Algorithms for accelerated single-particle 3D reconstruction

Hans Elmlund
Monash University
Biomedicine Discovery Institute
hans.elmlund@monash.edu
http://simplecryoem.com

Outline

- ✓Background
- ✓Mathematical techniques for accelerating convergence of single-particle orientation search
- ✓Our Stochastic Hill Climbing (SHC) approach to single-particle 3D refinement
- ✓Results

Categorization of 3D orientation refinement approaches developed to date

1. Discrete or continuous search
2. Probabilistic or deterministic orientation determination
3. Stochastic or deterministic optimization

Discrete search typically implies the use of polar rather than Cartesian image representations, *i.e.* projection matching (SPIDER, SPARX, EMAN2, PRIME)

Probabilistic algorithms assign each particle image a distribution of orientations with weights (RELION, PRIME, Cryo-SPARC)

Stochastic optimization algorithms are typically less dependent on accurate starting models (PRIME, VIPER, Cryo-SPARC + more)

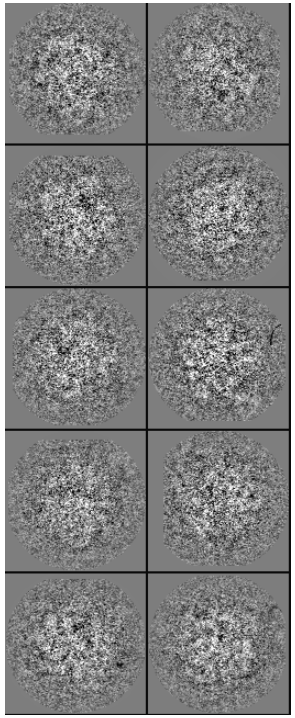
Why projection matching (Penczek early 90s) is clever

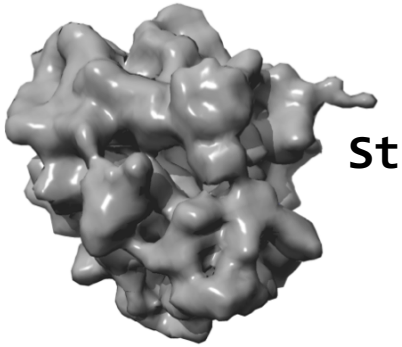
1. Extract projections from volume to generate 2D references
2. Extract concentric rings within a circular mask in real space from 2D references to create a polar representation
3. Same for particles
4. Do 1D FFTs along the rings
5. Use the circular convolution theorem to obtain rotational correlations

It is clever because: you get all the rotational correlations between a pair "for free" (or at a cost roughly equivalent of correlating a pair of Cartesian images)

Reference-based 3D reconstruction

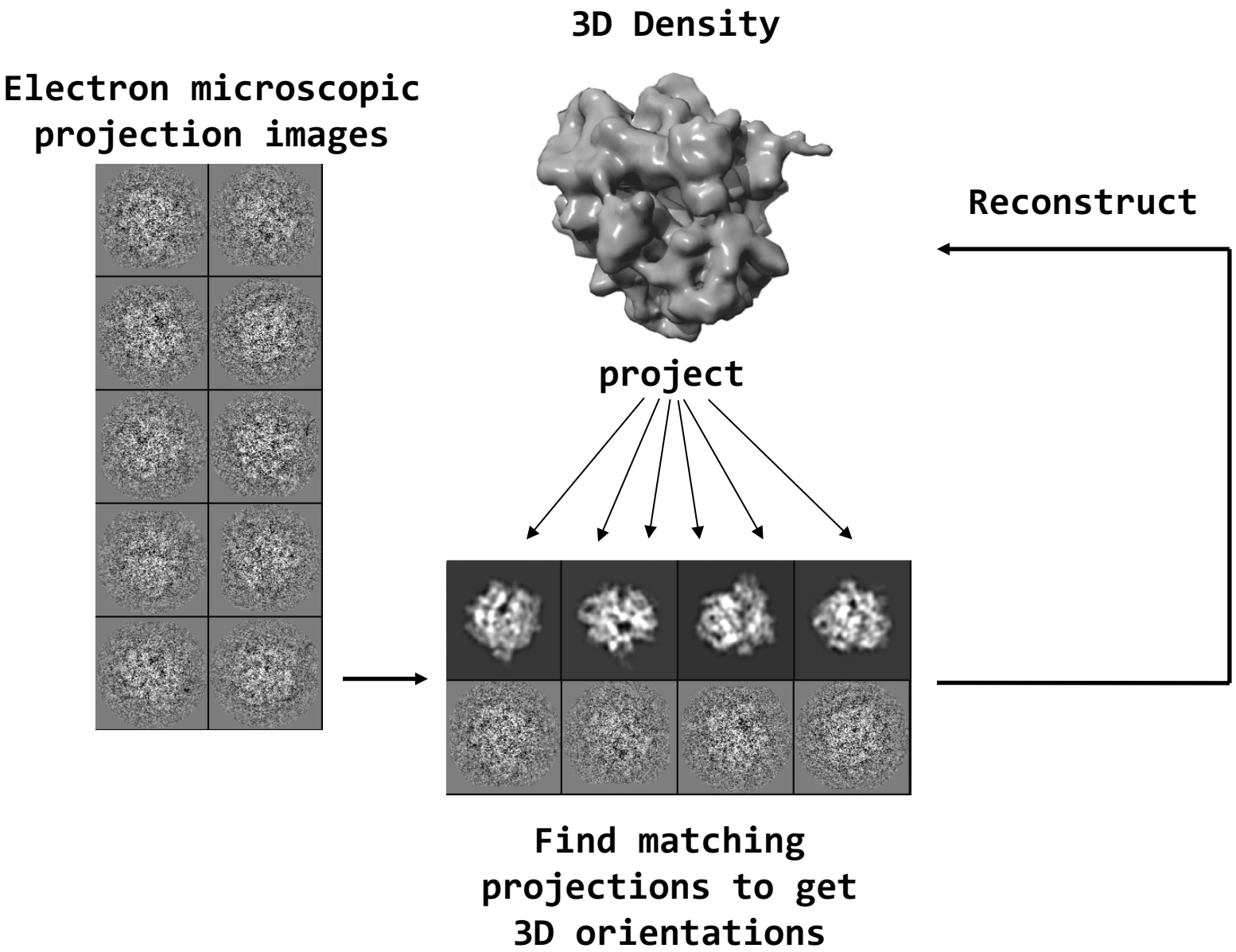
Electron microscopic projection images



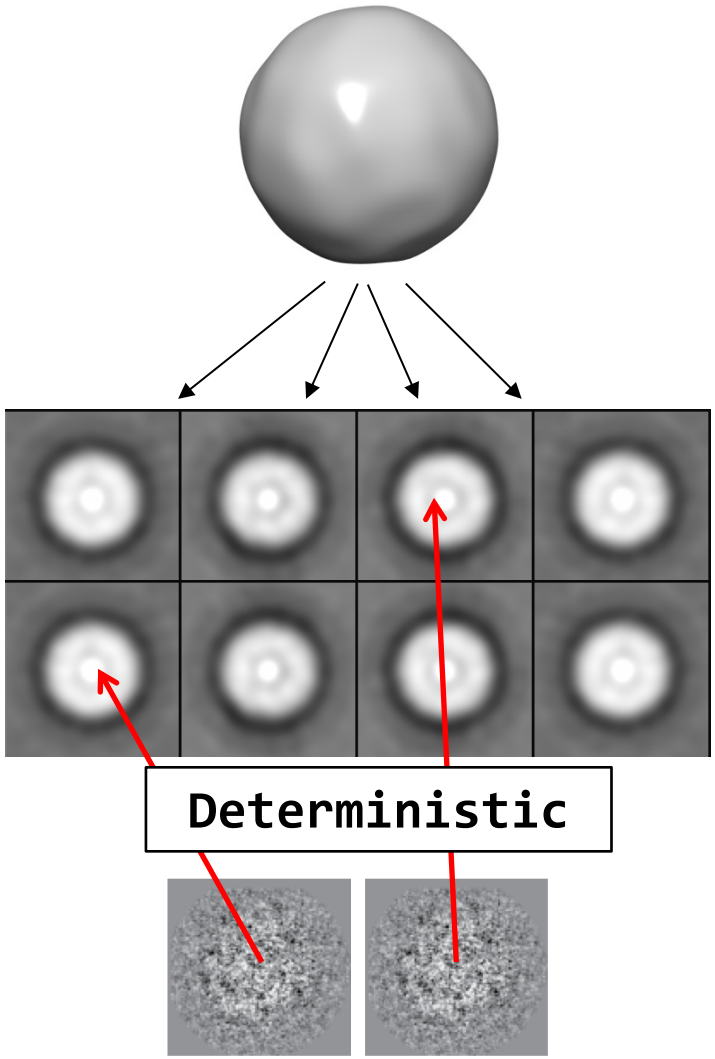


Starting model (3D)

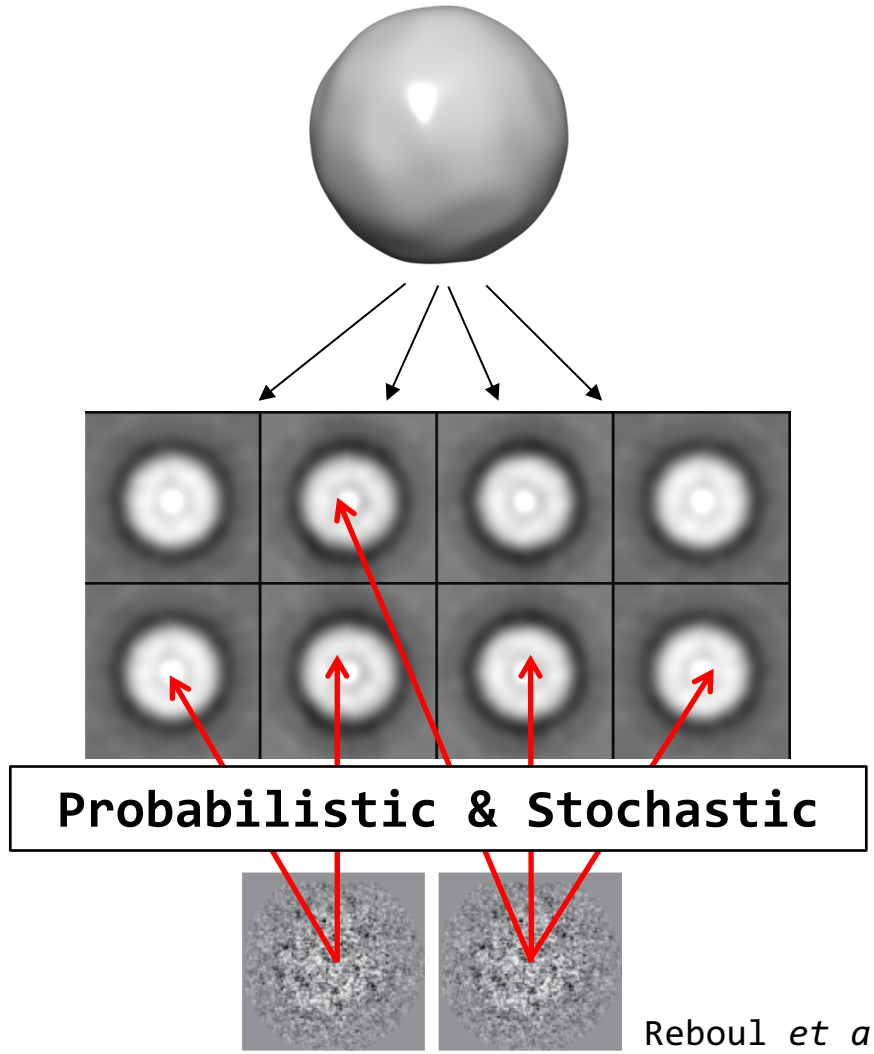
Algorithm Challenge



Deterministic methods



Our developments



Reboul *et al.* 2017
Reboul *et al.* 2016
Elmlund *et al.* 2013

Strategies for speedup (independent of hardware)

Code optimization
(*quite technical*)

1. Create larger parallel regions that do more work
2. Optimize memory access
3. Isolate memory allocations into a single step
4. Caching

Mathematical techniques
(*more fun to talk about*)

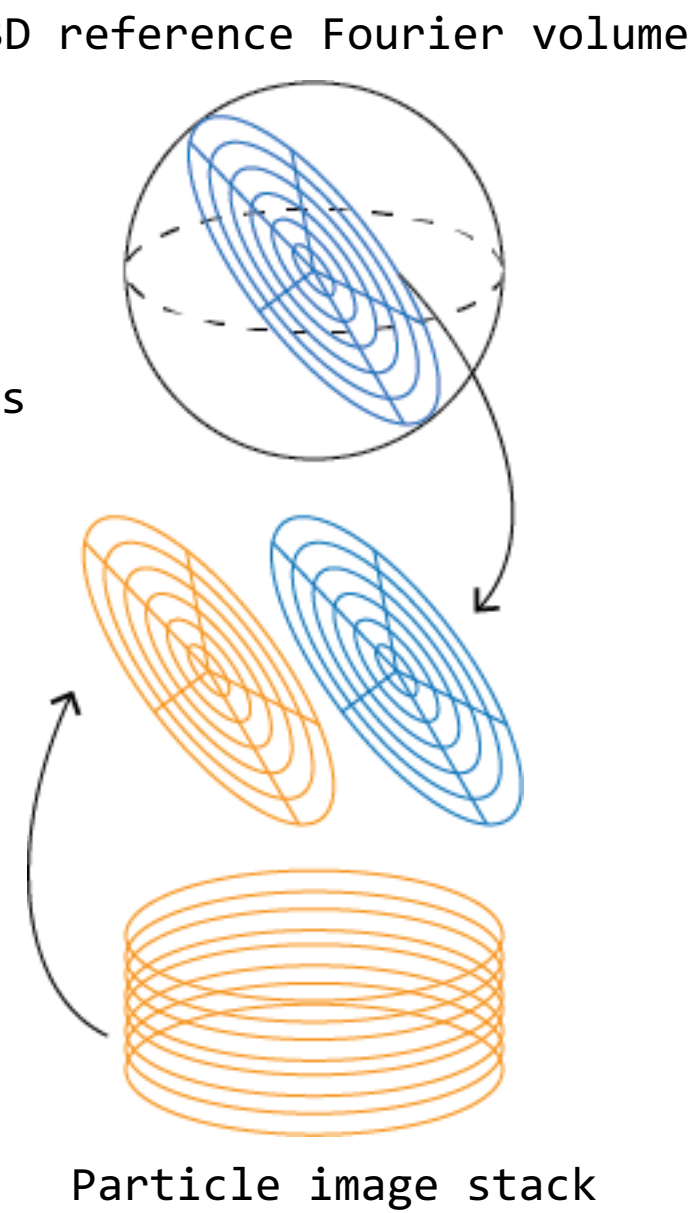
1. Reduce computational costs by *exploiting properties of the Fourier transform* (harmonic analysis)
2. The use of *analytical methods for optimization* to replace costly black-box direct search approaches
3. Formulating the orientation refinement as as an *incremental learning* approach

Our implementation of projection matching

1. Extract polar FTs directly from the reference volume with convolution interpolation
2. Convert the particle images to polar FTs as well
3. Do complex-to-complex 1D FFTs along the resolution rings in the polar FT representations (*e.g. Fourier-Fourier space*)
4. Use the circular convolution theorem to obtain rotational correlations

Why:

- ✓ Obtain 2D references in a single interpolation step
- ✓ Central pixels of the images in real-space are *not* omitted from the calculation
- ✓ Full control of band-pass range



Unpublished

In-plane rotational search. Objective function

2D polar reference central section

Particle

$$\lambda = \sum_k \sum_\phi J \cdot Y(k,\phi) \cdot X^*(k,\phi)/N$$

Jacobian is needed because polar coordinates

Normalization term (square root of powers)

We calculate the correlation per resolution ring:

$$\lambda_k = \sum_\phi Y_k(\phi) \cdot X_k^*(\phi - \phi_0)$$

Rotation by a discrete angle understood to be cyclical

- ✓ Drop the Jacobian because improved noise robustness
- ✓ N still there but not relevant for this discussion

In-plane rotational search. Objective function

The circular convolution theorem allows us to formulate the objective function as:

Inverse FFT to obtain the rotational correlations

$$\mathcal{F}^{-1}\{\mathcal{F}\{Y_k\} \cdot \mathcal{F}\{X_k\}^*\}\left(\tilde{\phi}\right) = \sum_{\phi=0}^{N_\phi - 1} Y_k(\phi) \cdot X_k^*\left(\tilde{\phi} - \phi\right)$$

1D C2C FFTs of the polar FTs along resolution rings

Unpublished

In-plane rotational search. Exploiting Friedel's symmetry

Strategy for speedup: divide the FFT of Y into two independent real-valued FFTs of the real and imaginary components separately

$$Y_k(\phi) = \begin{cases} f_k(\phi) & \phi < N_\phi/2 \\ f_k^*(\phi - N_\phi/2) & \text{else} \end{cases}$$

✓ Friedel's symmetry

$$\mathcal{F}\left\{Y^{(r)}\right\}(k) = \begin{cases} 2\mathcal{F}\left\{s^{(r)}\right\}(k/2) & k \text{ even} \\ 0 & k \text{ odd} \end{cases}$$

✓ Roughly 4X faster than F{Y}

$$\mathcal{F}\left\{Y^{(i)}\right\} =$$

$$= \begin{cases} 0 & k \text{ even} \\ 2\mathcal{F}\left\{s^{(i)} \exp\left(-i\pi n/N_s\right)\right\}\left(\frac{(k-1)}{2}\right) & k \text{ odd} \end{cases}$$

✓ Roughly 2X faster than F{Y} because only half of the components

Conclusion: Possible to cut the compute due to FFT of X and Y by 25% by exploiting Friedel's symmetry

Unpublished

Rotational origin refinement. L-BFGS-B with analytical derivatives

Fourier shift theorem in polar coordinates:

$$\mathcal{F}\left[f\left(x-x_0, y-y_0\right)\right] = F\left(k,\phi\right)\exp\left[-i\left(t^{(x_0)}(k,\phi)x_0 + t^{(y_0)}(k,\phi)y_0\right)\right]$$

$$t^{(x_0)}(k,\phi) = 2\pi\frac{k\cos\left(\phi\right)}{N_x} \qquad t^{(y_0)}(k,\phi) = 2\pi\frac{k\sin\left(\phi\right)}{N_y}$$

Results in the objective function:

$$\lambda = \sum_{k,\phi} Y\left(k,\phi\right) X^*\left(\phi-\phi_0,k\right)\exp\left[i\left(t^{(x_0)}\left(k,\phi\right)\cdot x_0 + t^{(y_0)}\left(k,\phi\right)\cdot y_0\right)\right]/N$$

with gradient:

$$\frac{\partial}{\partial x_0}\lambda = i\sum_{k,\phi} Y\left(k,\phi\right) X^*\left(\phi-\phi_0,k\right) t^{(x_0)}(k,\phi)\exp\left[\ldots\right]/N$$

$$\frac{\partial}{\partial y_0}\lambda = i\sum_{k,\phi} Y\left(k,\phi\right) X^*\left(\phi-\phi_0,k\right) t^{(y_0)}(k,\phi)\exp\left[\ldots\right]/N$$

Unpublished

Rotational origin refinement. L-BFGS-B with analytical derivatives

We developed to following procedure to include the discrete in-plane angle in the search:

1. Initialize (x_0,y_0), typically with (0,0). Exhaustively determine best in-plane rotation.
2. Perform one iteration of L-BFGS-B with fixed in-plane rotation.
3. Exhaustively determine best in-plane rotation, given origin shift of (x_0,y_0) from the last L-BFGS-B iteration. If stopping criterion not met, go to step 2. If stopping criterion is met, but in-plane rotation has changed, go to step 2. Otherwise, terminate.

✓ The bound constraint (L-BFGS**-B**) is VERY important
✓ This is about 5X faster than using a Nelder-Mead (simplex) direct search optimizer and gives slightly better results

Unpublished

Incremental learning update for accelerated convergence rate

Batch approach

1. Update all image orientations/weights
2. Reconstruct volume
3. Go to 1 or stop

Guarantees largest improvement/iteration

Incremental learning

Online approach

1. Update a single image orientation/weight
2. Update volume
3. Go to 1 or stop

Computationally inefficient & would slow down convergence rate

1. Update a fraction of the image orientations/weights
2. Update volume
3. Go to 1 or stop

Challenges:

How to update the volume?

How to select which particles to update?

Incremental learning update for accelerated convergence rate

Reference volume

Unnormalised volume

$$V(h,k,l) = \frac{U}{\rho}$$

Sampling density matrix

Reconstruction equations

$$U = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{h',k'} H_i(h',k')X_i(h',k')\tau_{ij}\tilde{w}_{j;(h',k')}^{(h,k,l)}$$

$$\rho = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{h',k'} H_i^2(h',k')\tau_{ij}\tilde{w}_{j;(h',k')}^{(h,k,l)}$$

CTF**2

Orientation weight

Interpolation function

Incremental learning approach

$$U^{(t)} = (1-\delta)U^{(t-1)} + U_{P_\delta}^{(t)}$$

$$\rho^{(t)} = (1-\delta)V^{(t-1)} + \rho_{P_\delta}^{(t)}$$

t is iteration number

δ is learning rate

P_δ is set of particles updated

$U_{P\delta}$ is fractional unnormalised volume

$\rho_{P\delta}$ is fractional sampling density matrix

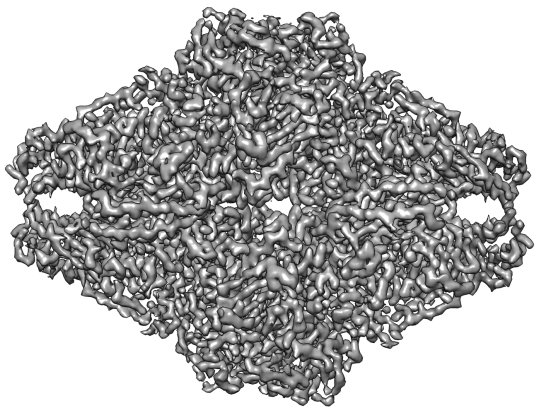
P_δ is deterministically determined based on update frequency

Unpublished

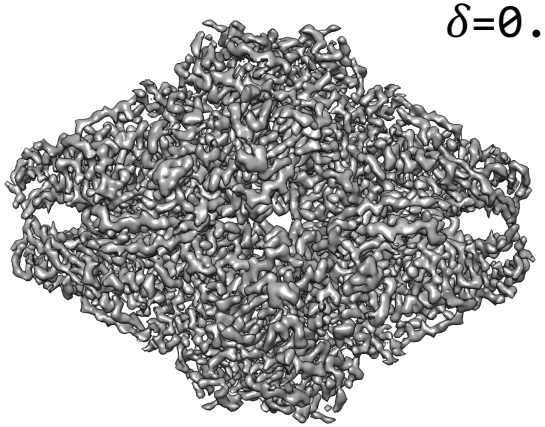
Incremental learning update for accelerated convergence rate

Test data set: 5000 256x256 images of beta-gal (Steve's EMAN2 test data set)

δ = 1.0



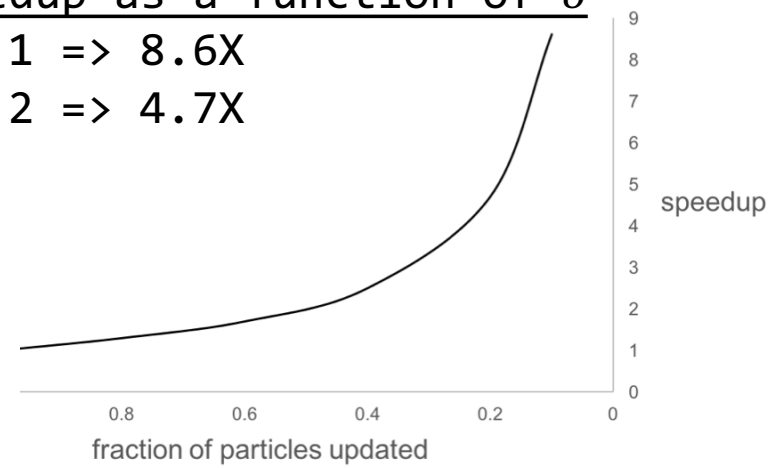
δ = 0.1

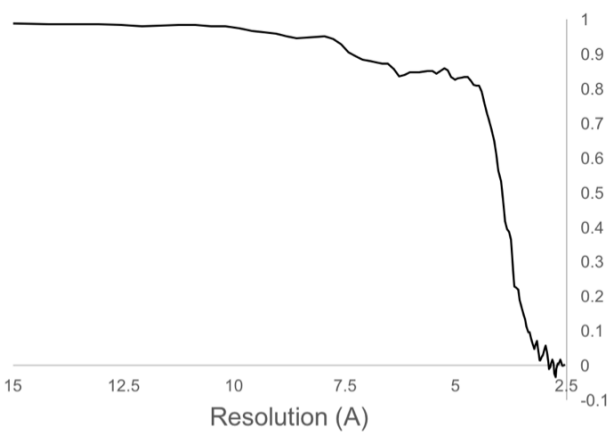


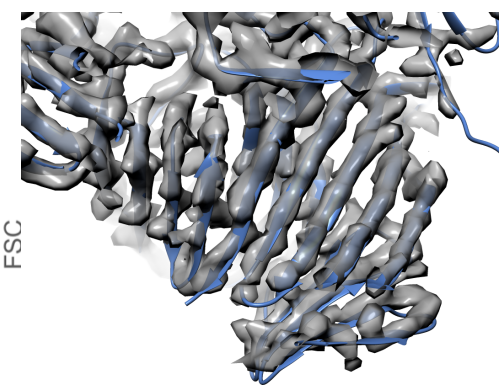
Speedup as a function of δ

δ=0.1 => 8.6X

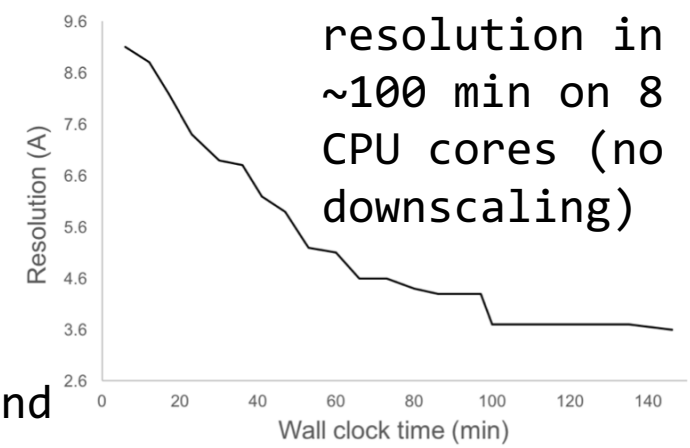
δ=0.2 => 4.7X







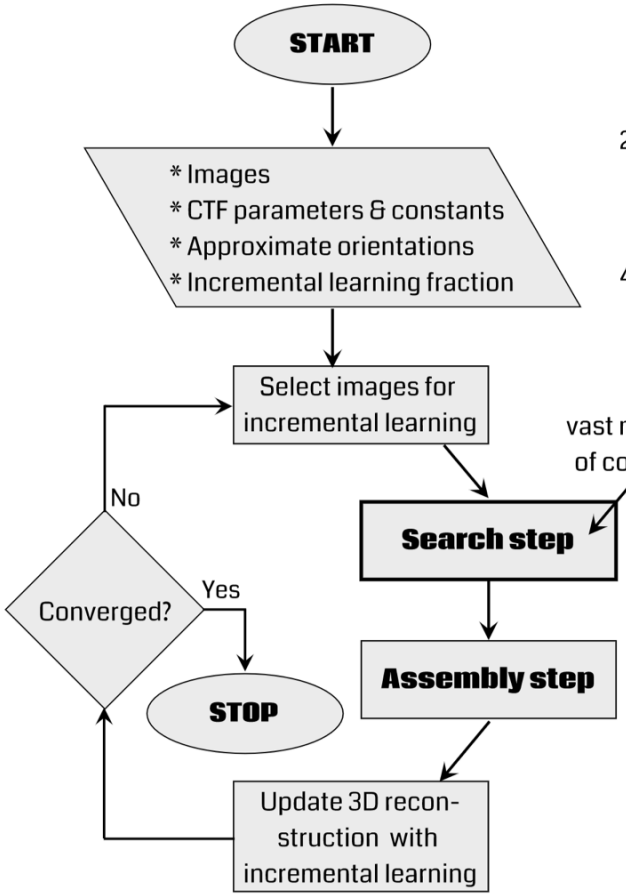
<3h from start to end on a MacBook Pro

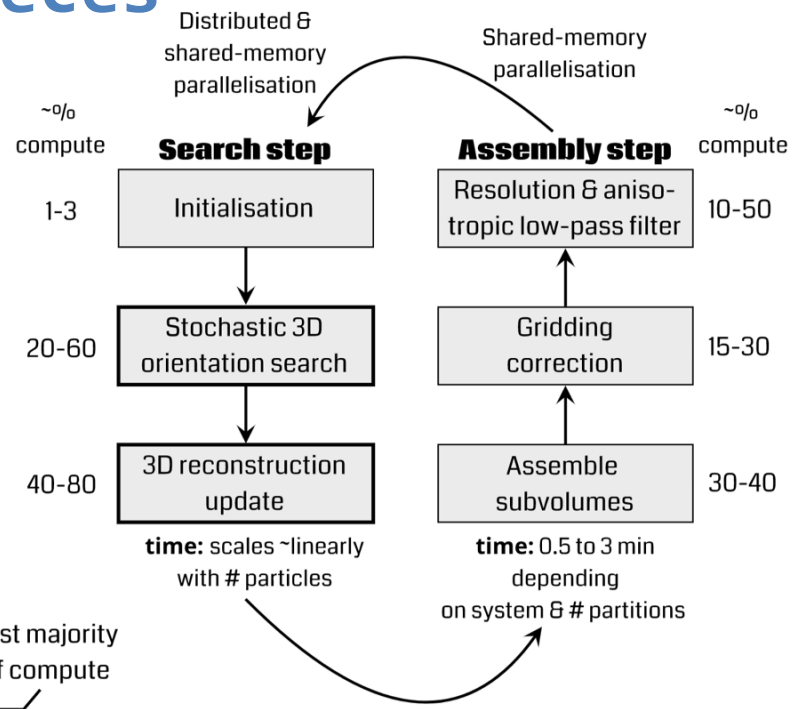


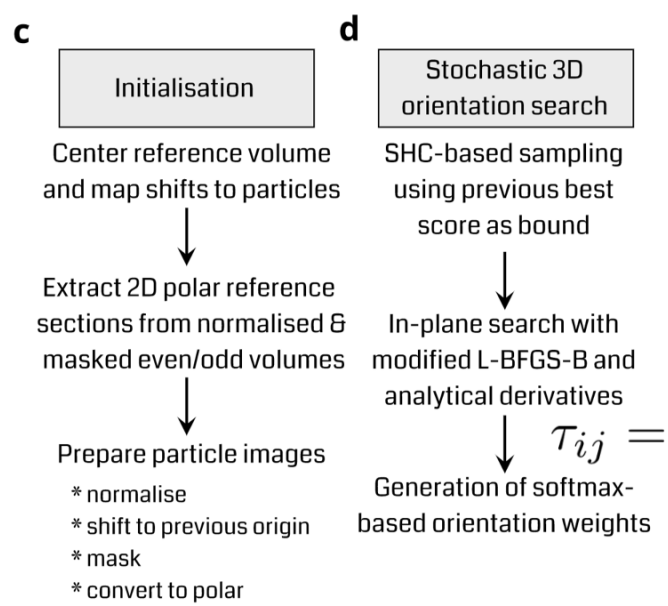
✓ Below 4 A resolution in ~100 min on 8 CPU cores (no downscaling)

Unpublished

Putting the pieces together







- ✓ Fourier techniques for accelerated rotational search
- ✓ Analytical derivatives in conjunction with L-BFGS-B for origin shift refinement
- ✓ Incremental learning with δ = 0.1
- ✓ Various code optimizations

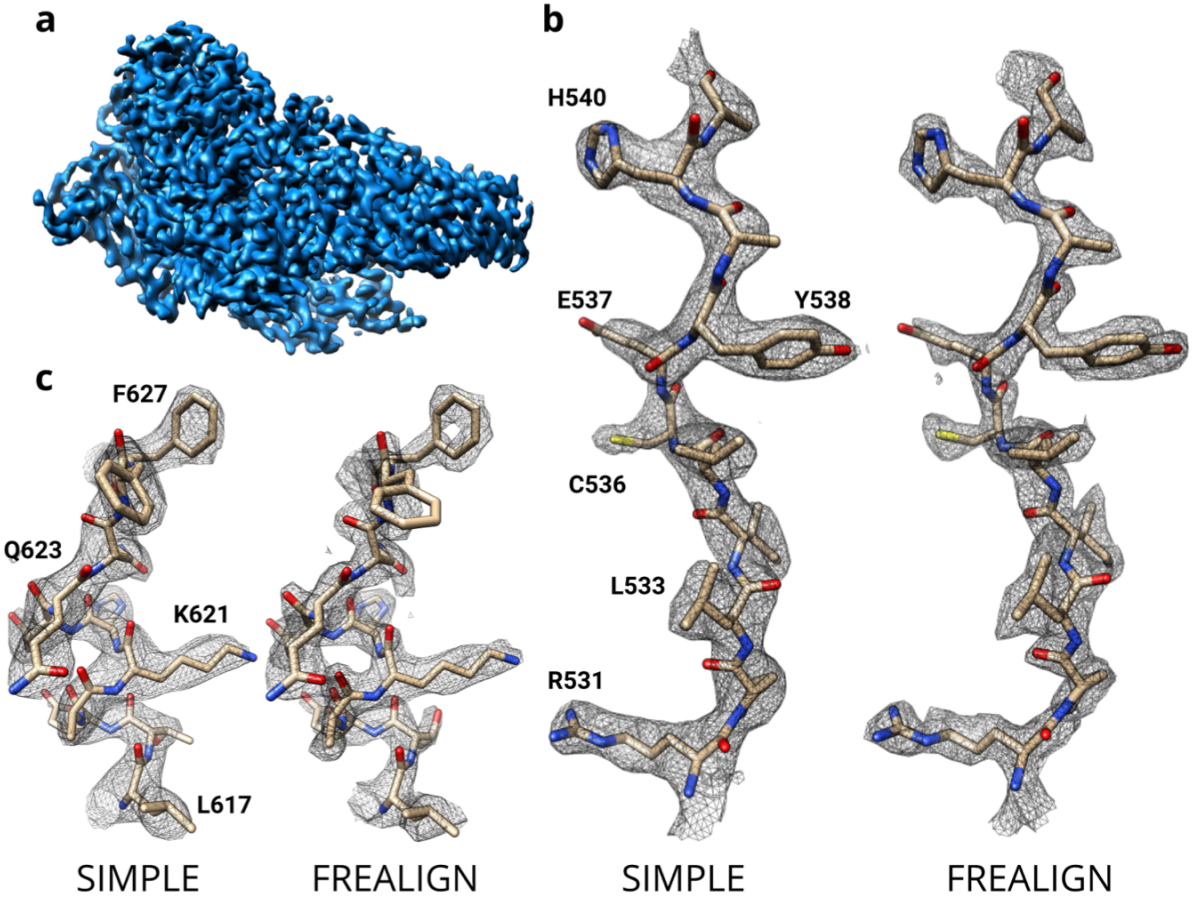
Total speedup with respect to Jan 2017 release: **86X**

$$\tau_{ij} = \frac{exp\{-(1-\lambda_{ij})/\sigma\}}{\sum_j exp\{-(1-\lambda_{ij})/\sigma\}}$$

$$\sigma = 0.005$$

Unpublished

Beta-galactosidase



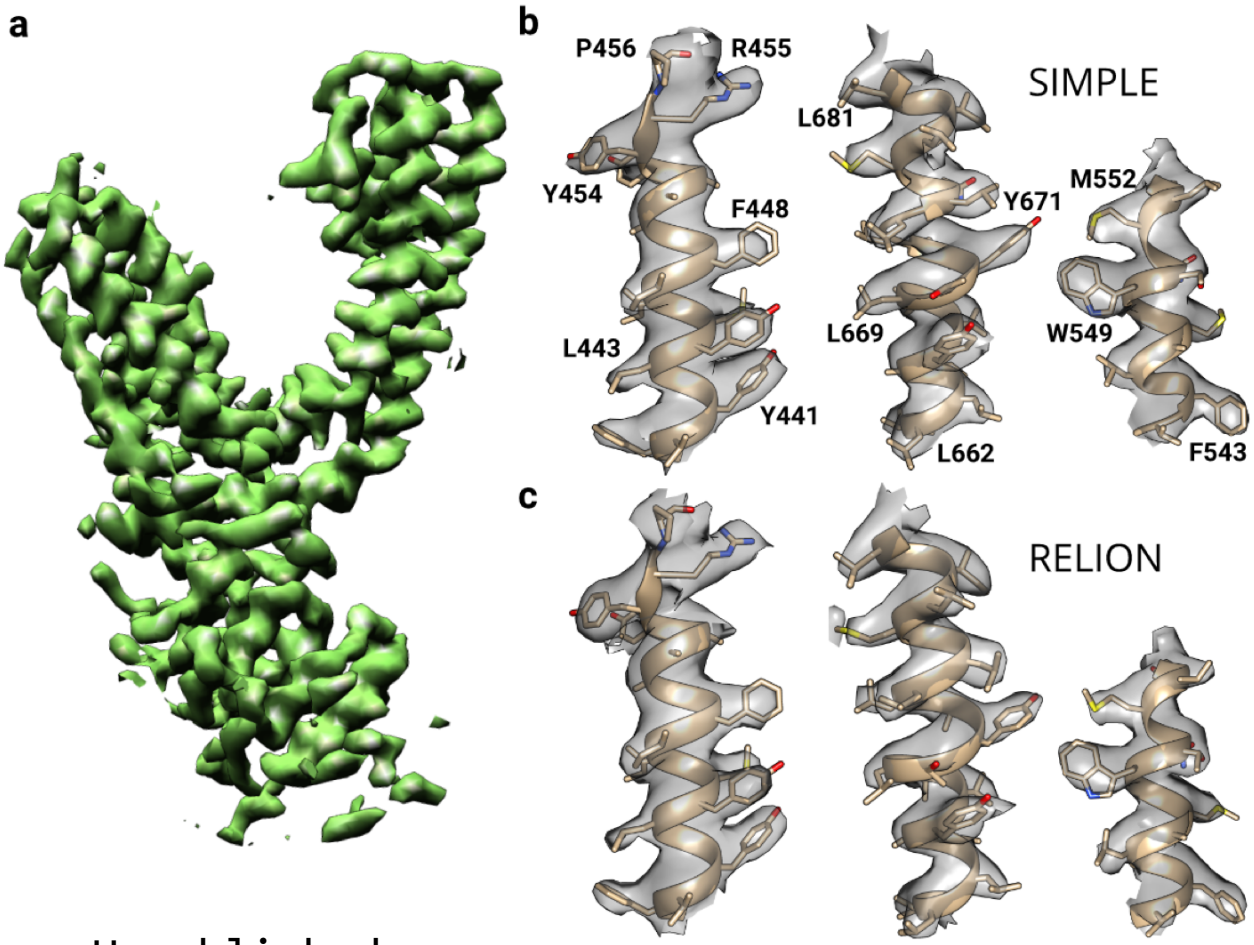
Unpublished

Data set: EMPIAR-10061

- ✓ Resolution @ 0.143 reported by SIMPLE: 2.8 A (”gold-standard”)
- ✓ Resolution @ 0.143 reported by FREALIGN: 2.2 A (not “gold-standard”)
- ✓ Maps have different character overall but very similar density in protein region
- ✓ No state sorting done in SIMPLE
- ✓ Structure built into SIMPLE map has better geometry than 5a1a.pdb

Beta-gal		5a1a.pdb	SIMPLE	
Protein Geometry	Statistic	(%)	(%)	Goal(%)
	Poor rotamers	1.0	0.1	<0.3
	Favoured rotamers	91.1	98.5	>98
	Ramachandran outliers	0.59	0	<0.05
	Ramachandran favoured	95.5	96.3	>98
	Bad bonds	0.1	0	0
	Bad angles	0.1	0	<0.1

TRPV1



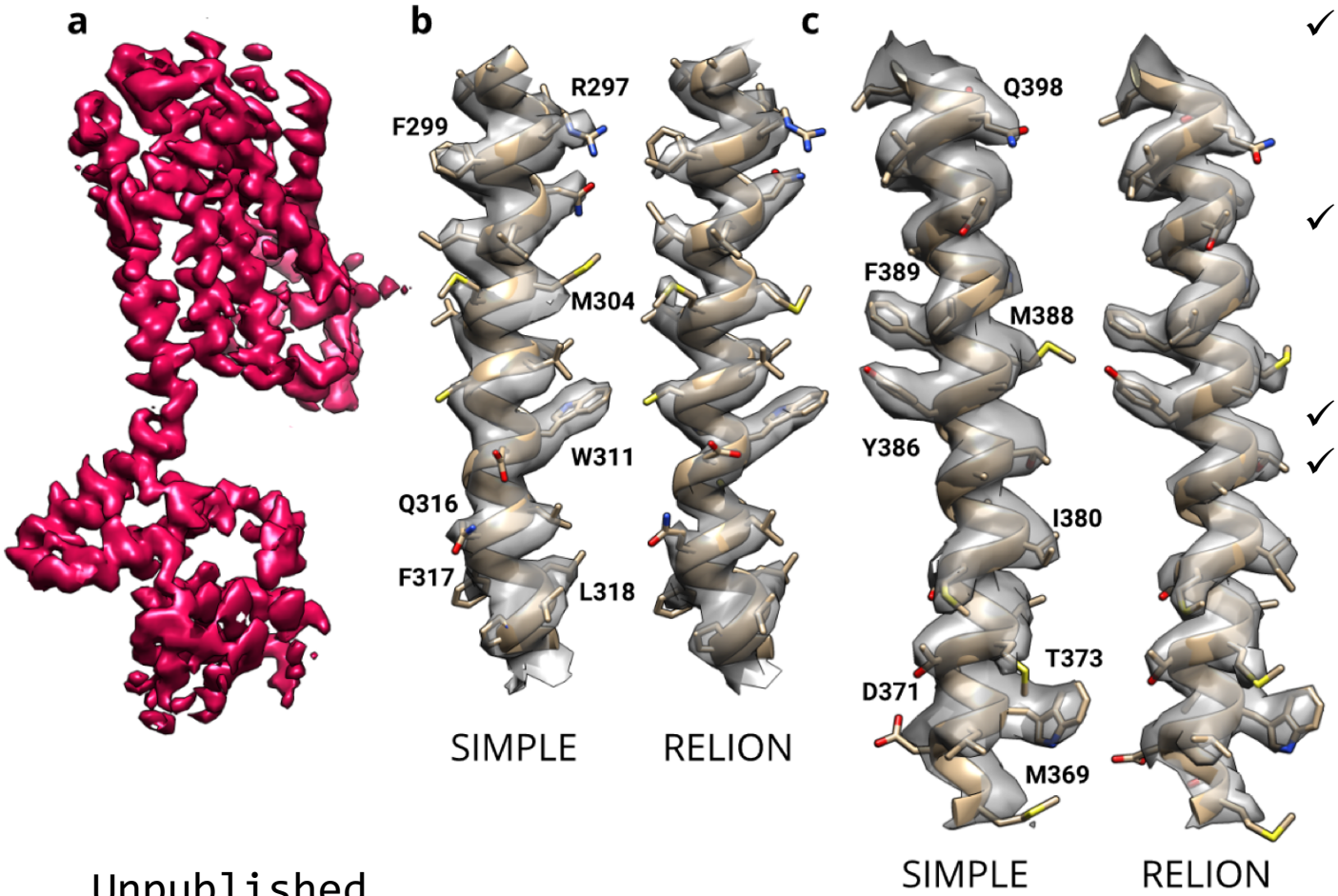
Data set: EMPIAR-10005

- ✓ Resolution @ 0.143 reported by SIMPLE: 3.6 A (”gold-standard”)
- ✓ Resolution @ 0.143 reported by RELION: 3.3 A (“gold-standard”)
- ✓ Maps are identical
- ✓ Structure built into SIMPLE map has better geometry than 3j5p.pdb

Unpublished

TRPV1		3j5p.pdb	SIMPLE	
Protein Geometry	Statistic	(%)	(%)	Goal(%)
	Poor rotamers	27.8	0.2	<0.3
	Favoured rotamers	55.4	96.7	>98
	Ramachandran outliers	0	0	<0.05
	Ramachandran favoured	94.3	94.2	>98
	Bad bonds	0.4	0	0
	Bad angles	0.1	0.1	<0.1

HCN



Data set: EMPIAR-10081

- ✓ Resolution @ 0.143 reported by SIMPLE: 3.4 A (”gold-standard”)
- ✓ Resolution @ 0.143 reported by RELION: 3.5 A (“gold-standard”)
- ✓ Maps are identical
- ✓ Structure built into SIMPLE map has same geometry stats as 5u6o.pdb

Unpublished

HCN		5u6o.pdb	SIMPLE	
Protein Geometry	Statistic	(%)	(%)	Goal(%)
	Poor rotamers	0	0	<0.3
	Favoured rotamers	99.1	97.7	>98
	Ramachandran outliers	0	0	<0.05
	Ramachandran favoured	95.4	95.4	>98
	Bad bonds	0	0	0
	Bad angles	0	0	<0.1

Key factors for enabling high-resolution 3D refinement with PRIME

- ✓ The Wiener restoration method used for Contrast Transfer Function (CTF) correction, similar to that in SPARX (Hohn et al., 2007).
- ✓ The method for preventing over-fitting, which combines two-fold cross-validated FSC (Scheres and Chen, 2012) with frequency limited refinement (Chen et al., 2013).
- ✓ The method for obtaining orientation weights from correlations, based on the parameterised softmax function

Acknowledgements

The other fearless leaders:
Dominika Elmlund, Monash
Susan Lea, Oxford

Postdocs/coders:
Cyril Reboul, Monash
Joe Ceasar, Oxford
Michael Eager, Monash
Simon Kiesewetter, Monash

Collaborators:
Hans De Sterck, MAXIMA, Monash
Matt Belousoff, Monash

Funding:
ARC
NHMRC
Monash