New fast algorithm for overall, anisotropic and bulk-solvent scaling

Pavel Afonine

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

Basics: Fmodel – total model structure factor

- Used virtually everywhere:

$$\mathbf{F}_{\text{model}} = k_{\text{overall}} k_{\text{isotropic}} k_{\text{anisotropic}} (\mathbf{F}_{\text{calc}} + k_{\text{mask}} \mathbf{F}_{\text{mask}})$$

$$R = \frac{\sum_{\mathbf{s}} \left| F_{\text{obs}} - |\mathbf{F}_{\text{model}}| \right|}{\sum_{\mathbf{s}} |F_{\text{obs}}|}$$

R-factors

$$2mF_{\text{obs}} - D|\mathbf{F}_{\text{model}}|$$

Maps

$$T = \sum_{\mathbf{s}} \left(\frac{(\alpha|\mathbf{F}_{\text{model}}|)^2}{\varepsilon\beta} + ln \left(I_0 \left(\frac{2\alpha F_{\text{obs}} |\mathbf{F}_{\text{model}}|}{\varepsilon\beta} \right) \right) \right) + \cdots$$

Refinement targets

Basics: Fmodel – total model structure factor

- Overall scale (scalar) factor k_{overall}

$$\mathbf{F}_{\text{model}} = k_{\text{overall}} k_{\text{isotropic}} k_{\text{anisotropic}} (\mathbf{F}_{\text{calc}} + k_{\text{mask}} \mathbf{F}_{\text{mask}})$$

$$LS = \sum\nolimits_{\mathbf{s}} (F_{\text{obs}} - k_{\text{overall}} |\mathbf{F}'_{\text{model}}|)^2$$

$$\mathbf{F}'_{\text{model}} = k_{\text{isotropic}} k_{\text{anisotropic}} (\mathbf{F}_{\text{calc}} + k_{\text{mask}} \mathbf{F}_{\text{mask}})$$

$$\partial LS / \partial k_{\text{overall}} = 0$$

$$k_{\text{overall}} = \sum\nolimits_{\mathbf{s}} (F_{\text{obs}} |\mathbf{F}'_{\text{model}}| / |\mathbf{F}'_{\text{model}}|^2)$$

Basics: Fmodel – total model structure factor

- Overall isotropic scale factor $k_{\text{isotropic}}$

$$\mathbf{F}_{\text{model}} = k_{\text{overall}} k_{\text{isotropic}} k_{\text{anisotropic}} (\mathbf{F}_{\text{calc}} + k_{\text{mask}} \mathbf{F}_{\text{mask}})$$

$k_{\text{isotropic}}$ – resolution dependent scale factor. For example:

$$k_{isotropic} = \exp\left(-\frac{B_{\text{isotropic}}}{4} s^2\right)$$

$$s^2 = \mathbf{s}^t \mathbf{G}^* \mathbf{s}$$

$$\mathbf{s} = \begin{pmatrix} h \\ k \\ l \end{pmatrix}$$

$$\mathbf{G}^* = \begin{pmatrix} \mathbf{a}^*\mathbf{a}^* & \mathbf{a}^*\mathbf{b}^* & \mathbf{a}^*\mathbf{c}^* \\ \mathbf{a}^*\mathbf{b}^* & \mathbf{b}^*\mathbf{b}^* & \mathbf{b}^*\mathbf{c}^* \\ \mathbf{a}^*\mathbf{c}^* & \mathbf{b}^*\mathbf{c}^* & \mathbf{c}^*\mathbf{c}^* \end{pmatrix} = \begin{pmatrix} a^*a^* & a^*b^*cos\gamma^* & a^*c^*cos\beta^* \\ a^*b^*cos\gamma^* & b^*b^* & b^*c^*cos\alpha^* \\ a^*c^*cos\beta^* & b^*c^*cos\alpha^* & c^*c^* \end{pmatrix}$$

$B_{\text{isotropic}}$ – refinable parameter

Basics: Fmodel – total model structure factor

- Overall scale factor $k_{\text{anisotropic}}$ – Exponential model (Sheriff & Hendrickson, 1987)

$$\mathbf{F}_{\text{model}} = k_{\text{overall}} k_{\text{isotropic}} k_{\text{anisotropic}} (\mathbf{F}_{\text{calc}} + k_{\text{mask}} \mathbf{F}_{\text{mask}})$$

$k_{\text{anisotropic}}$ – Miller index dependent scale factor. For example:

$$k_{\text{anisotropic}} = \exp\left(-2\pi^2 \mathbf{s}^t \mathbf{U}_{\text{cryst}} \mathbf{s}\right) =$$

$$= -2\pi^2 (u_{11}h^2 + u_{22}k^2 + u_{33}l^2 + u_{12}2hk + u_{13}2hl + u_{23}2kl)$$

$\mathbf{U}_{\text{CRYST}}$ is 3x3 symmetric anisotropy scale matrix with 6 (or less) refinable parameters

Symmetry constraints apply →

Crystal System	Restrictions on U
Triclinic 1-2	None
Monoclinic 3-15	$U_{13}=U_{23}=0$ when $\beta=\alpha=90^\circ$ $U_{12}=U_{23}=0$ when $\gamma=\alpha=90^\circ$ $U_{12}=U_{13}=0$ when $\gamma=\beta=90^\circ$
Orthorhombic 16-74	$U_{12}=U_{13}=U_{23}=0$
Tetragonal 75-142	$U_{11}=U_{22}$ and $U_{12}=U_{13}=U_{23}=0$
Rhombohedral (trigonal) 143-167	$U_{11}=U_{22}=U_{33}$ and $U_{12}=U_{13}=U_{23}$
Hexagonal 168-194	$U_{11}=U_{22}$ and $U_{13}=U_{23}=0$
Cubic 195-230	$U_{11}=U_{22}=U_{33}$ and $U_{12}=U_{13}=U_{23}=0$ (=isotropic)

Basics: Fmodel – total model structure factor

- Overall scale factor $k_{\text{anisotropic}}$ – Polynomial model (Usón et al., 1999; Parkin et al., 1995)

$$\mathbf{F}_{\text{model}} = k_{\text{overall}} k_{\text{isotropic}} k_{\text{anisotropic}} (\mathbf{F}_{\text{calc}} + k_{\text{mask}} \mathbf{F}_{\text{mask}})$$

$$k_{\text{anisotropic}} = [h^2 a^{*2}(a_1 s + a_7) + k^2 b^{*2}(a_2 s + a_8) + l^2 c^{*2}(a_3 s + a_9) + 2klb^* c^*(a_4 s + a_{10}) + 2hla^* c^*(a_5 s + a_{11}) + 2hka^* b^*(a_6 s + a_{12})]$$

Relation to exponential model: Taylor series expansion of exponential model…

$$exp(-2\pi^2 \mathbf{s}^t \mathbf{U}_{\text{cryst}} \mathbf{s}) \approx 1 - 2\pi^2 \mathbf{s}^t \mathbf{U}_{\text{cryst}} \mathbf{s} + 2\pi^4 (\mathbf{s}^t \mathbf{U}_{\text{cryst}} \mathbf{s})(\mathbf{s}^t \mathbf{U}_{\text{cryst}} \mathbf{s})$$

$$k_{\text{anisotropic}} = \mathbf{s}^t \mathbf{V}_0 \mathbf{s} + (\mathbf{s}^t \mathbf{V}_1 \mathbf{s}) s^2$$

$\mathbf{V}_0$ and $\mathbf{V}_1$ are symmetric 3*3 matrices (12 refinable parameters)

… with constant terms omitted (or included into refinable parameters)

=> Obvious problem: $k_{\text{anisotropic}}$ = 0 for F(000) reflection

Basics: Fmodel – total model structure factor

- Bulk-solvent scale factor k_{mask}

$$\mathbf{F}_{\text{model}} = k_{\text{overall}} k_{\text{isotropic}} k_{\text{anisotropic}} (\mathbf{F}_{\text{calc}} + k_{\text{mask}} \mathbf{F}_{\text{mask}})$$

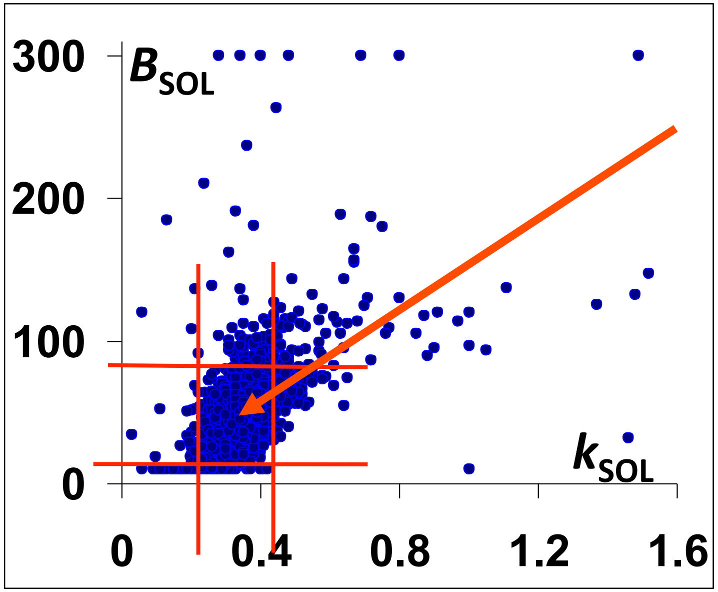
k_{mask} – resolution dependent bulk-solvent mask scale. For example, it can be:

$$k_{\text{mask}} = k_{\text{solvent}} \exp\left(-\frac{B_{\text{solvent}}}{4} s^2\right)$$

Phillips, 1980; Jiang & Brünger, 1994

Fokine & Urzhumtsev, 2002

PDB survey



Average values:

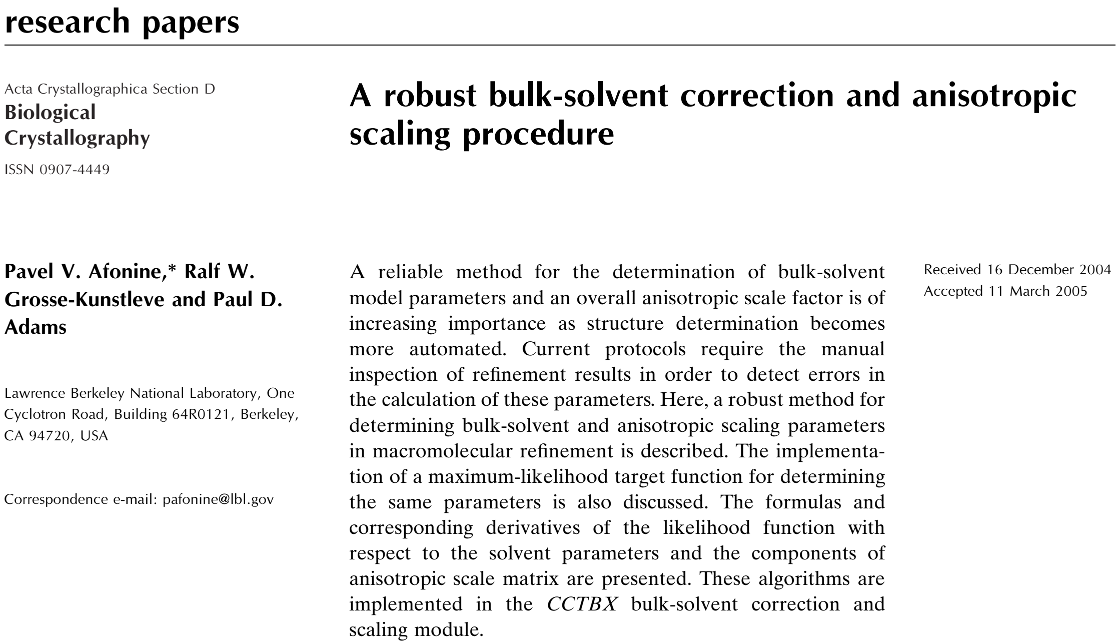
k_{SOL} = 0.35 (e/Å^3)

B_{SOL} = 46.0 (Å^2)

Basics: Fmodel – total model structure factor

- Algorithm used in many Phenix tools (*phenix.refine*, *phenix.maps*, *phenix.model_vs_data*, …)

$$\mathbf{F}_{\text{model}} = k_{\text{overall}} k_{\text{isotropic}} k_{\text{anisotropic}} (\mathbf{F}_{\text{calc}} + k_{\text{mask}} \mathbf{F}_{\text{mask}})$$



research papers

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

A robust bulk-solvent correction and anisotropic scaling procedure

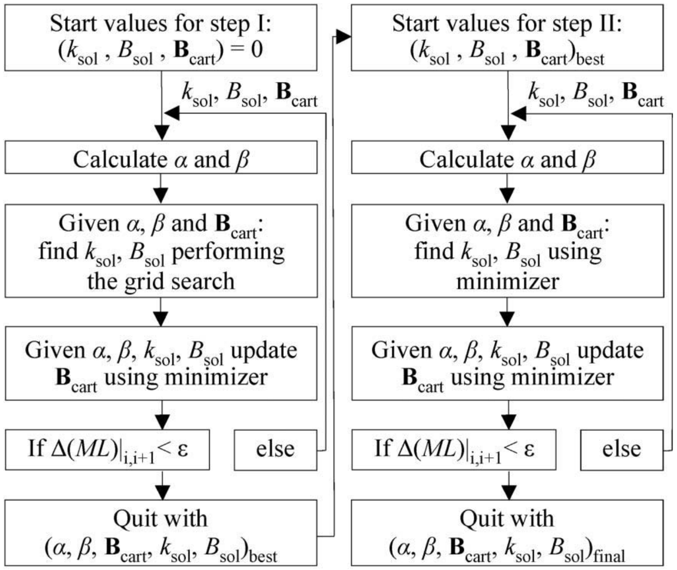
Pavel V. Afonine,* Ralf W. Grosse-Kunstleve and Paul D. Adams

Lawrence Berkeley National Laboratory, One Cyclotron Road, Building 64R0121, Berkeley, CA 94720, USA

Correspondence e-mail: pafonine@lbl.gov

Received 16 December 2004
Accepted 11 March 2005

A reliable method for the determination of bulk-solvent model parameters and an overall anisotropic scale factor is of increasing importance as structure determination becomes more automated. Current protocols require the manual inspection of refinement results in order to detect errors in the calculation of these parameters. Here, a robust method for determining bulk-solvent and anisotropic scaling parameters in macromolecular refinement is described. The implementation of a maximum-likelihood target function for determining the same parameters is also discussed. The formulas and corresponding derivatives of the likelihood function with respect to the solvent parameters and the components of anisotropic scale matrix are presented. These algorithms are implemented in the *CCTBX* bulk-solvent correction and scaling module.



Start values for step I: $(k_{sol}, B_{sol}, \mathbf{B}_{cart}) = 0$
$k_{sol}, B_{sol}, \mathbf{B}_{cart}$
Calculate α and β
Given α, β and $\mathbf{B}_{cart}$: find k_{sol}, B_{sol} performing the grid search
Given α, β, k_{sol}, B_{sol} update $\mathbf{B}_{cart}$ using minimizer
If $\Delta(ML)|_{i,i+1} < \varepsilon$
else
Quit with $(\alpha, \beta, \mathbf{B}_{cart}, k_{sol}, B_{sol})_{best}$
Start values for step II: $(k_{sol}, B_{sol}, \mathbf{B}_{cart})_{best}$
$k_{sol}, B_{sol}, \mathbf{B}_{cart}$
Calculate α and β
Given α, β and $\mathbf{B}_{cart}$: find k_{sol}, B_{sol} using minimizer
Given α, β, k_{sol}, B_{sol} update $\mathbf{B}_{cart}$ using minimizer
If $\Delta(ML)|_{i,i+1} < \varepsilon$
else
Quit with $(\alpha, \beta, \mathbf{B}_{cart}, k_{sol}, B_{sol})_{final}$

Works great since 2004 …

… **but slow**: **~20 minutes** for ribosome (on our fastest computer), 2.7Å , 1582749 reflections

- **Re-thinking determination of overall and bulk-solvent scales**

$$\mathbf{F}_{\mathrm{model}} = k_{\mathrm{overall}} k_{\mathrm{isotropic}} k_{\mathrm{anisotropic}} (\mathbf{F}_{\mathrm{calc}} + k_{\mathrm{mask}} \mathbf{F}_{\mathrm{mask}})$$

- Make it fast:
  - Do not use minimizer;
  - Find all scales analytically;
  - Do not spend <u>*any*</u> time on decoupling scales.

… took a week to achieve

- Make sure it is as reliable as the current algorithm:
  - works for large sample of PDB (40,000).

… where another 2 months went

- **Overall scale factor** $k_{\text{anisotropic}} = \exp\left(-2\pi^2 \mathbf{s}^t \mathbf{U}_{\text{cryst}} \mathbf{s}\right)$ (Exponential model)

$$\mathbf{F}_{\text{model}} = k_{\text{overall}} k_{\text{isotropic}} k_{\text{anisotropic}} \left(\mathbf{F}_{\text{calc}} + k_{\text{mask}} \mathbf{F}_{\text{mask}}\right)$$

- **Approach:**

  Instead of minimizing $LS = \sum_{\mathbf{s}} (F_{\text{obs}} - k_{\text{overall}} |\mathbf{F}'_{\text{model}}|)^2$

  Minimize $LSL = \sum_{\mathbf{s}} (ln(F_{\text{obs}}) - ln(|\mathbf{F}_{\text{model}}|))^2$

  Assumptions:

  1. Fobs > 0 and |Fmodel| > 0, which is always the case
  2. Minima of *LS* and *LSL* are close enough to each other and to the minimum of R-factor. This is not an obvious statement.

- **Solution:** $\widetilde{\mathbf{U}}_{\text{cryst}} = \mathbf{M}^{-1}\mathbf{b}$

$\mathbf{M} = \sum_{\mathbf{s}} \mathbf{V} \otimes \mathbf{V}$, $\mathbf{V} = (h^2, k^2, l^2, 2hk, 2hl, 2kl)^t$, $\otimes$ denotes outer product and $\mathbf{b} = -\sum_{\mathbf{s}} Z\, \mathbf{V}$

$$Z = \frac{1}{2\pi^2} ln\left(F_{\text{obs}}\left(k_{\text{overall}} k_{\text{isotropic}} |\mathbf{F}_{\text{calc}} + k_{\text{mask}} \mathbf{F}_{\text{mask}}|\right)^{-1}\right)$$

Takes a tiny fraction of a second even for ribosome

55 lines of C++ code (mmtbx/bulk_solvent)

- Overall scale factor $k_{\text{anisotropic}}$ (Exponential model): symmetry constraints (**Ralf**)

$$\mathbf{U}_{\text{cryst}} = (U_{11} \quad U_{22} \quad U_{33} \quad U_{12} \quad U_{13} \quad U_{23}) = \mathbf{C}^t\,\mathbf{U}_{\text{ind}}$$

C – symmetry constraints matrix

Example: for tetragonal system

$$\mathbf{C} = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

$$\mathbf{M}_{\mathrm{C}}\,\widetilde{\mathbf{U}}_{\text{ind}} = \mathbf{b}_{\mathrm{C}}$$

$$\mathbf{M}_{\boldsymbol{C}} = \textstyle\sum_{\mathbf{h}} \mathbf{V}_{\boldsymbol{C}} \otimes \mathbf{V}_{\boldsymbol{C}},\ \mathbf{V}_{\mathrm{C}} = \mathbf{CV}, \mathbf{b}_{\mathrm{C}} = -\textstyle\sum_{\mathbf{h}} Z\mathbf{V}_{\mathrm{C}}.$$

Solution: $\widetilde{\mathbf{U}}_{\text{ind}} = \mathbf{M}_{\mathrm{C}}^{-1}\mathbf{b}_{\mathrm{C}}$

- Overall scale factor $k_{\text{anisotropic}}$ (Polynomial model)

$$\mathbf{F}_{\text{model}} = k_{\text{overall}} k_{\text{isotropic}} k_{\text{anisotropic}} (\mathbf{F}_{\text{calc}} + k_{\text{mask}} \mathbf{F}_{\text{mask}})$$

$$k_{\text{anisotropic}} = 1 + \mathbf{s}^t \mathbf{V}_0 \mathbf{s} + (\mathbf{s}^t \mathbf{V}_1 \mathbf{s}) s^2$$

$\widetilde{\mathbf{V}}_\mathbf{0}$ and $\widetilde{\mathbf{V}}_\mathbf{1}$ are found by solving system of 12 linear equations arising from $\nabla_\mathbf{V} LS = 0$

$LS = \sum_\mathbf{s} (F_{\text{obs}} - |\mathbf{F}_{\text{model}}|)^2$

- Bulk solvent scale and overall isotropic scales: k_{overall}, $k_{\text{isotropic}}$ and k_{mask}

$$\mathbf{F}_{\text{model}} = k_{\text{overall}} k_{\text{isotropic}} k_{\text{anisotropic}} (\mathbf{F}_{\text{calc}} + k_{\text{mask}} \mathbf{F}_{\text{mask}})$$

- **Approach:**

Define: $K = \left(k_{\text{overall}} k_{\text{isotropic}} k_{\text{anisotropic}}\right)^{-2}$, $x = k_{\text{mask}}$ and $I = F_{\text{obs}}^2$

And fit: $KI = |\mathbf{F}_{\text{calc}} + x\mathbf{F}_{\text{mask}}|^2$

by minimizing in resolution bins $LS_s(K,x) = \sum_s (KI - |\mathbf{F}_{\text{calc}} + x\mathbf{F}_{\text{mask}}|^2)^2 \rightarrow min|_{K,x}$

- **Solution:**

$\mathbf{F}_{\text{calc}} = p + ir$ and $\mathbf{F}_{\text{mask}} = q + it$ $\quad w = q^2 + t^2, v = pq + rt$ and $u = p^2 + r^2$

$$\begin{cases} \frac{\partial}{\partial K} LS_s(K,x) = -\sum_s \{(x^2 w_s + 2x v_s + u_s) - K I_s\} I_s = 0 \\ \frac{\partial}{\partial x} LS_s(K,x) = 2\sum_s \{(x^2 w_s + 2x v_s + u_s) - K I_s\}(x w_s + v_s) = 0 \end{cases}$$

$x^3 + ax^2 + bx + c = 0$ (solved using standard methods available in scitbx)

~150 lines of C++ code (mmtbx/bulk_solvent);
plus 123 lines of C++ code for cubic equation solver (scitbx/math/cubic_equation.h)

- **Overall procedure**

$$\mathbf{F}_{\mathrm{model}} = k_{\mathrm{overall}} k_{\mathrm{isotropic}} k_{\mathrm{anisotropic}} (\mathbf{F}_{\mathrm{calc}} + k_{\mathrm{mask}} \mathbf{F}_{\mathrm{mask}})$$

- Find scales *K* and *x*
- Then find $k_{\mathrm{anisotropic}}$
- Repeated several times until convergence

700 lines of Python code (mmtbx/bulk_solvent/scaler.py)

Usage examples and tests: phenix_regression/bulk_sol_and_scaling_fast/tst.py

Python interface is obvious (minimum parameters):

```
result = mmtbx.bulk_sovlent.scaler.run(
       f_obs = f_obs,
       f_calc = f_calc,
       f_mask = f_mask)

f_model = result.f_model
k_mask  = result.k_mask
…
```

- **Caveats at very low resolution**

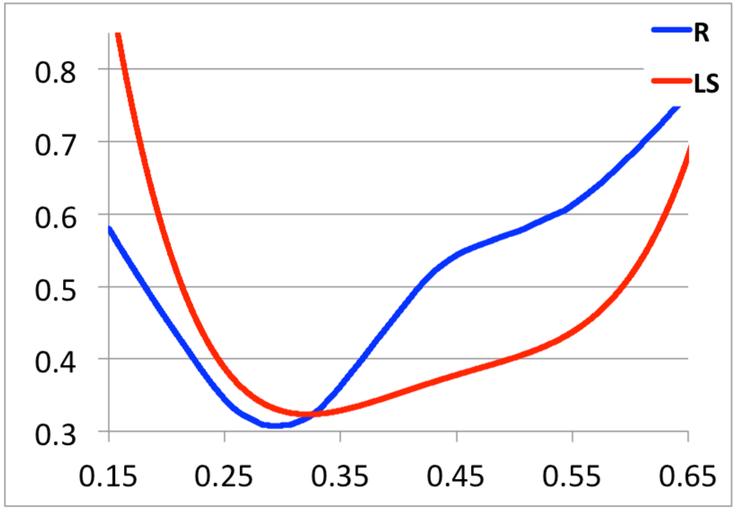
$$\mathbf{F}_{\text{model}} = k_{\text{overall}} k_{\text{isotropic}} k_{\text{anisotropic}} (\mathbf{F}_{\text{calc}} + k_{\text{mask}} \mathbf{F}_{\text{mask}})$$

1. In lowest resolution zone: $\mathbf{F}_{\text{mask}} = -\lambda \mathbf{F}_{\text{calc}}$

   All combinations $k_{\text{overall}}(1 - \lambda k_{\text{mask}}) = const$ will result in the same *R*-factor

2. *<u>Rarely</u> in lowest resolution zone the minimum of LS and R can be very different:*

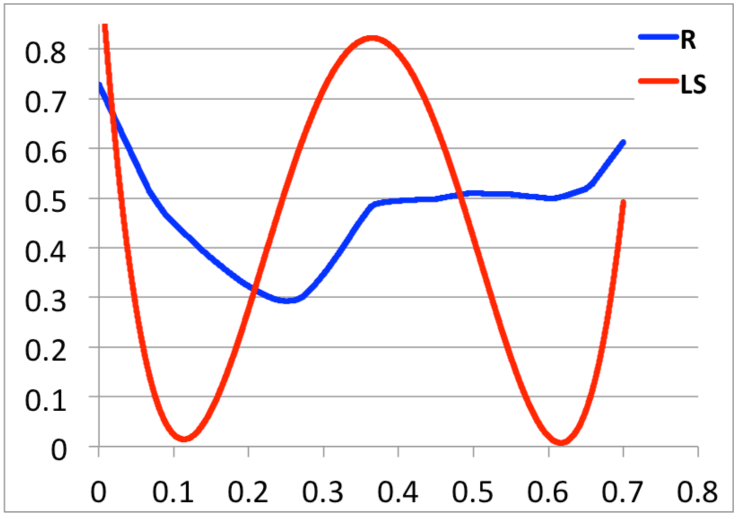
1kwn



$R(K=0.0961, k_{\text{mask}}=0.2913)=0.3073$

$R(K=0.0863, k_{\text{mask}}=0.3218)=0.3372$

1hqw



$R(K=0.0131, k_{\text{mask}}=0.2500)=0.2924$

$R(K=0.0151, k_{\text{mask}}=0.6166)=0.5046$

• **Twinning** - definitions

For *N* twin domains the total intensity is $I(\mathbf{s}) = \alpha_1 I_{\text{model}}(\mathbf{s}_1) + \ldots + \alpha_N I_{\text{model}}(\mathbf{s}_N)$

$\mathbf{s}_N = \mathbf{T}_N\mathbf{s}$

$\mathbf{T}_N$ – matrix defining twin operator for *N*th twin domain

α_N – twin fraction for *N*th twin domain

$$I(\mathbf{s}) = I_{\text{model}} = |\mathbf{F}_{\text{model}}|^2 = K|\mathbf{F}_{\text{calc}} + x\mathbf{F}_{\text{mask}}|^2$$

K and *x* are all the same for all twin domains

Constraint condition

$$C(\alpha_1, \ldots, \alpha_N) = \alpha_1 + \ldots + \alpha_N - 1 = 0$$

Note: it is not the same as defining (N-1) refinable *α*

$$I(\mathbf{s}) = \alpha_1 I_{\text{model}}(\mathbf{s}_1) + \ldots + (1 - \sum_{i=1}^{N-1} \alpha_i) I_{\text{model}}(\mathbf{s}_N)$$

Which would make $I_{\text{model}}(\mathbf{s}_N)$ “*special*” (and which may not be numerically good)

• **Twinning** – finding twin fractions

We find N twin fractions by minimizing

$$LSA(\alpha_1,\ldots,\alpha_N)=\frac{1}{2}\sum_{\mathbf{s}}\left[\alpha_1 I_{\mathrm{model}}(\mathbf{s}_1)+\ldots+\alpha_N I_{\mathrm{model}}(\mathbf{s}_N)-I(\mathbf{s})\right]^2$$

with constraint condition

$$C(\alpha_1,\ldots,\alpha_N)=\alpha_1+\ldots+\alpha_N-1=0$$

This is equivalent to unconstrained minimization of the Lagrangian:

$$LSA(\alpha_1,\ldots,\alpha_N,\lambda)=LSA(\alpha_1,\ldots,\alpha_N)+\lambda C(\alpha_1,\ldots,\alpha_N)\rightarrow \min$$

This generates a system of N+1 linear equations with N+1 variables:

$$\begin{cases}\partial LS(\alpha_1,\ldots,\alpha_N,\lambda)/\partial\alpha_1 = 0\\ \qquad\qquad\ldots\\ \partial LS(\alpha_1,\ldots,\alpha_N,\lambda)/\partial\alpha_N = 0\\ \partial LS(\alpha_1,\ldots,\alpha_N,\lambda)/\partial\lambda \quad = 0\end{cases}$$

Solution: $\left(\tilde{\alpha}_1,\ldots,\tilde{\alpha}_N,\tilde{\lambda}\right)^t = \mathbf{M}^{-1}\mathbf{b}$

$$\mathbf{M}=\begin{bmatrix}\sum_{\mathbf{s}}\mathbf{V}\otimes\mathbf{V} & \mathbf{1}\\ \mathbf{1} & 0\end{bmatrix}$$

$$\mathbf{V}=\left(I_1(\mathbf{T}_1\mathbf{s}),\ldots,I_N(\mathbf{T}_N\mathbf{s})\right)$$

$$\mathbf{b}=\left(\textstyle\sum_{\mathbf{s}} I(\mathbf{s})I_1(\mathbf{T}_1\mathbf{s}),\ldots,\sum_{\mathbf{s}} I(\mathbf{s})I_N(\mathbf{T}_N\mathbf{s}),1\right)^t$$

- **Twinning** – finding overall and bulk-solvent scales *K* and k_{mask}

Once twin fractions $\alpha_{1,\ \dots,}\ \alpha_N$ are available…

… we can search for scales *K* and k_{mask}

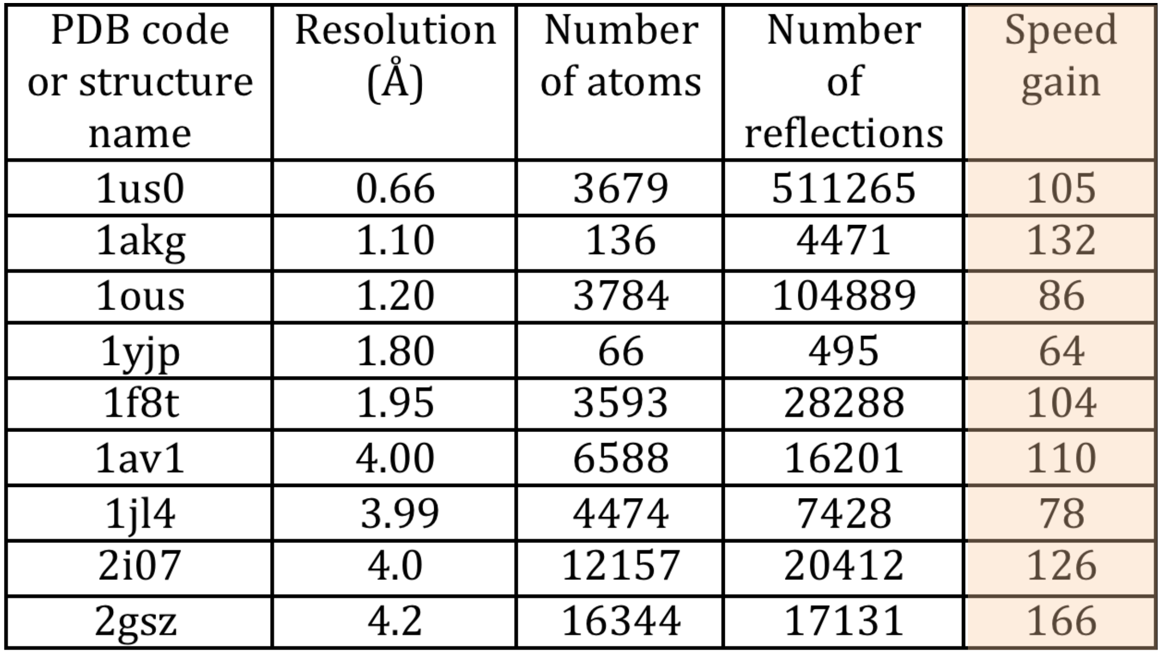
$$KI(\mathbf{s}) = \alpha_1 \left|\mathbf{F}_{calc}(\mathbf{s}_1) + x\mathbf{F}_{mask}\left(\mathbf{s}_1\right)\right|^2 + \ldots + \alpha_N \left|\mathbf{F}_{calc}(\mathbf{s}_N) + x\mathbf{F}_{mask}\left(\mathbf{s}_N\right)\right|^2$$

… and we can re-use procedure described for non-twinned case to solve:

$$LS_s(K, x) = \sum_{s} (KI - |\mathbf{F}_{\text{calc}} + x\mathbf{F}_{\text{mask}}|^2)^2 \rightarrow min|_{K,x}$$

- Overall procedure consists of iterations of twin fractions and overall and bulk-solvent scales determination

- **Results**: runtime comparisons *new* vs *old* (Afonine *et al.*, 2005)



PDB code or structure name	Resolution (Å)	Number of atoms	Number of reflections	Speed gain
1us0	0.66	3679	511265	105
1akg	1.10	136	4471	132
1ous	1.20	3784	104889	86
1yjp	1.80	66	495	64
1f8t	1.95	3593	28288	104
1av1	4.00	6588	16201	110
1jl4	3.99	4474	7428	78
2i07	4.0	12157	20412	126
2gsz	4.2	16344	17131	166

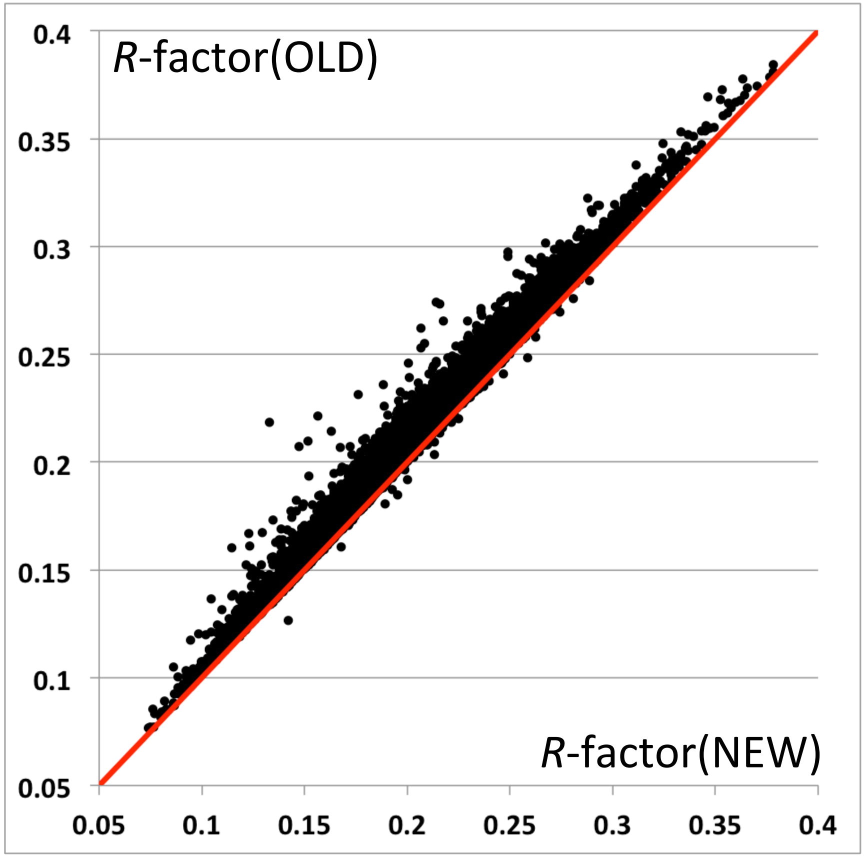
Ribosome (2.7Å , 1582749 reflections; computer: *chevy*):

old: **~20 minutes**

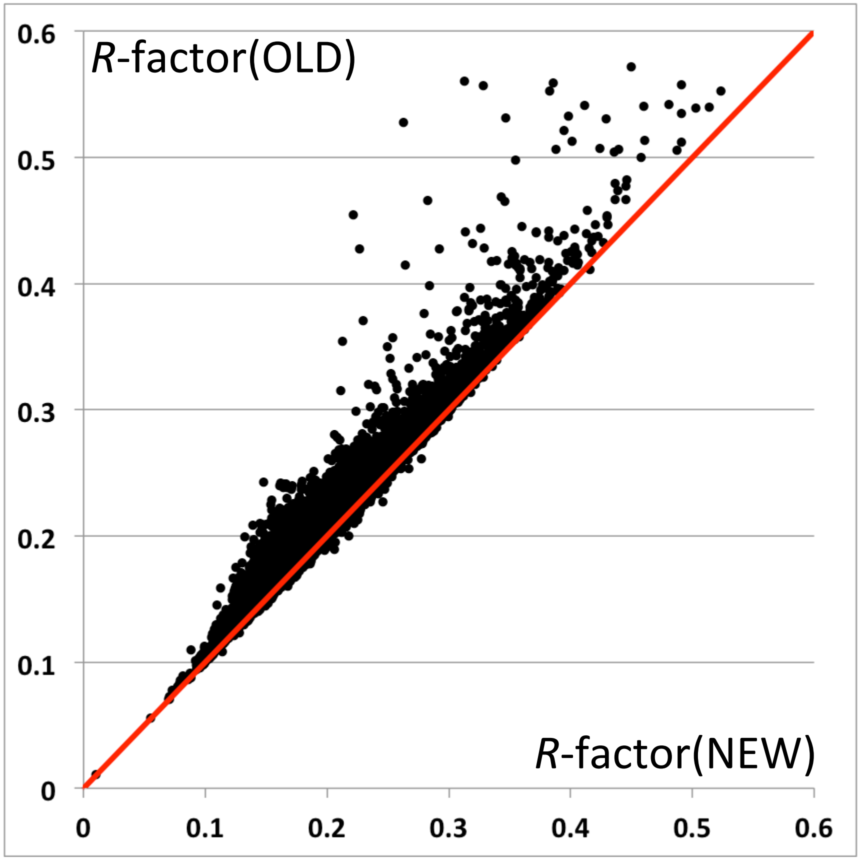
new: **~30 seconds**

- **Results**: Systematic tests for approximately 40,000 datasets from PDB

All reflections

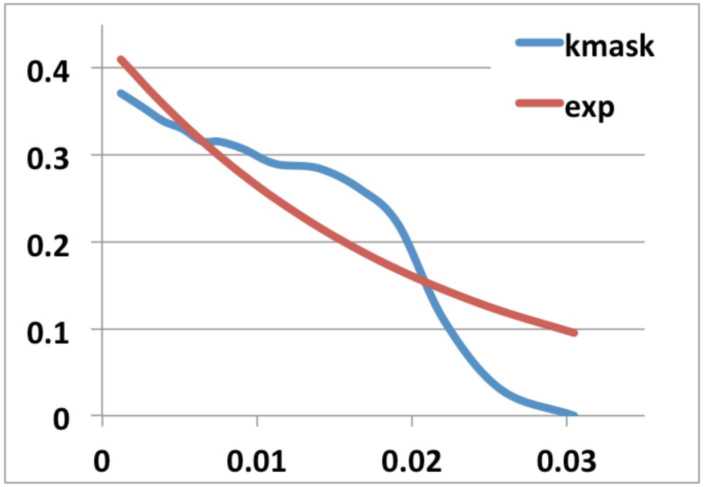


Low resolution reflections

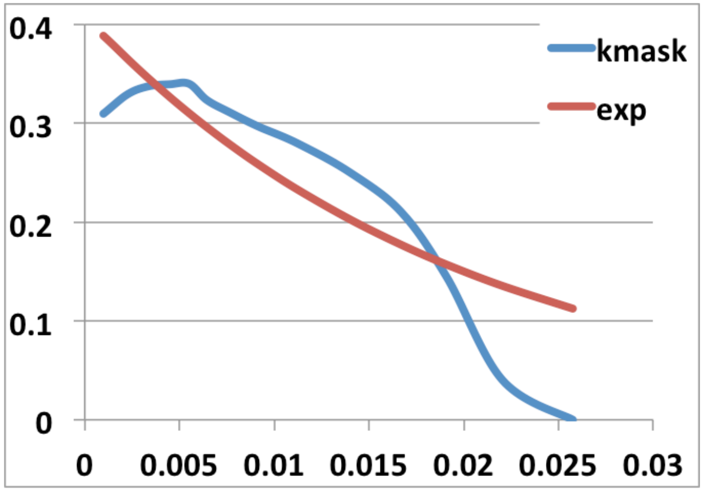


- **Results**: k_{mask} – selected examples

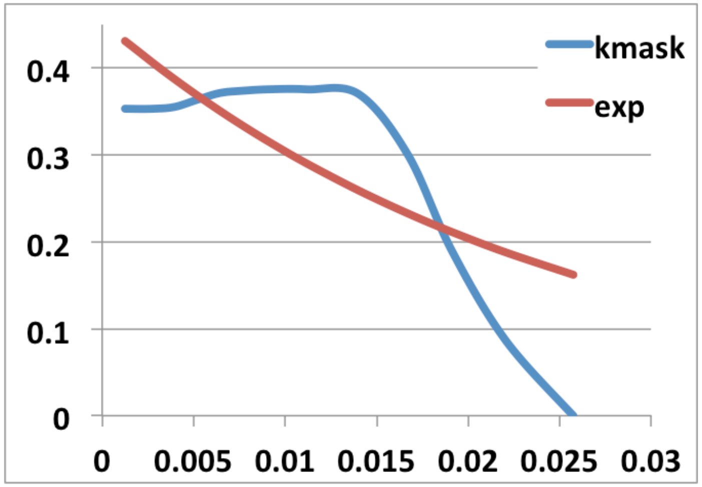
2wxd



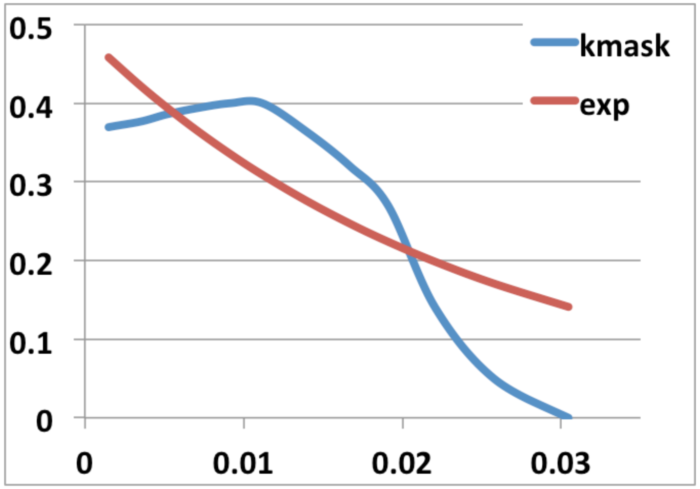
2bwx



2x8h

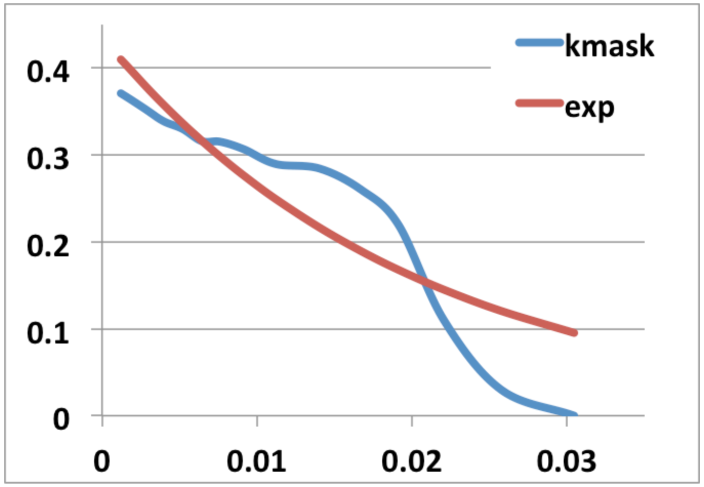


3ahs

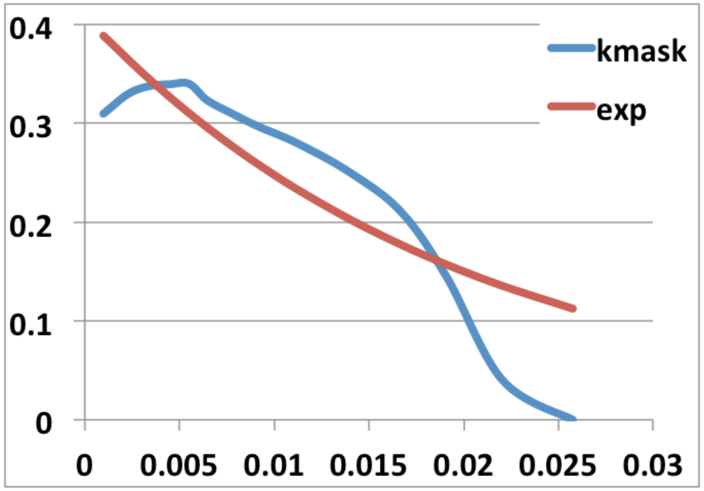


- **Results**: k_{mask} – selected examples

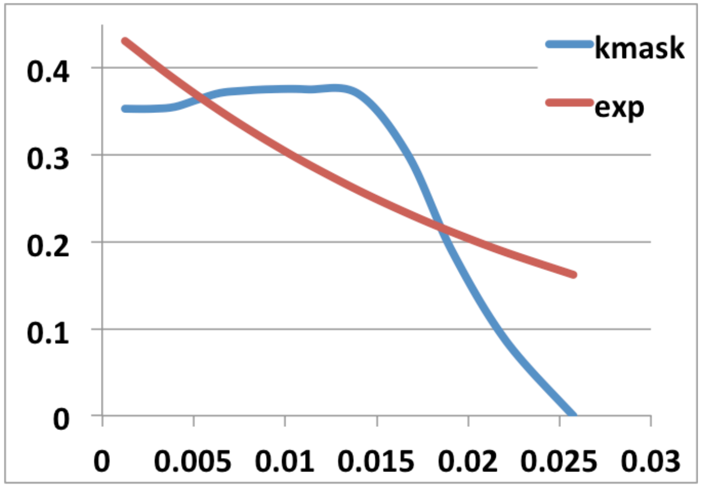
2wxd



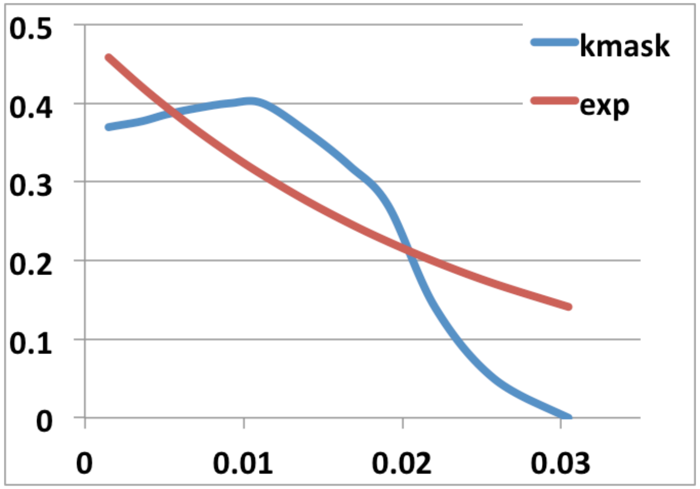
2bwx



2x8h



3ahs



$$\xcancel{k_{\text{mask}} = k_{\text{solvent}} \exp\left(-\frac{B_{\text{solvent}}}{4} s^2\right)}$$

- **Results**: *R*-factors – selected examples

2wxd

Resolution	Rold	Rnew
68.784-11.003	**0.3128**	**0.2672**
10.984- 8.503	**0.1972**	**0.1764**
8.500- 7.500	**0.2210**	**0.1977**
7.499- 6.503	**0.2296**	**0.2061**
6.498- 6.002	**0.2191**	**0.2042**
5.999- 5.501	0.1948	0.1785
5.498- 5.000	0.1717	0.1657
4.999- 4.500	0.1412	0.1405
4.500- 4.000	0.1275	0.1263
4.000- 3.750	0.1317	0.1298
3.750- 3.500	0.1368	0.1338
3.500- 3.250	0.1417	0.1377
3.250- 3.000	0.1562	0.1517
3.000- 2.750	0.1619	0.1581
2.750- 2.500	0.1514	0.1480
2.500- 2.250	0.1473	0.1440
2.250- 2.000	0.1438	0.1374
2.000- 1.750	0.1493	0.1485
1.750- 1.600	**0.1988**	**0.1647**

- **Results**: *R*-factors – selected examples

3ahs

```
 Resolution    Rold   Rnew
38.265-10.001 0.3235 0.2896
 9.992- 7.508 0.2411 0.1807
 7.491- 6.504 0.2401 0.1731
 6.499- 5.501 0.2364 0.1801
 5.499- 5.001 0.1977 0.1500
 4.996- 4.502 0.1601 0.1484
 4.500- 4.001 0.1464 0.1423
 4.000- 3.751 0.1512 0.1457
 3.750- 3.501 0.1539 0.1483
 3.500- 3.250 0.1715 0.1639
 3.250- 3.000 0.1835 0.1747
 3.000- 2.750 0.1928 0.1854
 2.749- 2.500 0.1922 0.1855
 2.500- 2.250 0.1709 0.1660
 2.250- 2.000 0.1562 0.1472
 2.000- 1.750 0.1480 0.1468
 1.750- 1.500 0.1430 0.1272
 1.500- 1.320 0.2162 0.1475
```

- **Who work on this:**

Pavel Afonine

Sacha Urzhumtsev **(math, algorithms)**

Ralf Grosse-Kunstleve **(coding, symmetry)**

Paul Adams **(valuable critiques and encouragement to look into how kmask looks like)**