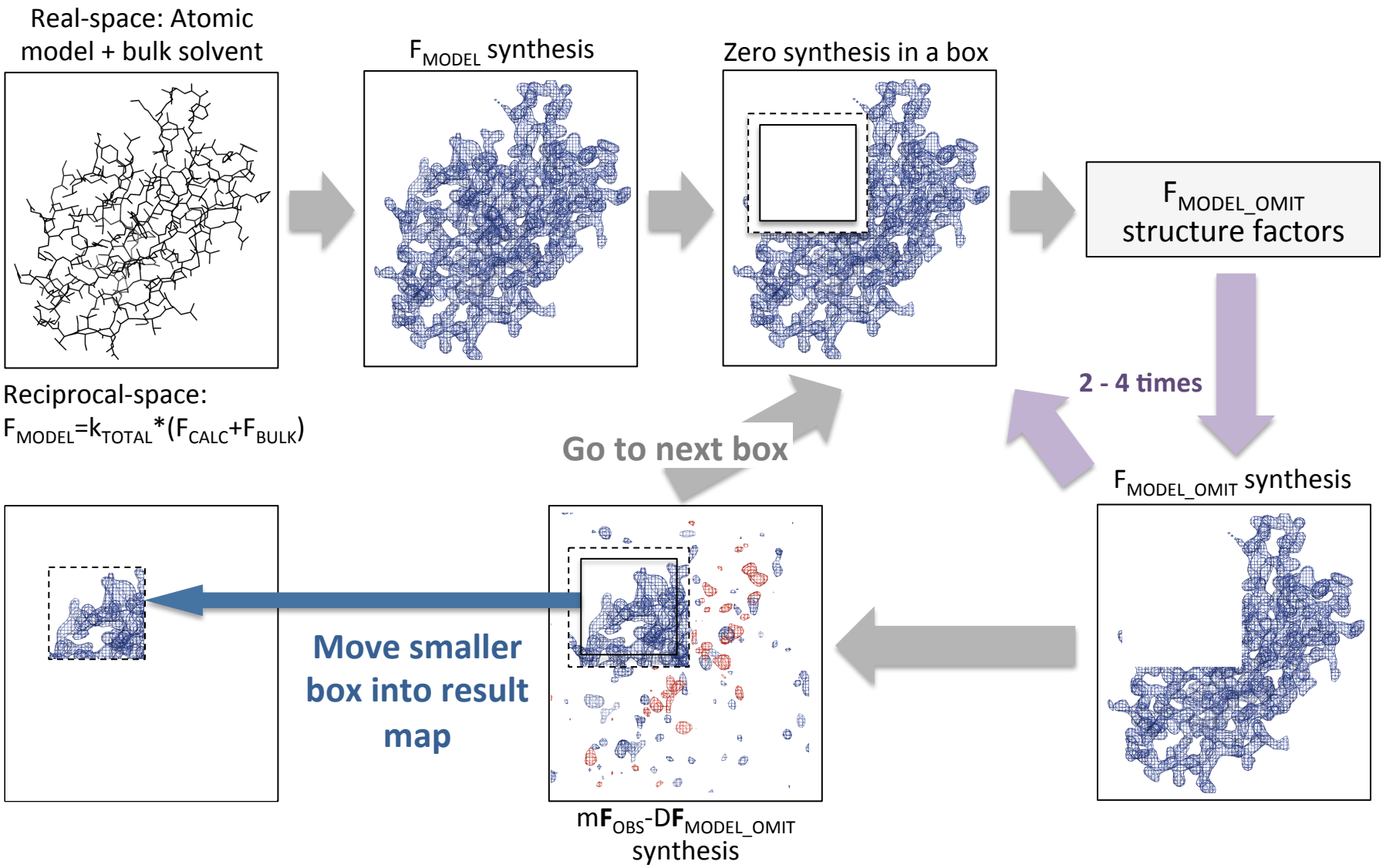
e. Add $M_{filtered}$ to IMC

4. Compute median map M_m from 16 maps in IMC, which is resulting Feature Enhanced Map, $M_{FEM} = M_m$

Composite Residual OMIT map: facts

- Fast to calculate: from tens of seconds to a few minutes
- Original implementation of OMIT map: Bhat & Cohen, 1984; Bhat, 1988
- Omit the map, not the model: Cowtan, 2012
- Original (alternative) implementation in *Phenix* by Tom Terwilliger
- ASU maps: Grosse-Kunstleve, Mustyakimov et al, 2011
  - CCTBX implementation by Marat Mustyakimov
- Available in *Phenix*: phenix.composite_omit_map
  - Command-line tool by Nat Echols
- This will be published as part of FEM related work

Composite Residual OMIT map as implemented as part of FEM

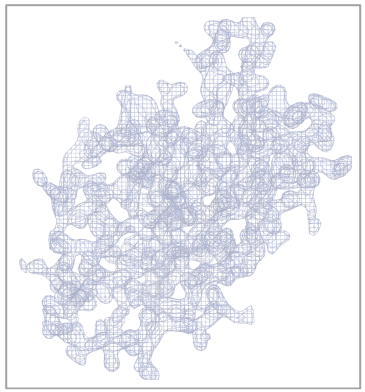


Composite difference OMIT map

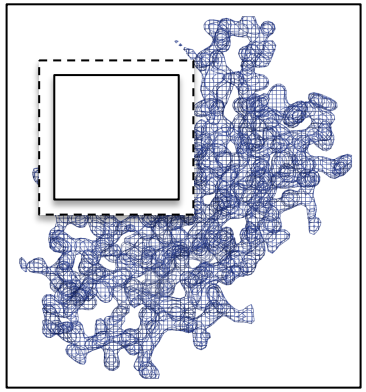
Real-space: Atomic model + bulk solvent

Reciprocal-space: $F_{MODEL}=k_{TOTAL}*(F_{CALC}+F_{BULK})$

F_{MODEL} synthesis



Zero synthesis in a box

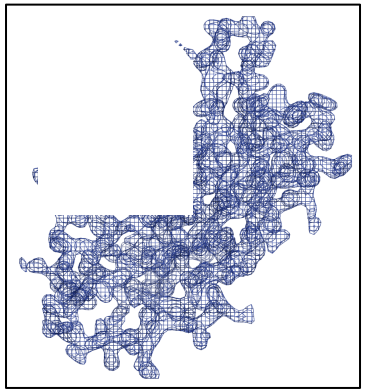


F_{MODEL_OMIT} structure factors

2 - 4 times

Go to next box

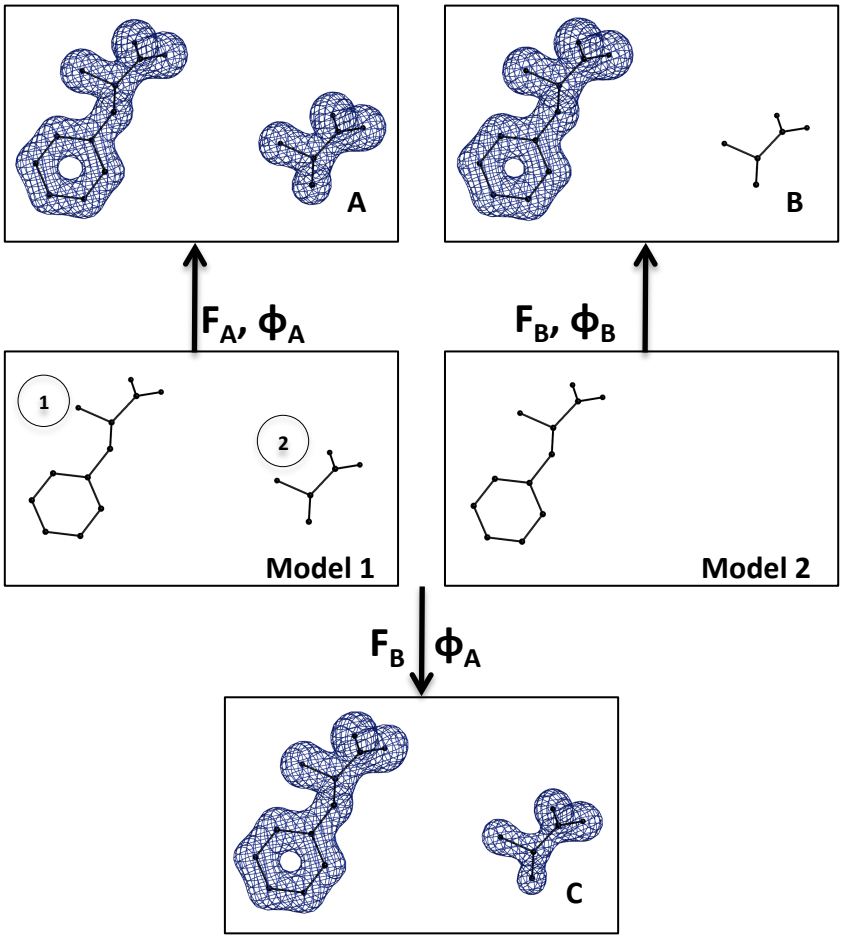
F_{MODEL_OMIT} synthesis



- Each atom and bulk-solvent contribute to each $\mathbf{F}_{MODEL}$
  - Each pixel of $\mathbf{F}_{MODEL}$ map contains contribution from each atom
    - Omitting a box from $\mathbf{F}_{MODEL}$ synthesis does not entirely omits scattering contribution arising from atoms and bulk-solvent contained in that box.
      - Recycling zeroing in the box 2-4 times typically flattens the map in it

Composite difference OMIT map: example/illustration

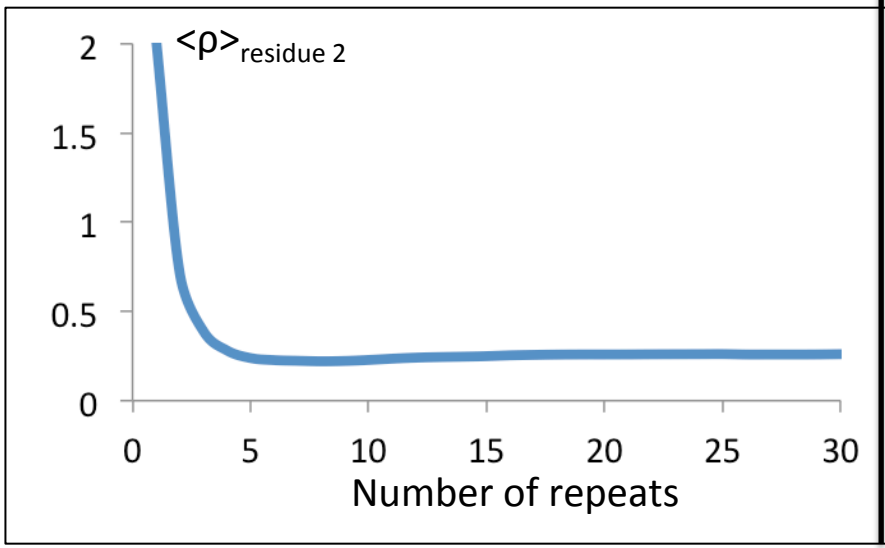
Numerical test setup to illustrate performance of OMIT map procedure



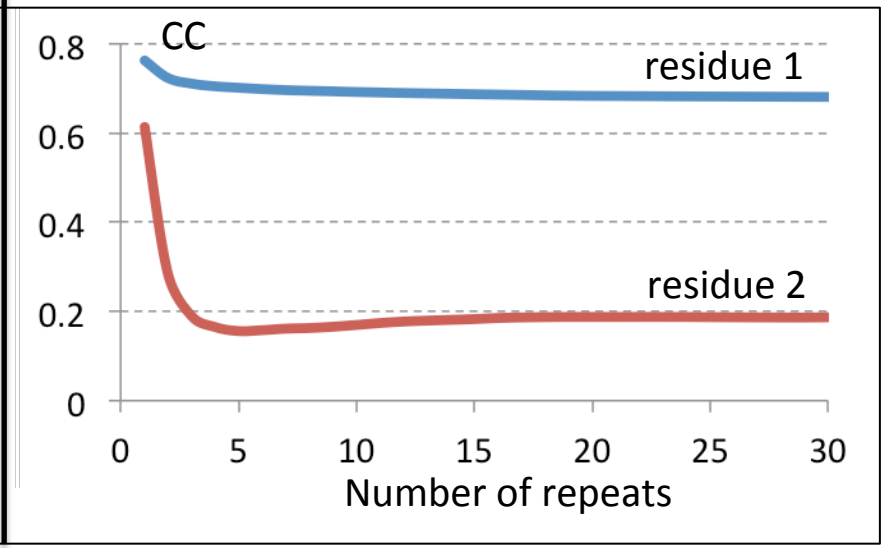
- Model 1: two residues, 1 and 2, placed in P1 box, and it is used to calculate (F_A, ϕ_A) synthesis, A.
- Model 2: one residue, 1, (otherwise identical to model 1) and it is used to calculate (F_B, ϕ_B) synthesis, B.
- Amplitudes F_B and phases ϕ_A are used to compute synthesis C.
- All syntheses are contoured at 3σ. Positive map around residue 1 in C is purely the model bias.

Composite difference OMIT map: example/illustration

Mean OMIT map values calculated at atomic centers of residue 2 as a function of the number of recycling.



CC(OMIT map, model map) for residues 1 (blue) and 2 (red) as a function of the number of recycling.



- Two to four repeats are sufficient to eliminate model bias in this test setup

Map randomization and averaging

1. OMIT map filter
- Compute composite residual OMIT map: M_{cromit}
- Scale M_{cromit} by standard deviation (σ)
- Compute filter: $M_{filter}=0$ if $M_{cromit}<0.5\sigma$ else $M_{filter}=1$

2. Initialize collector of integer maps, IMC

3. For j in j=1,16:

a. Map randomization and averaging

For i, i=1,10:
- Compute 100 map coefficients and average them: $MC_{average}$
- Randomly remove 5% of terms from $MC_{average}$: $\widetilde{MC}_{average}$
- Compute Fourier map M_i from $\widetilde{MC}_{average}$
- Scale M_i by standard deviation
- Truncate low values: set $M_i=0$ if $M_i<0.5\sigma$
- Eliminate regions in M_i with small volume

b. Sharpen M_i , M_{sharp}

c. Histogram equalize M_{sharp} , M_{HE}

d. Filter M_{sharp} by OMIT map: $M_{filtered} = M_{HE} * M_{filter}$

e. Add $M_{filtered}$ to IMC

4. Compute median map M_m from 16 maps in IMC, which is resulting Feature Enhanced Map, $M_{FEM} = M_m$

Map randomization and averaging: tools

- **Map coefficients**

$$\mathbf{F}_{\mathrm{map}} = \begin{cases} \frac{w}{k_{\mathrm{total}}}(2mF_{\mathrm{obs}} - DF_{\mathrm{model}}), \varphi_{\mathrm{model}} & \text{acentric refletions} \\ \frac{w}{k_{\mathrm{total}}} mF_{\mathrm{obs}}, \varphi_{\mathrm{model}} & \text{centric reflections} \\ F_{\mathrm{fill}}, \varphi_{\mathrm{fill}} & \text{no } F_{\mathrm{obs}} \text{ in } [d_{\mathrm{min}}, \infty) \end{cases}$$

- **Shelx weighting**

$$w = \left(1 + \frac{\sigma^2_{I_{\mathrm{obs}}}}{I^2_{\mathrm{model}}} + \delta^2 \frac{I^2_{\mathrm{obs}}}{I^2_{\mathrm{model}}}\right)^{-1}$$

δ - some small value

  - No official publication that explains the rationale
  - Only mentioned in SHELXL documentation:
    - SHELX homepage -> Wikis and manuals -> SHELXL command list (HTML) -> FMAP

- **By choosing slightly different values of δ one can obtain as many slightly randomized maps as desired**

Map randomization and averaging: tools

- **Problem with Shelx weighting**

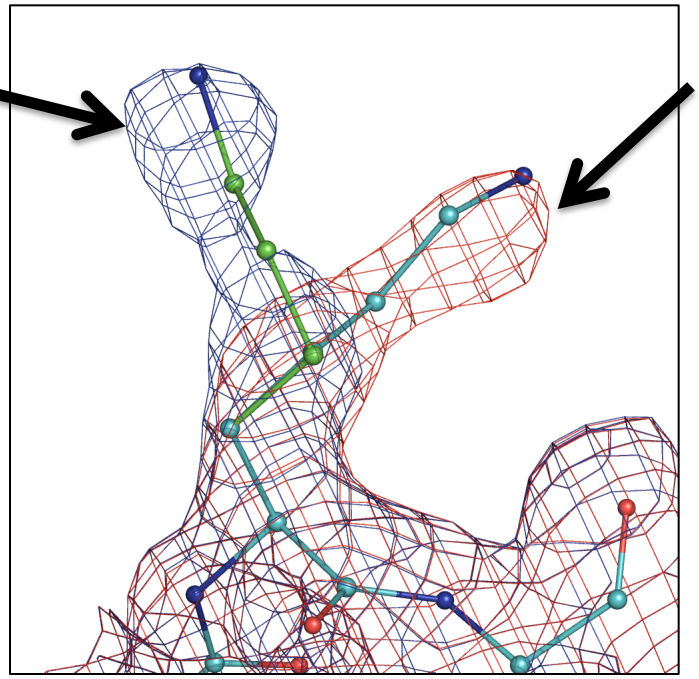
$$w = \left(1 + \frac{\sigma^2_{I_{obs}}}{I^2_{model}} + \delta^2 \frac{I^2_{obs}}{I^2_{model}}\right)^{-1}$$

δ - some small value

  - Some δ value may result in severely model biased maps

- **Example**: residue L107 in 1F8T structure, maps shown at 1σ

Correct map (blue) calculated with w=1



Residue L107 switched to an incorrect rotamer. Red map calculated using δ=1.5

Map randomization and averaging: tools

- **Map coefficients**

$$\mathbf{F}_{\mathrm{map}} = \begin{cases} \frac{w}{k_{\mathrm{total}}}(2mF_{\mathrm{obs}} - DF_{\mathrm{model}}), \varphi_{\mathrm{model}} & \text{acentric refletions} \\ \frac{w}{k_{\mathrm{total}}} mF_{\mathrm{obs}}, \varphi_{\mathrm{model}} & \text{centric reflections} \\ F_{\mathrm{fill}}, \varphi_{\mathrm{fill}} & \text{no } F_{\mathrm{obs}} \text{ in } [d_{\mathrm{min}}, \infty) \end{cases}$$

- **Alternative weighting (introduced as part of FEM development)**

$$w = \left(1 + \alpha \sum \frac{|I_{\mathrm{obs}} - I_{\mathrm{model}}|}{|I_{\mathrm{obs}} + I_{\mathrm{model}}|} + \beta \frac{\sigma_{I_{\mathrm{obs}}}}{I_{\mathrm{obs}}}\right)^{-1}$$

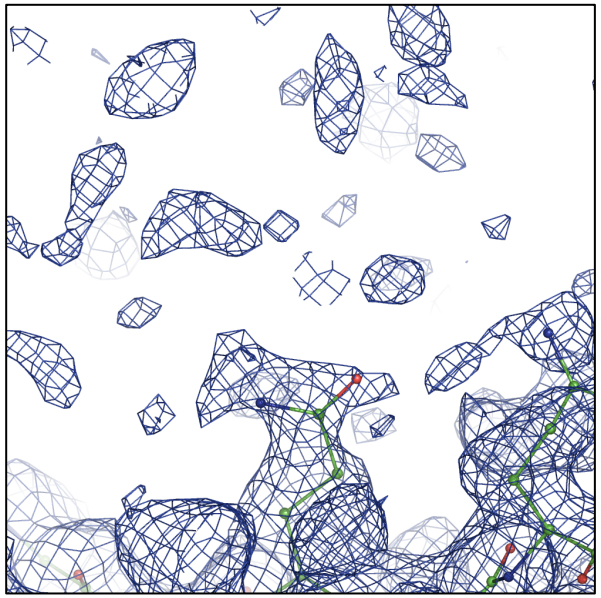
α and β – some numbers

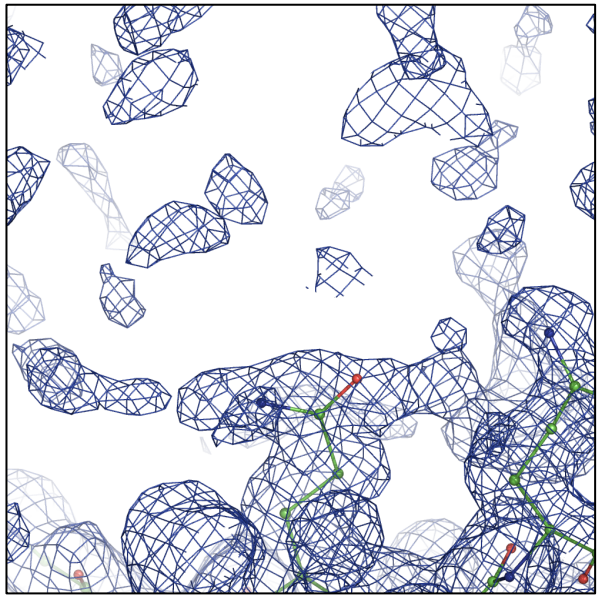
- **By defining α and β individually for each reflection one can obtain as many slightly randomized maps as desired**
  - α and β picked from (0,5) range at random

Map randomization and averaging: example

- Map coefficients (all maps shown at equivalent cutoffs ~0.5σ)

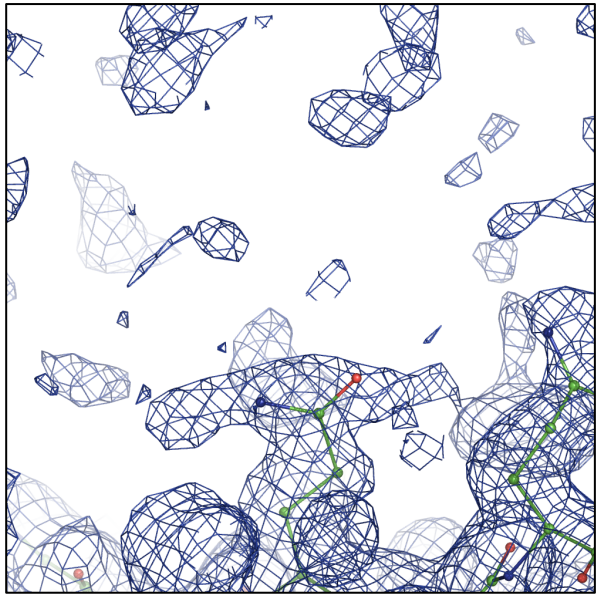
w=1

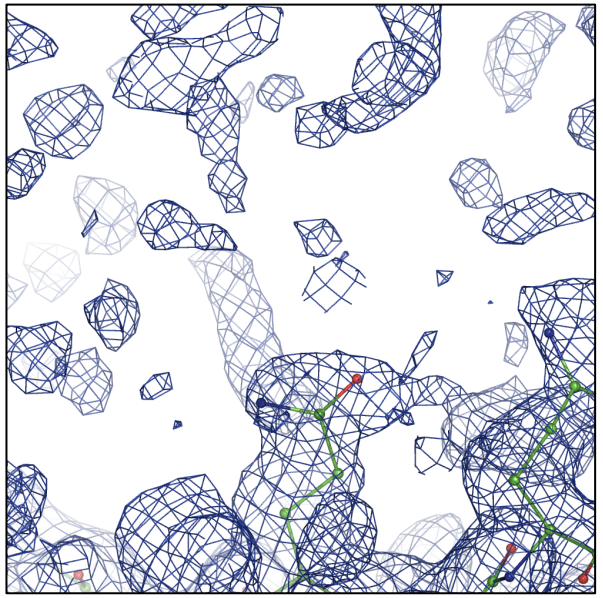




w≠1
Random seed=
2679941

w≠1
Random seed=
1312425





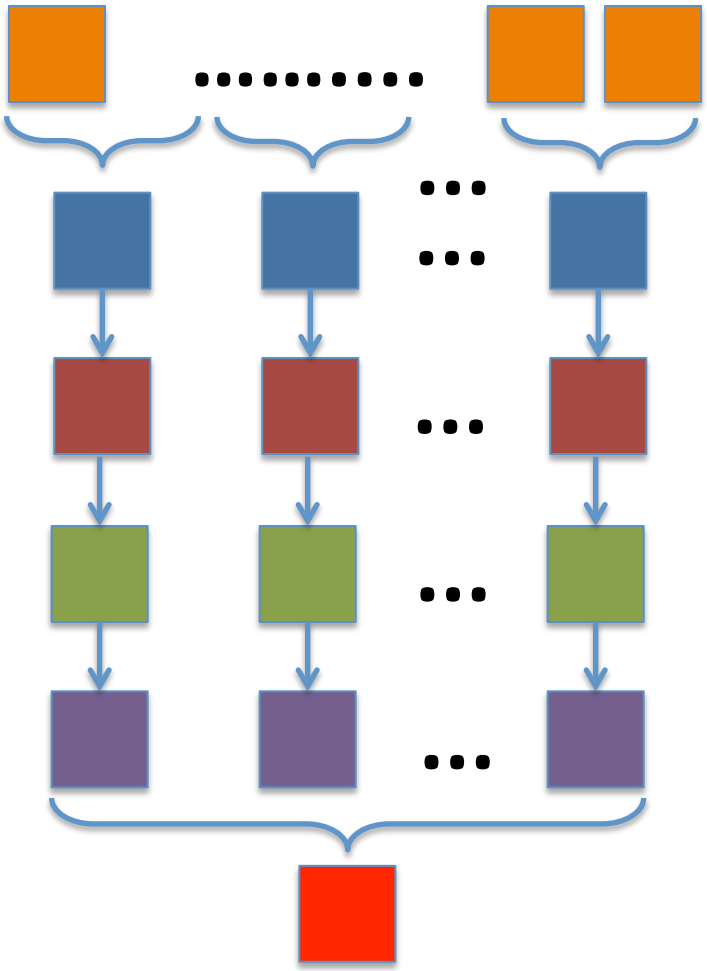
w≠1
Random seed=
1312425
5% reflections
removed

Map randomization and averaging: schematic illustration of workflow

$$\mathbf{F}_{map} = \begin{cases} \frac{w}{k_{total}}(2mF_{obs} - DF_{model}), \varphi_{model} & \text{acentric refletions} \\ \frac{w}{k_{total}} mF_{obs}, \varphi_{model} & \text{centric reflections} \\ F_{fill}, \varphi_{fill} & \text{no } F_{obs} \text{ in } [d_{min}, \infty) \end{cases}$$

$$w = \left(1 + \alpha \sum \frac{|I_{obs} - I_{model}|}{|I_{obs} + I_{model}|} + \beta \frac{\sigma_{I_{obs}}}{I_{obs}}\right)^{-1}$$

α and β ~ 0…5



1000 randomized map coefficients

Average 100 map coefficients

10 sets of averaged map coefficients

Randomly remove 5% of map coefficients

FT (Compute Fourier maps) and scale by s.d.

Eliminate low-value and low-volume regions

Average 10 maps to yield **one averaged map**

Low density elimination

1. OMIT map filter
- Compute composite residual OMIT map: M_{cromit}
- Scale M_{cromit} by standard deviation (σ)
- Compute filter: $M_{filter}=0$ if $M_{cromit}<0.5\sigma$ else $M_{filter}=1$

2. Initialize collector of integer maps, IMC

3. For j in j=1,16:

a. Map randomization and averaging
For i, i=1,10:
- Compute 100 map coefficients and average them: $MC_{average}$
- Randomly remove 5% of terms from $MC_{average}$: $\widetilde{MC}_{average}$
- Compute Fourier map M_i from $\widetilde{MC}_{average}$
- Scale M_i by standard deviation
- Truncate low values: set $M_i=0$ if $M_i<0.5\sigma$
- Eliminate regions in M_i with small volume

b. Sharpen M_i , M_{sharp}

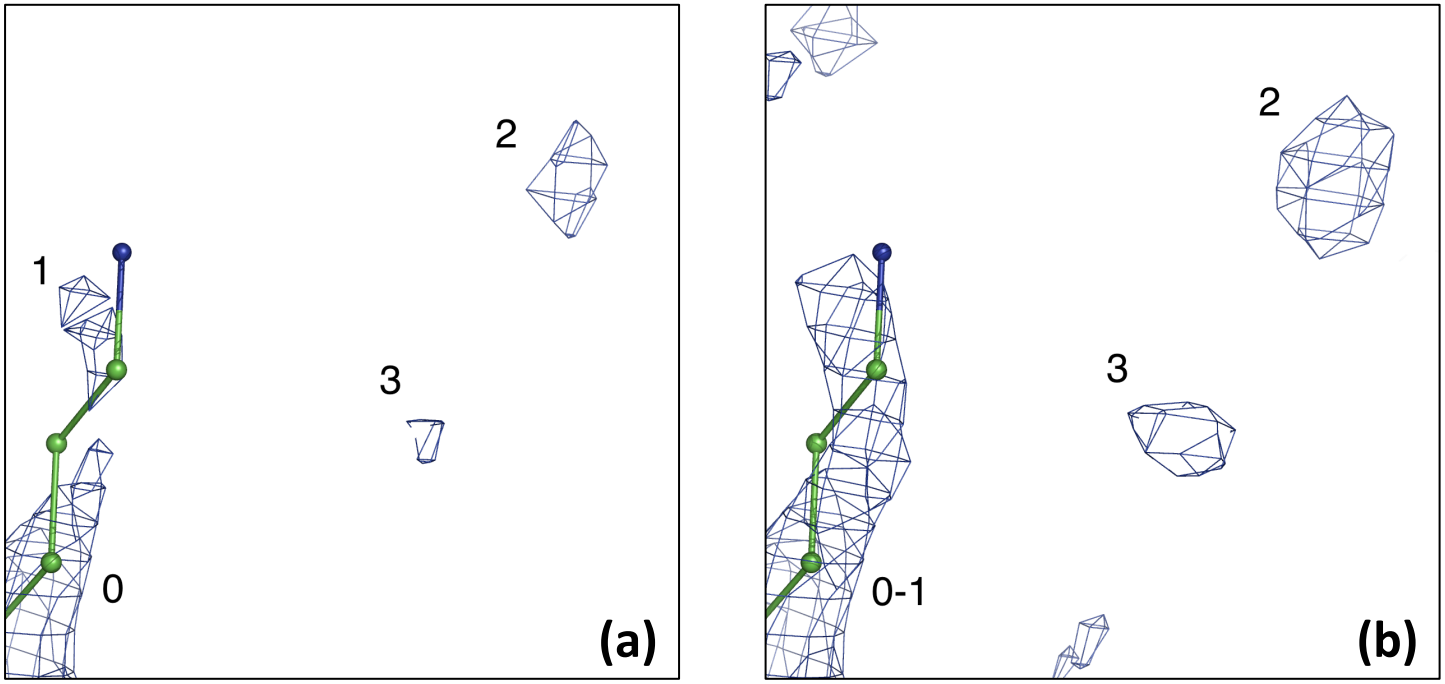
c. Histogram equalize M_{sharp} , M_{HE}

d. Filter M_{sharp} by OMIT map: $M_{filtered} = M_{HE} * M_{filter}$

e. Add $M_{filtered}$ to IMC

4. Compute median map M_m from 16 maps in IMC, which is resulting Feature Enhanced Map, $M_{FEM} = M_m$

Low density elimination



- **Schematic illustration of noise peaks (small density droplets) elimination:**
  (a) map contoured at threshold level t1
    - blobs 1,2,3 are selected for elimination
    - whole blob 0 is retained

  (b) Map contoured at threshold level t1-δ.
    - blob 1 merges with blob 0, and therefore it is retained
    - bigger blobs 2 and 3 are removed
- **To ensure comprehensive filtering, the procedure is repeated through a range of different threshold levels t1**

Map sharpening

1. OMIT map filter
- Compute composite residual OMIT map: M_{cromit}
- Scale M_{cromit} by standard deviation (σ)
- Compute filter: $M_{filter}=0$ if $M_{cromit}<0.5\sigma$ else $M_{filter}=1$

2. Initialize collector of integer maps, IMC

3. For j in j=1,16:

a. Map randomization and averaging
For i, i=1,10:
- Compute 100 map coefficients and average them: $MC_{average}$
- Randomly remove 5% of terms from $MC_{average}$: $\widetilde{MC}_{average}$
- Compute Fourier map M_i from $\widetilde{MC}_{average}$
- Scale M_i by standard deviation
- Truncate low values: set $M_i=0$ if $M_i<0.5\sigma$
- Eliminate regions in M_i with small volume

b. Sharpen M_i , M_{sharp}

c. Histogram equalize M_{sharp} , M_{HE}

d. Filter M_{sharp} by OMIT map: $M_{filtered} = M_{HE} * M_{filter}$

e. Add $M_{filtered}$ to IMC

4. Compute median map M_m from 16 maps in IMC, which is resulting Feature Enhanced Map, $M_{FEM} = M_m$

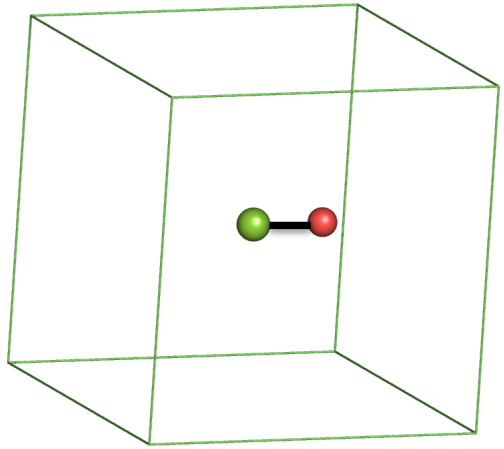
Map sharpening

- **Map randomization blurs map peaks – sharpening is needed**
  - **FEM uses two very different methods:**
    - **Exponential sharpening ("B-factor map sharpening"):**
      - Find optimal sharpening B-factor
        - Use map kurtosis as a criterion for finding optimal sharpening B
    - **Unsharp mask:**
      - New map = max(*Original Map* - *Averaged Map*, 0)

Map sharpening: see how map kurtosis may be useful

Toy example:

Mg-O in 10*10*10Å P1 box

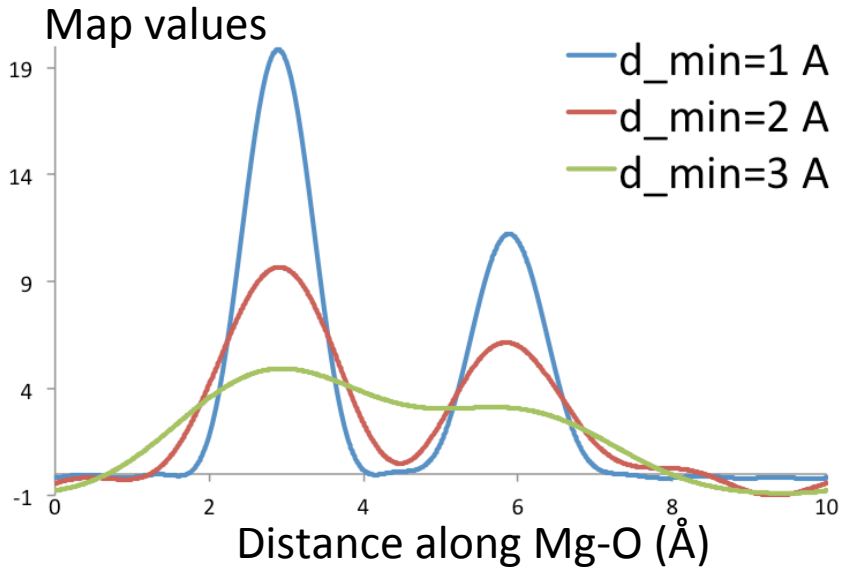


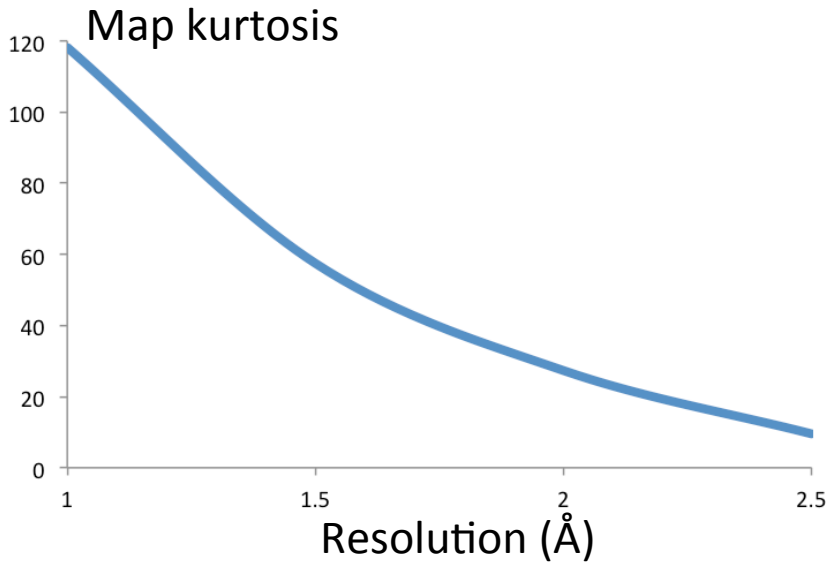
- Plot Fourier map distribution along Mg-O bond vector, and see how it varies as function of:
  - **Resolution**
  - **B-factor**

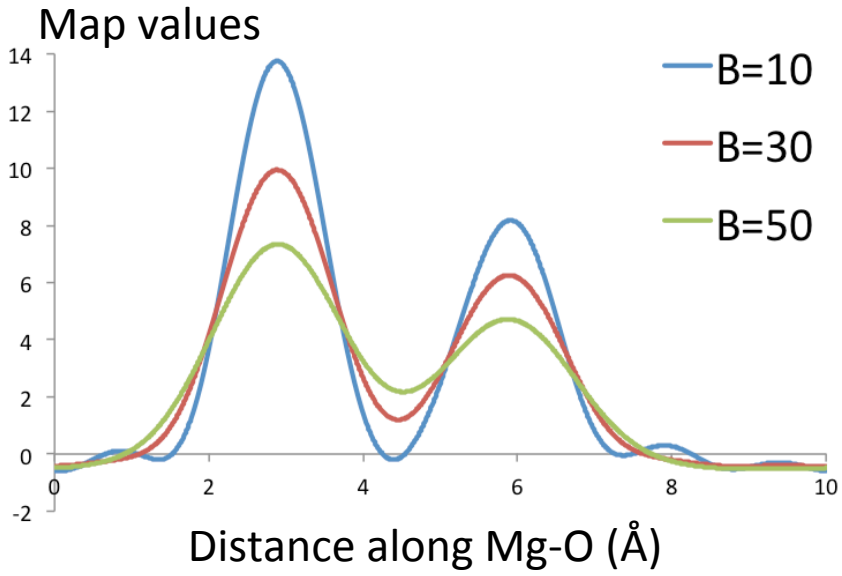
Also compute *kurtosis* of map distribution for each trial map: $$kurtosis = \frac{\sum (x_i - \bar{x})^4}{N\sigma^4}$$

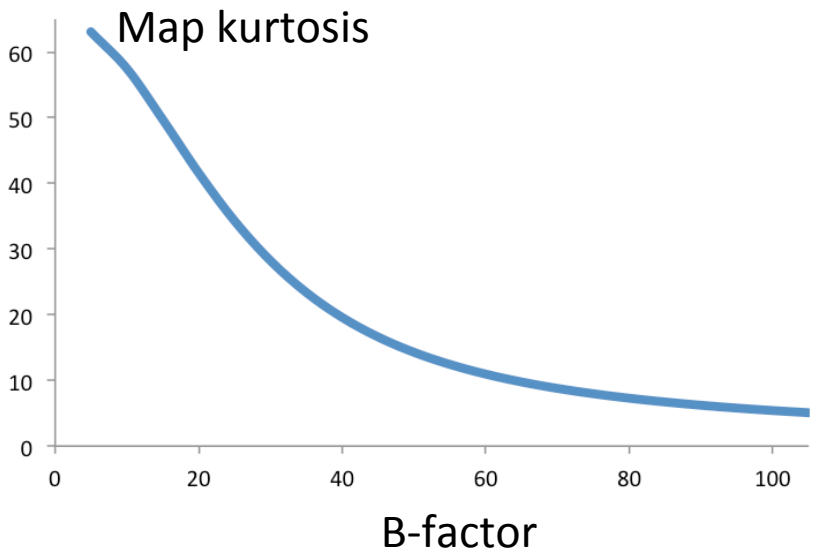
- **Goal**: see if map kurtosis correlates with map sharpness change due to varying resolution and B-factors

Map sharpening: see how map kurtosis may be useful



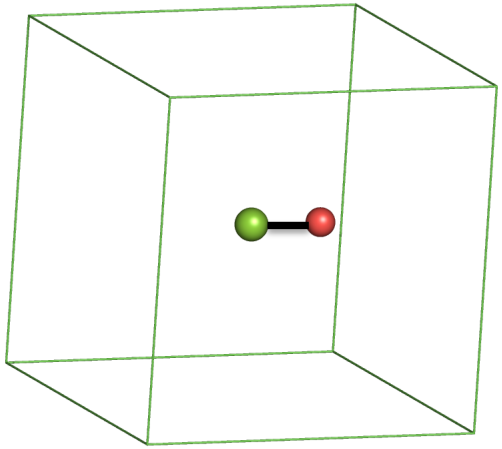






Map sharpening: see how map kurtosis may be useful

Toy example: Mg-O in 10*10*10Å P1 box

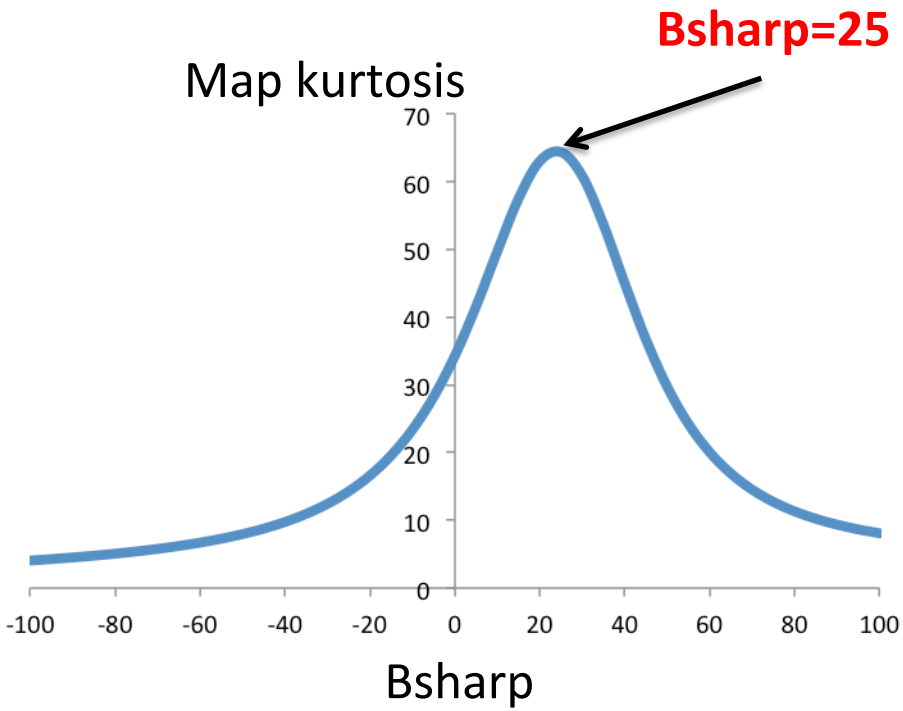


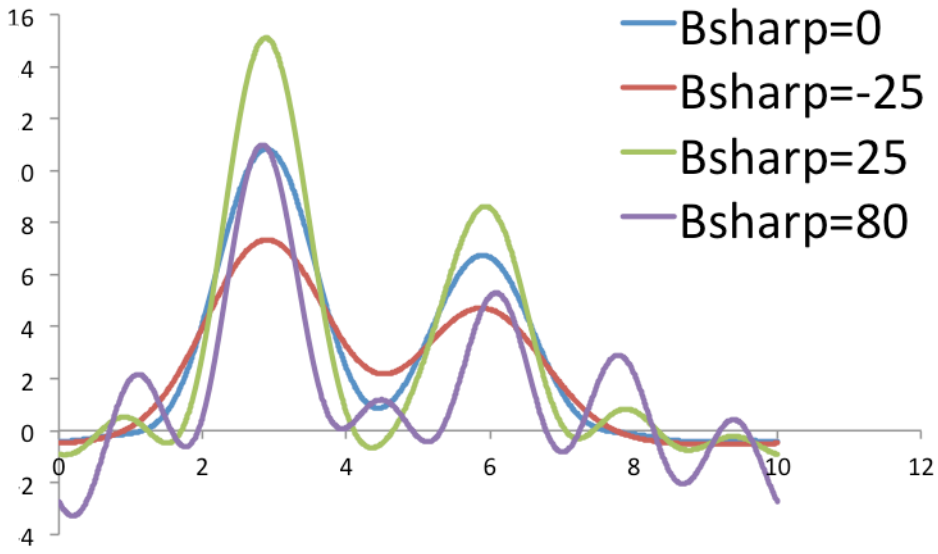
- Set **B=25** to both atoms

  … and see how map kurtosis depends on sharpening B-factor: Bsharp

Map sharpening: see how map kurtosis may be useful

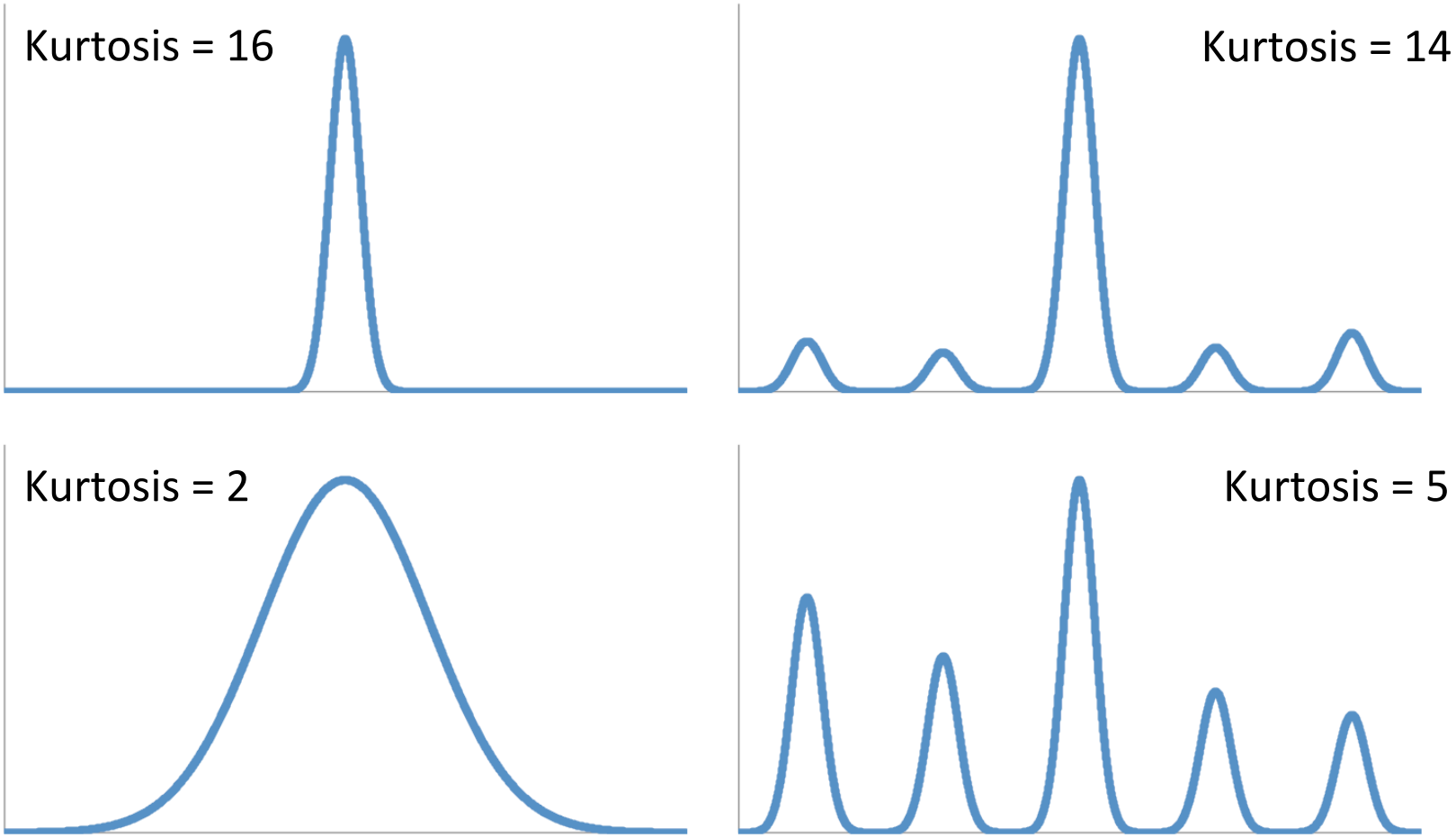
- Sample Bsharp in [-100, 100]:





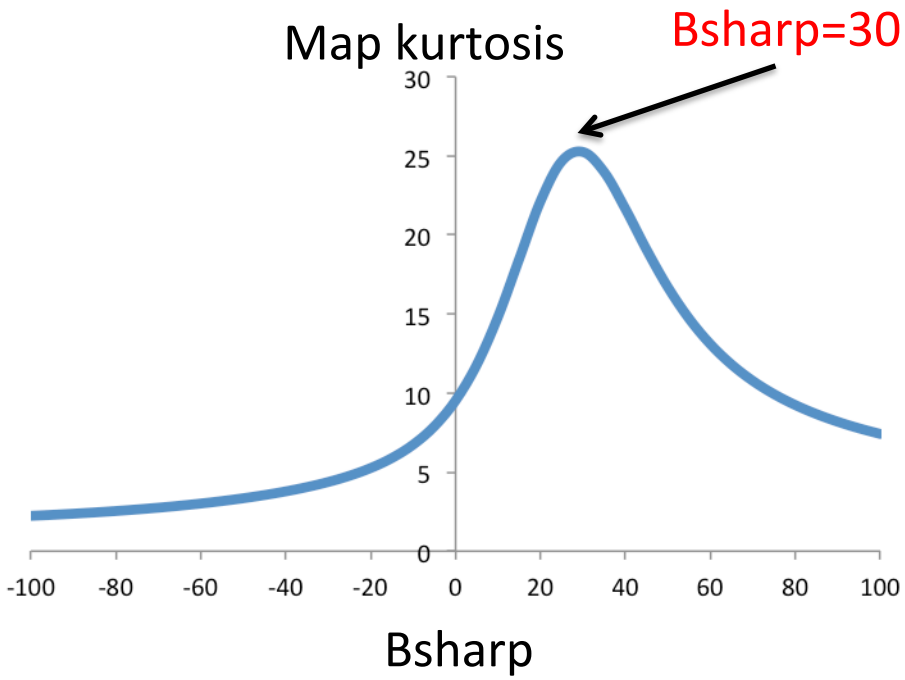
- **Best Bsharp** maximizes kurtosis and corresponds to overall B that reduces atomic B-factors to zero, which corresponds to sharpest peaks

Map sharpening: illustration of kurtosis



Map sharpening: see how map kurtosis may be useful

- Real structure: PDB code 1F8T
  - Mean B=29



Map kurtosis is a great criterions for choosing optimal sharpening B-factor!

Map sharpening

1. OMIT map filter
- Compute composite residual OMIT map: M_{cromit}
- Scale M_{cromit} by standard deviation (σ)
- Compute filter: $M_{filter}=0$ if $M_{cromit}<0.5\sigma$ else $M_{filter}=1$

2. Initialize collector of integer maps, IMC

3. For j in j=1,16:

a. Map randomization and averaging
For i, i=1,10:
- Compute 100 map coefficients and average them: $MC_{average}$
- Randomly remove 5% of terms from $MC_{average}$: $\widetilde{MC}_{average}$
- Compute Fourier map M_i from $\widetilde{MC}_{average}$
- Scale M_i by standard deviation
- Truncate low values: set $M_i=0$ if $M_i<0.5\sigma$
- Eliminate regions in M_i with small volume

b. Sharpen M_i , M_{sharp}

c. Histogram equalize M_{sharp} , M_{HE}

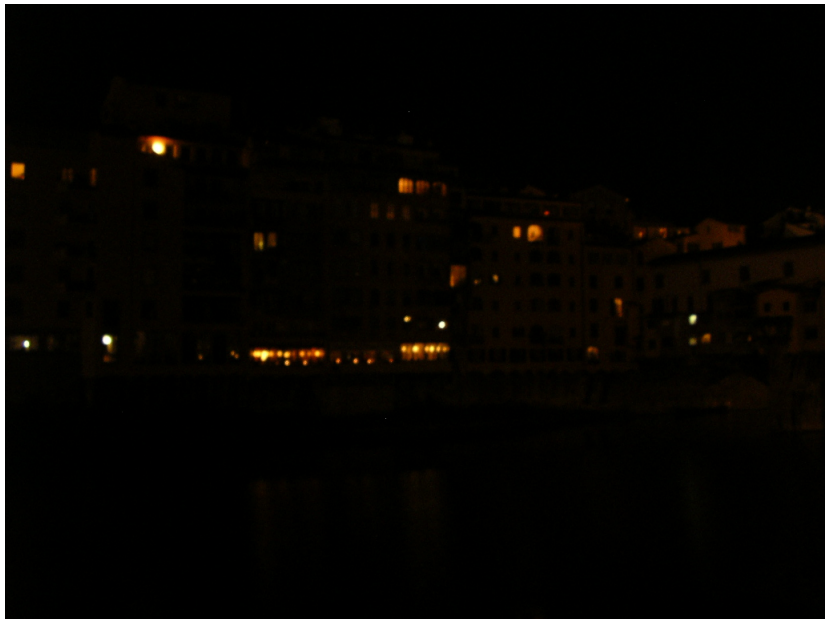
d. Filter M_{sharp} by OMIT map: $M_{filtered} = M_{HE} * M_{filter}$

e. Add $M_{filtered}$ to IMC

4. Compute median map M_m from 16 maps in IMC, which is resulting Feature Enhanced Map, $M_{FEM} = M_m$

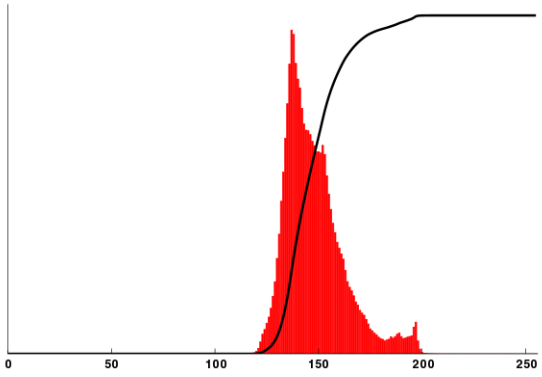
Histogram equalization (learning from digital image processing)

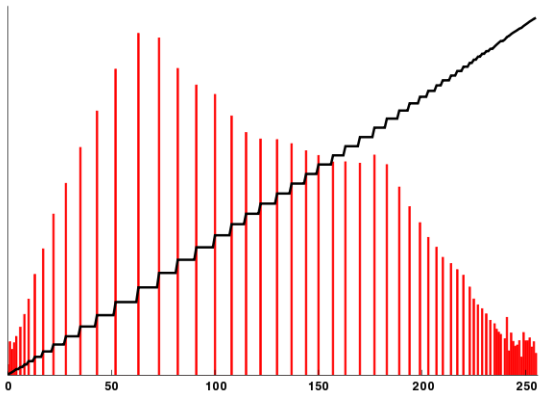
Unequalized histogram



HE

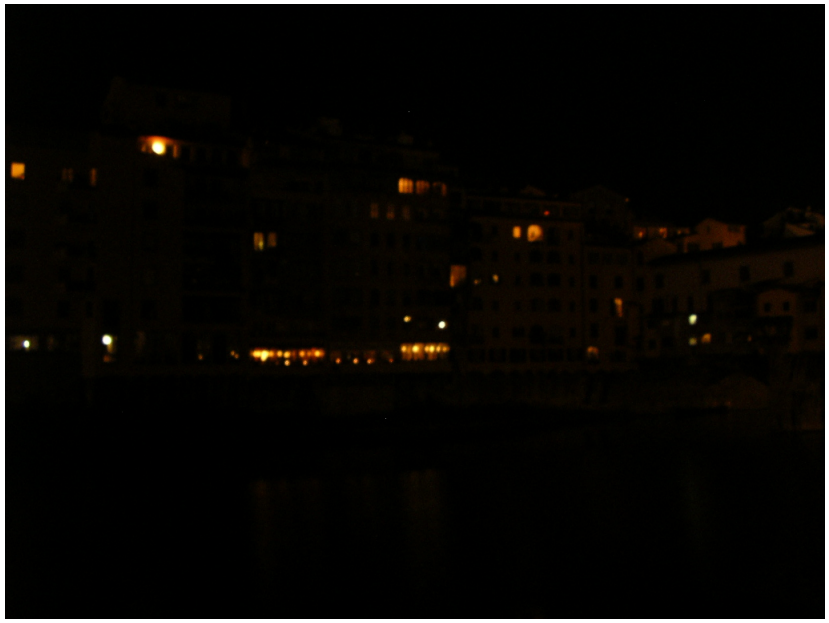
Equalized histogram





Histogram equalization (learning from digital image processing)

Unequalized histogram



HE

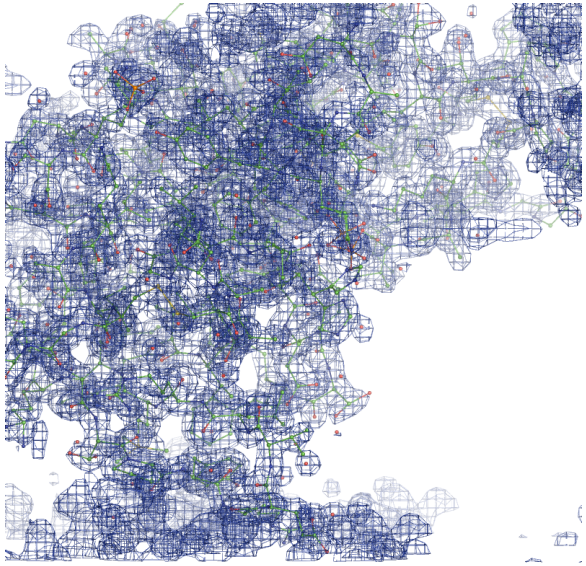
Equalized histogram

- Result of HE: an unrealistic image (image on the right misleadingly makes an impression of a brighter environment) with enhanced features and potentially enhanced noise

Histogram equalization: does it work for crystallographic maps?

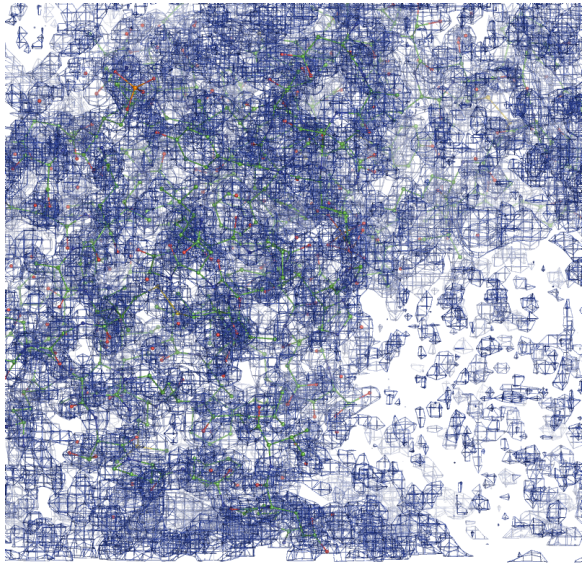
- 2mFo-DFc contoured at equivalent cutoffs corresponding approx. to 1.5σ

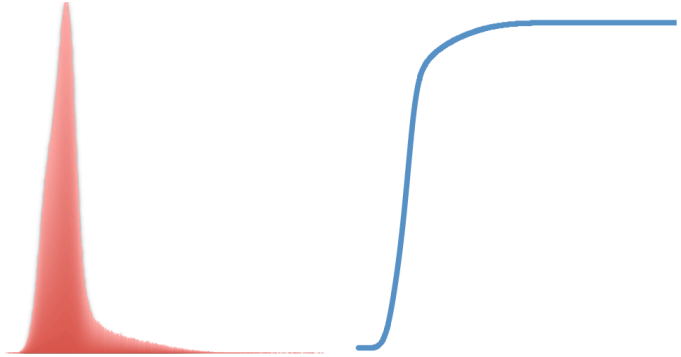
Before histogram equalization

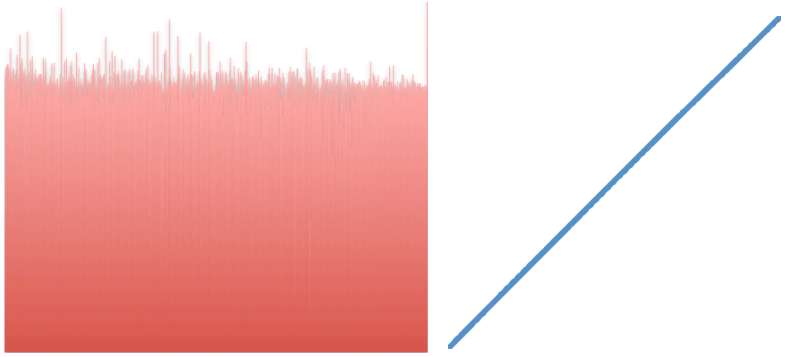


HE

After histogram equalization





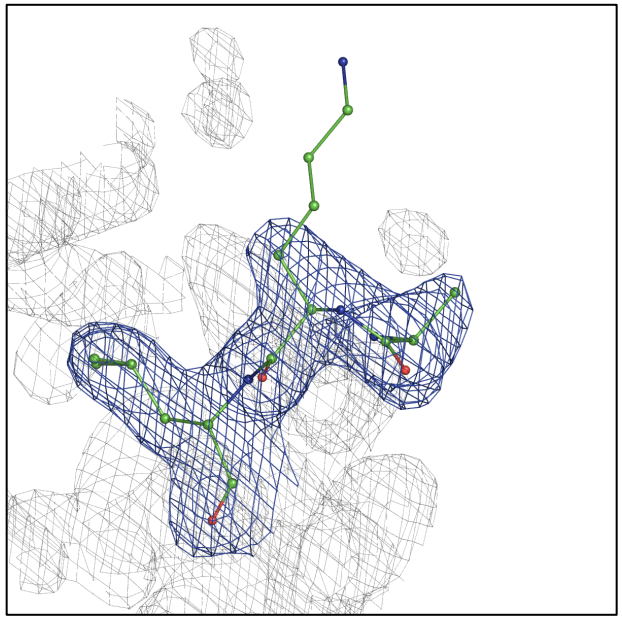


- Noise is enhanced as good as the signal
  - **It is critical to remove the noise before HE**

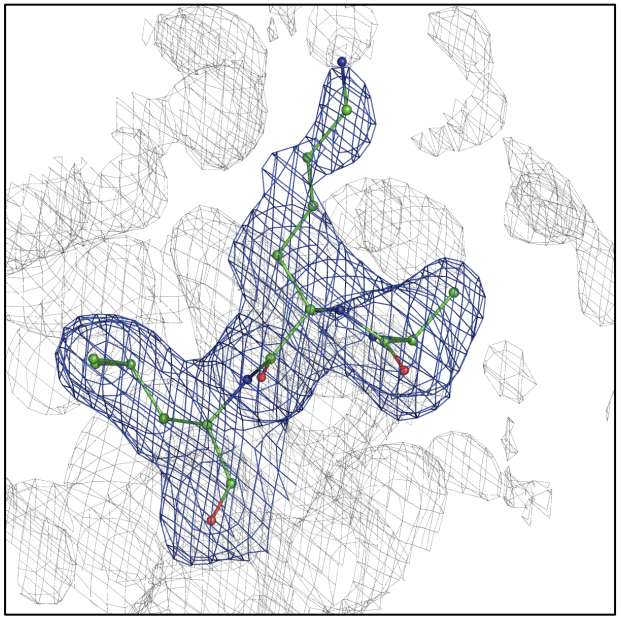
Histogram equalization (crystallographic maps)

- PDB code: 1SSW, residue A83

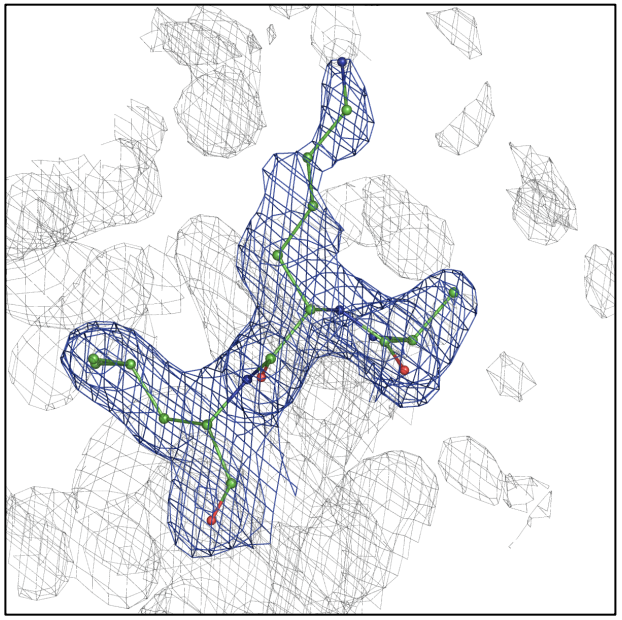
2mFo-DFc , 1σ



2mFo-DFc , 0.4σ



HE, equivalent 1σ

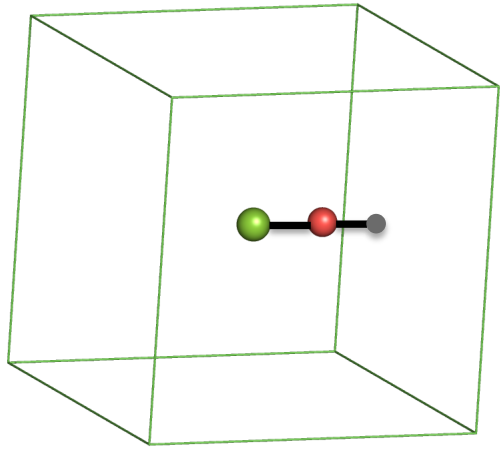


- **Blue: only density within 1.5Å radius around atoms is shown**
- **Grey: same as blue shown within 5Å radius around atoms**
- **Blue map is shown on top of grey map**

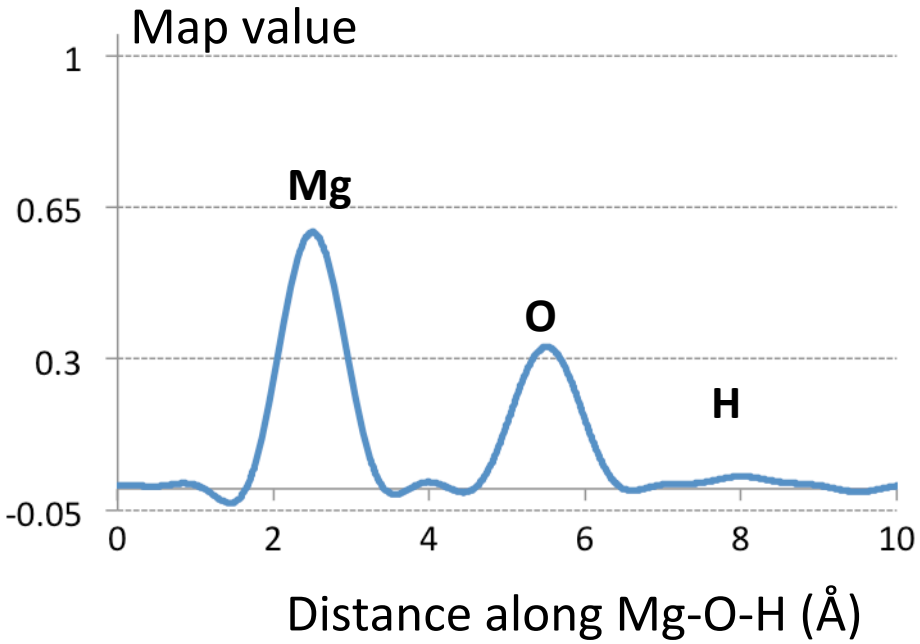
Histogram equalization: illustration

Toy example:

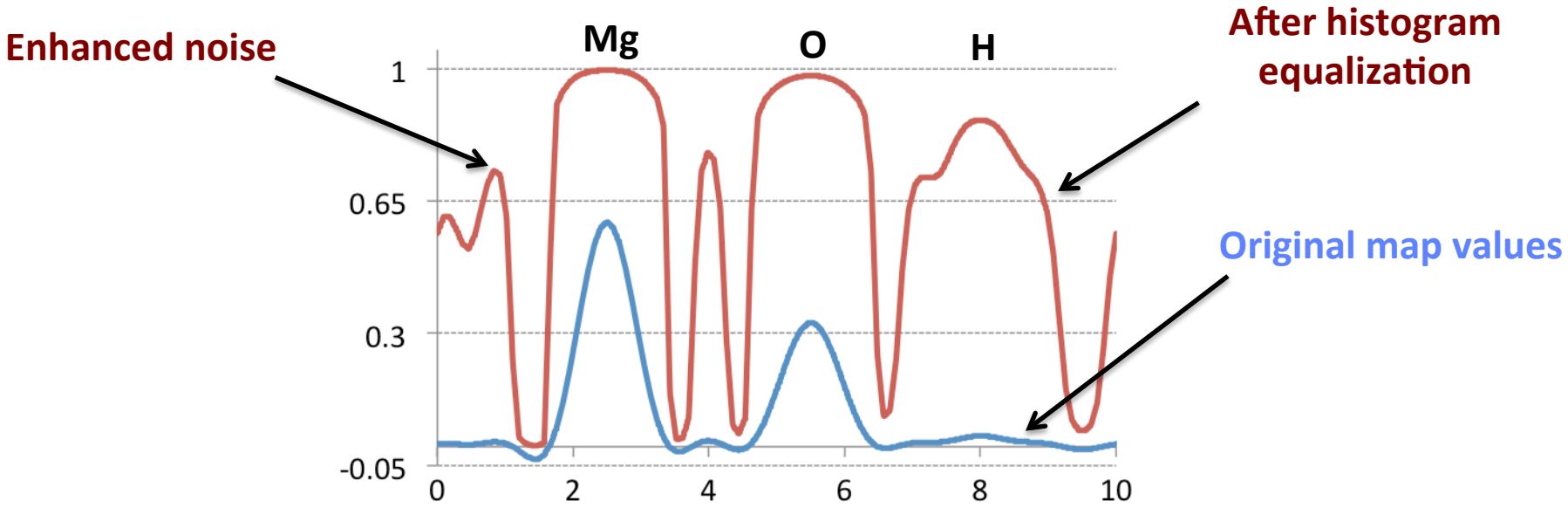
Mg-O-H in 10*10*10Å P1 box



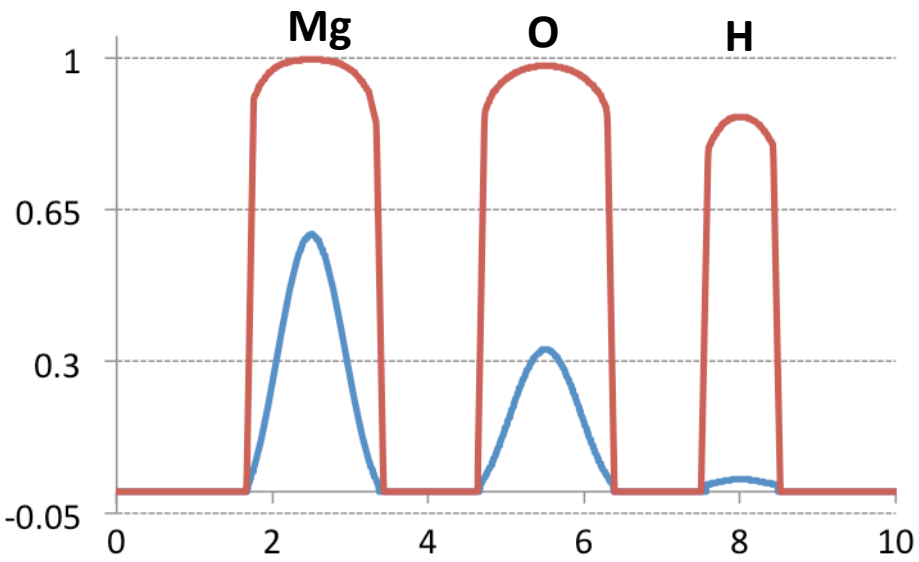
1Å resolution Fourier map along Mg-O-H bond vector



Histogram equalization: illustration



If noise is removed:



Storing many maps in memory and computing median map

1. OMIT map filter
- Compute composite residual OMIT map: M_{cromit}
- Scale M_{cromit} by standard deviation (σ)
- Compute filter: $M_{filter}=0$ if $M_{cromit}<0.5\sigma$ else $M_{filter}=1$

2. Initialize collector of integer maps, IMC

3. For j in j=1,16:

a. Map randomization and averaging
For i, i=1,10:
- Compute 100 map coefficients and average them: $MC_{average}$
- Randomly remove 5% of terms from $MC_{average}$: $\widetilde{MC}_{average}$
- Compute Fourier map M_i from $\widetilde{MC}_{average}$
- Scale M_i by standard deviation
- Truncate low values: set $M_i=0$ if $M_i<0.5\sigma$
- Eliminate regions in M_i with small volume

b. Sharpen M_i , M_{sharp}

c. Histogram equalize M_{sharp} , M_{HE}

d. Filter M_{sharp} by OMIT map: $M_{filtered} = M_{HE} * M_{filter}$

e. Add $M_{filtered}$ to IMC

4. Compute median map M_m from 16 maps in IMC, which is resulting Feature Enhanced Map, $M_{FEM} = M_m$

Storing many maps in memory and computing median map

- Maps are typically stored as arrays of type double numbers
  - **Size of double = 8 bytes**
  - **Storing many (more than 2) maps in memory is typically problematic**

- Store maps as arrays of 1-byte integer (0, 255)
  - **Having 16 "integer maps" is equivalent to 2 "double maps"**

- Let's agree we can afford having 2 double maps in memory at once
  - **This will let us having 16 integers maps at once!**

Storing many maps in memory and computing median map

- Initialize new map with gridding N1*N2*N3
  - **Each grid node is 1-byte integer array of length 16**

- As new map arrives:
  - **Histogram equalize it, so all values are within [0,1] range**
  - **Convert each value *p* into 1-byte integer and add to corresponding grid node array**

$$integer\ j = \begin{Bmatrix} 0, & if\ p \le p_0 \\ min\left\{\frac{256(p-p_0)}{1-p_0}, 255\right\}, & if\ p > p_0 \end{Bmatrix}$$

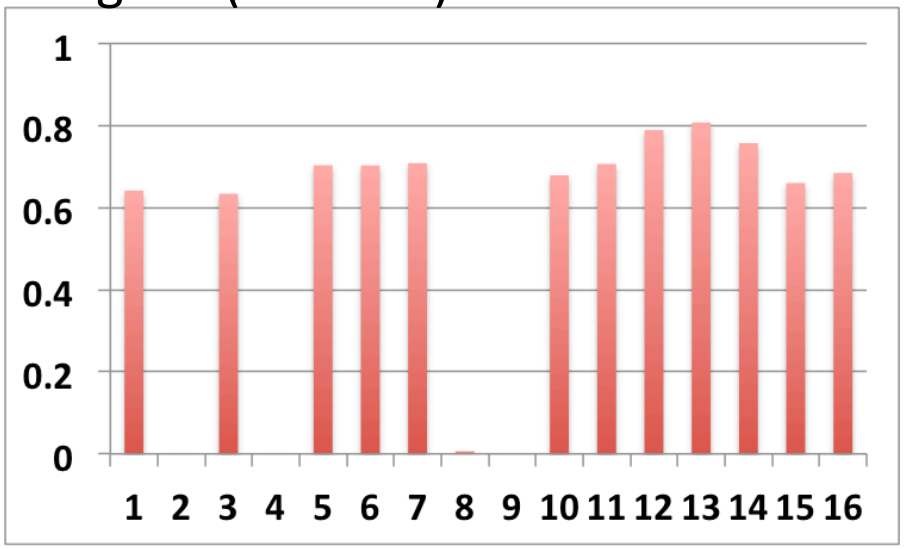
- Final map with each grid point being an array of 1-byte integers

- We want to choose the most frequent (most persistent) map value in each grid node, which is the median of the set of points – **this is resulting FEM map!**

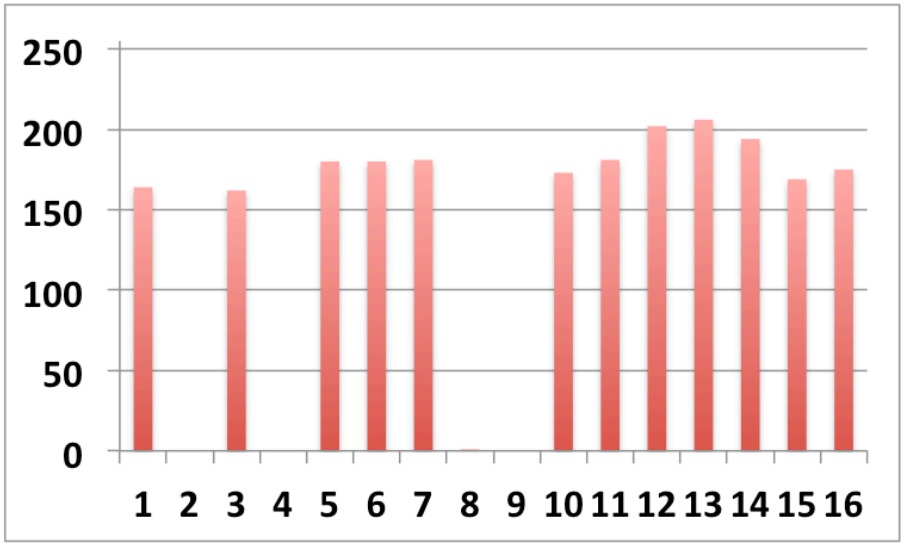
Storing many maps in memory and computing median map

- Values in particular grid node typically look like this:

Original (after HE)



Converted to integers



- Given rather small set of data points (16) spread across [0,255] range (after conversion to integer) it is likely that none of 16 numbers will coincide exactly making it impossible to calculate the median

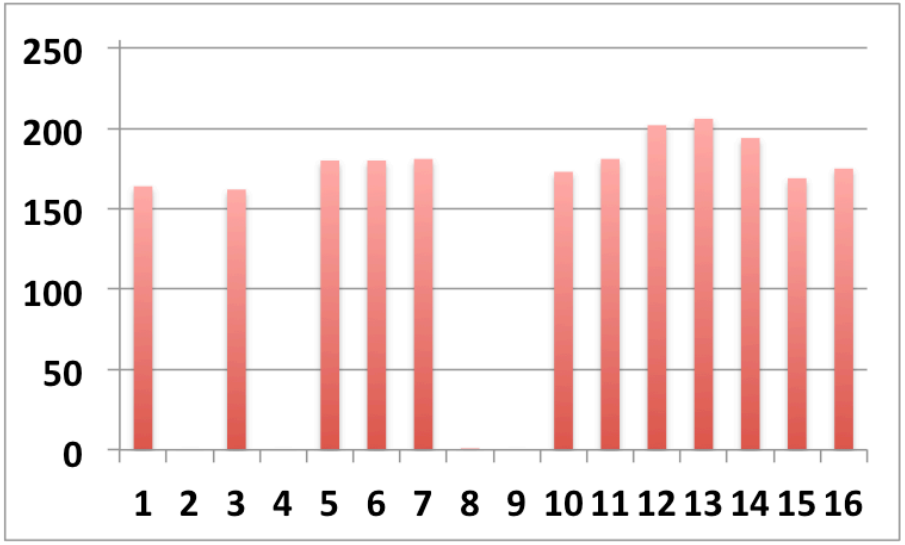
Storing many maps in memory and computing median map

- Convert integer array of 16 values into 256 long array of doubles using transformation

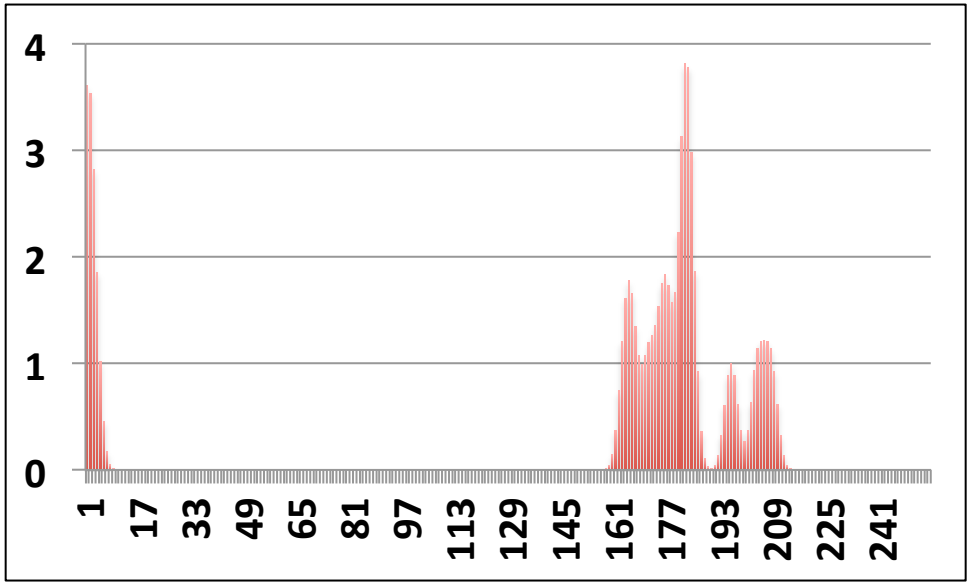
$$f(j) = \sum_{n=1}^{N} exp\left\{\frac{(j-j_n)^2}{2b^2}\right\},\ j = 0, 1, \ldots, 255$$

  - Since we are working with one grid node at a time this conversion is memory cheap

Array of 16 integers

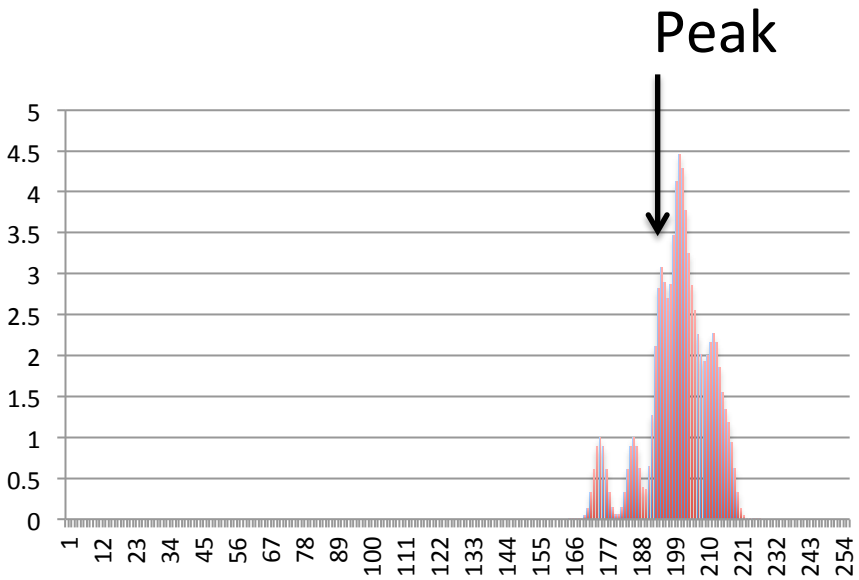


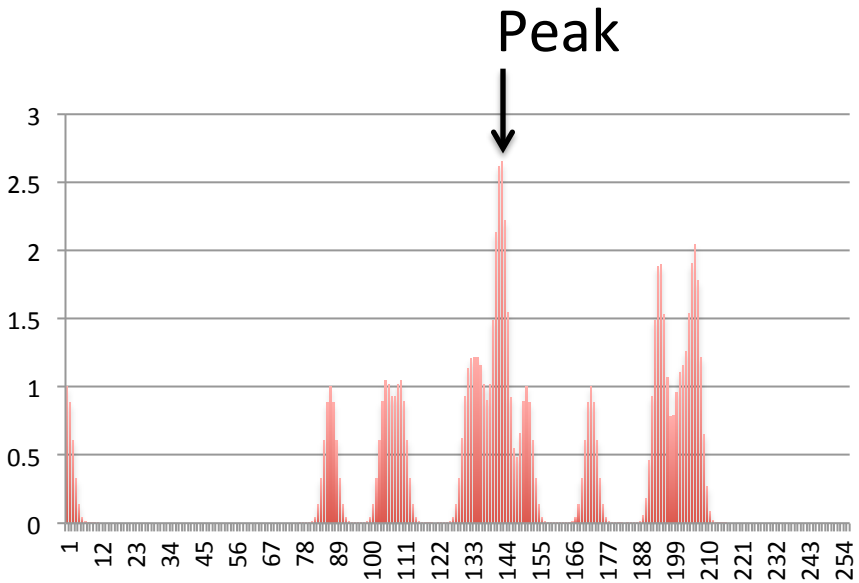
Gaussian smeared array of frequencies

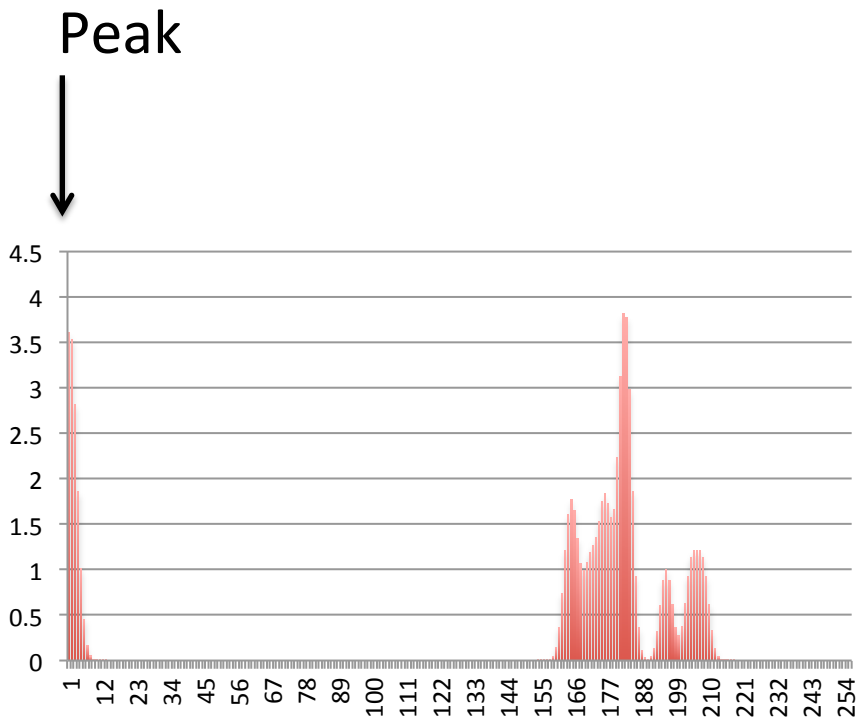


- Find highest peak, $f_{max}(j_{max})$
- Build quadratic approximation f(j-1), f(j_{max}), f(j+1)
- New (resulting) map value is the argument corresponding to the max point of quadratic approximation

Storing many maps in memory and computing median map







Storing many maps in memory and computing median map

1. OMIT map filter
- Compute composite residual OMIT map: M_{cromit}
- Scale M_{cromit} by standard deviation (σ)
- Compute filter: $M_{filter}=0$ if $M_{cromit}<0.5\sigma$ else $M_{filter}=1$

2. Initialize collector of integer maps, IMC

3. For j in j=1,16:

a. Map randomization and averaging
For i, i=1,10:
- Compute 100 map coefficients and average them: $MC_{average}$
- Randomly remove 5% of terms from $MC_{average}$: $\widetilde{MC}_{average}$
- Compute Fourier map M_i from $\widetilde{MC}_{average}$
- Scale M_i by standard deviation
- Truncate low values: set $M_i=0$ if $M_i<0.5\sigma$
- Eliminate regions in M_i with small volume

b. Sharpen M_i , M_{sharp}

c. Histogram equalize M_{sharp} , M_{HE}

d. Filter M_{sharp} by OMIT map: $M_{filtered} = M_{HE} * M_{filter}$

e. Add $M_{filtered}$ to IMC

4. Compute median map M_m from 16 maps in IMC, which is resulting Feature Enhanced Map, $M_{FEM} = M_m$

FEM (Feature Enhanced Map): SUMMARY

- FEM procedure modifies $2m\mathbf{F}_{obs}-D\mathbf{F}_{model}$ σ_A-weighted map to:
  - **Reduce noise**
  - **Retain existing features**
  - **Enhance existing weak features**
  - **Reduce model bias**
- FEM:
  - **Fast to compute: from a few to several minutes**
  - **No user adjustable parameters**
  - **Requires basic input: PDB model, Fobs or Iobs**
  - **Eliminates the need to choose (arbitrary!) map contouring cutoffs**
  - **One map to use for all purposes**
  - **Good for X-ray or neutron data**
- Availability:
  - **Phenix build 1.9-1692 and up: http://www.phenix-online.org/**
  - **GUI**
  - **Command line:**

    phenix.fem model.pdb data.mtz

This will be published (manuscript in works)

FEM development team

- Tom Terwilliger

  ***Resolve*, density modification, treating missing reflections, valuable discussions and ideas, proofreading the manuscript**

- Alexandre Urzhumtsev

  Algorithms for memory efficient handling of many maps in memory, valuable discussions and ideas, proofreading the manuscript

- Dušan Turk

  Kick map idea, valuable discussions

- Nigel Moriarty

  Discussion and help with choosing non-linear map transformation functions at initial steps of FEM development, proofreading the manuscript, suggestions for test cases

- Marat Mustyakimov

  Implementation of ASU maps in CCTBX and providing various related support utility functions

- Nat Echols

  GUI, OMIT maps, suggestions for test cases

- Oleg Sobolev

  Implementation of map connectivity analysis and providing various related support utility functions

- Paul Adams

  Valuable discussions, ideas and critiques; suggestions for test cases, proofreading the manuscript, project PI

Funding

- This work was supported:
  - by the NIH (Project 1P01 GM063210)
  - by the Phenix Industrial Consortium
  - in part by the US Department of Energy under Contract No. DE-AC02-05CH11231

- Alexandre Urzhumtsev thanks the French Infrastructure for Integrated Structural Biology (FRISBI) ANR-10-INSB-05-01 and Instruct, part of the European Strategy Forum on Research Infrastructures (ESFRI) and supported by national member subscription.