



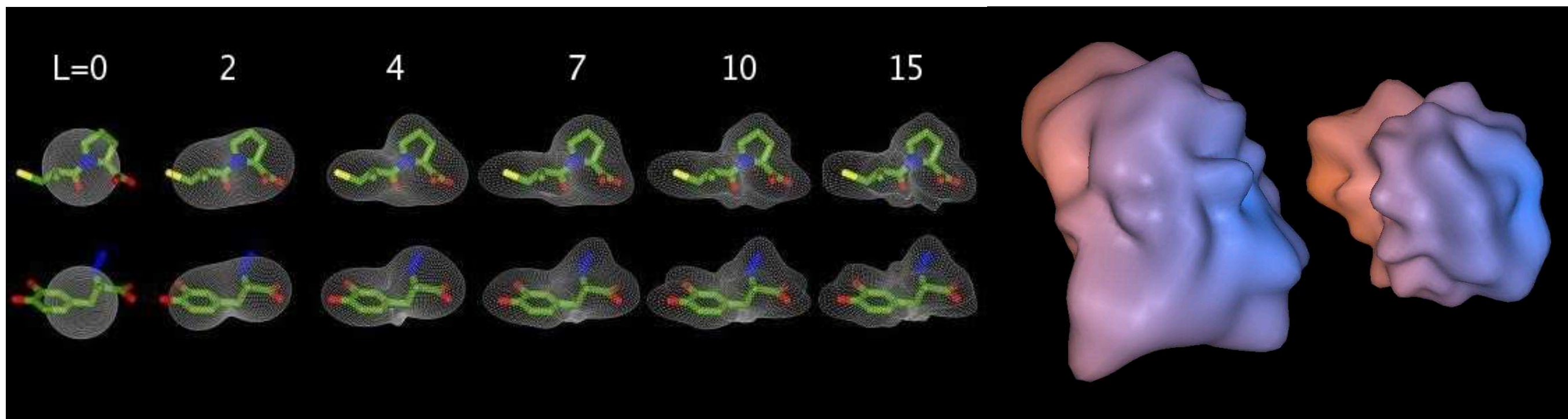
For the memory of Dave Ritchie

Surface Equations in 2D

Example: 2D radial functions (256 coefficients)

$$r(\theta, \phi) = \sum_{l=0}^{15} \sum_{m=-l}^l a_{lm} y_{lm}(\theta, \phi)$$

$$\int y_{lm}(\theta, \phi) y_{l'm'}(\theta, \phi) d\Omega = \delta_{mm'} \delta_{ll'} \quad a_{l,m} = \int r(\theta, \phi) y_{lm}(\theta, \phi) d\Omega$$



Ritchie & Kemp (1999) J Comp Chem 20 383–395

- Fits well for finding similarities between 2 surfaces
- Not suitable for docking!

ParaFit - High Throughput SH Surface & Property Matching

Distance: $D = \int (r_A(\theta, \phi) - r_B(\theta, \phi)')^2 d\Omega$

Orthogonality: $D = |\underline{a}|^2 + |\underline{b}|^2 - 2\underline{a}.\underline{b}'$

Rotation: $b'_{lm} = \sum_{m'} R_{mm'}^{(l)}(\alpha, \beta, \gamma) b_{lm'}$

Hodgkin: $S = 2\underline{a}.\underline{b}' / (|\underline{a}|^2 + |\underline{b}|^2)$

Carbo: $S = \underline{a}.\underline{b}' / (|\underline{a}|.\underline{b}|)$

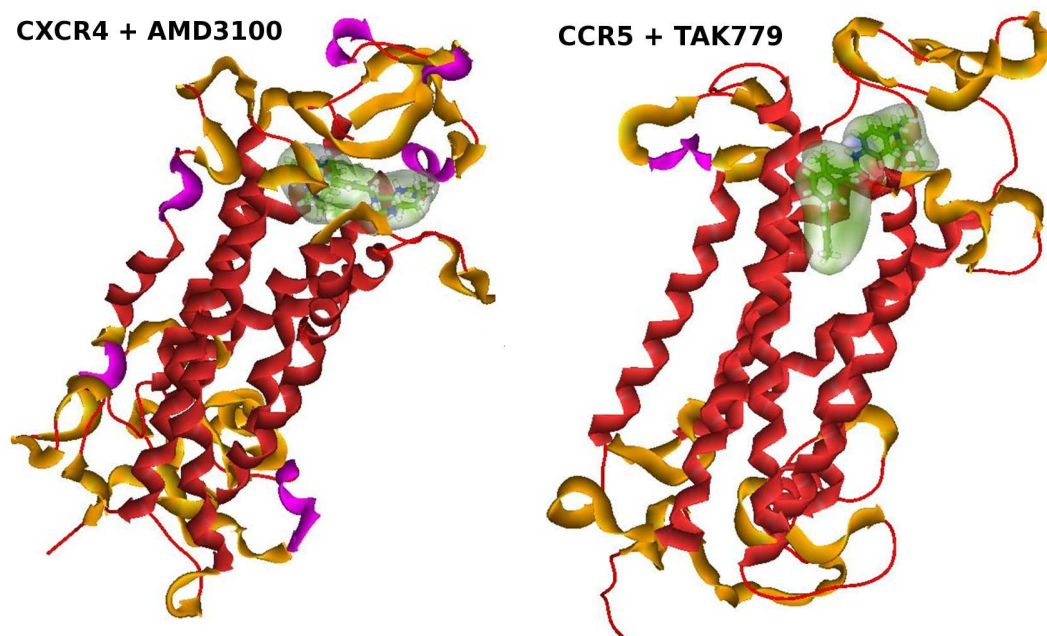
Tanimoto: $S = \underline{a}.\underline{b}' / (|\underline{a}|^2 + |\underline{b}|^2 - \underline{a}.\underline{b}')$

Multi-property: $S = pS^{\text{shape}} + qS^{\text{MEP}} + rS^{\text{IE}_L} + sS^{\text{EA}_L} + tS^{\alpha_L}$

Drug screening applications

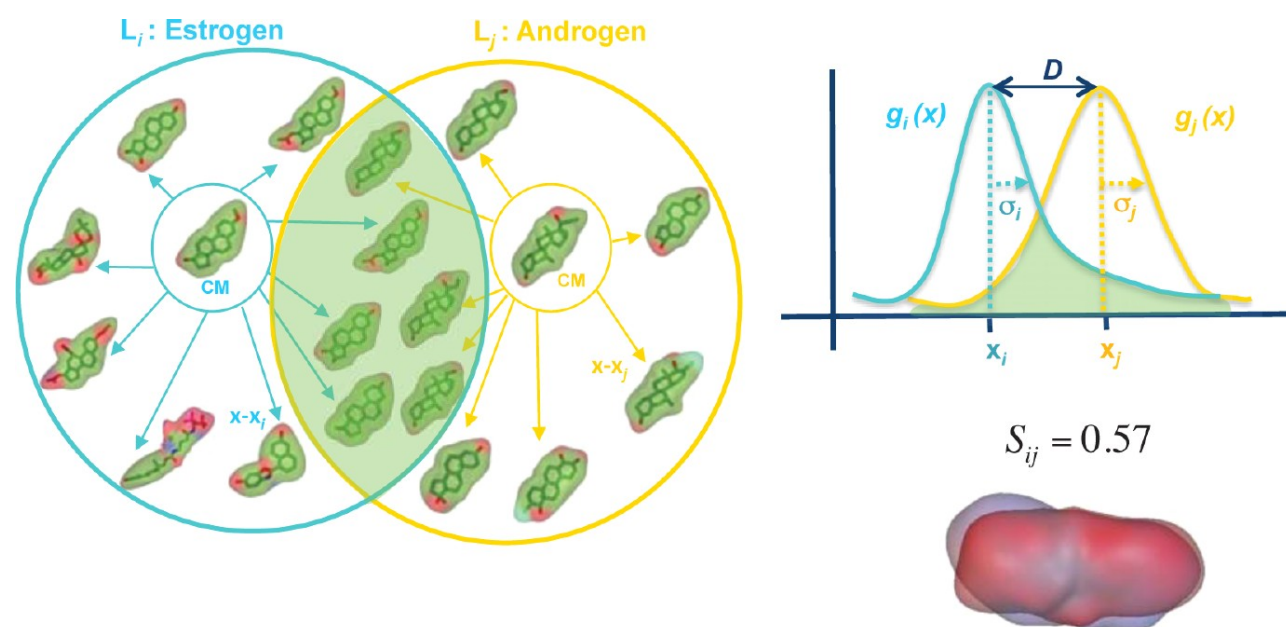
Shape-Based Virtual Screening

- Assembled 602 known actives against CXCR4 & CCR5
- Performed virtual screening against 4700 inactives



V. Perez-Nueno et al., (2008) J. Chem. Inf. Model.
48(3) 509-533

Gaussian Ensemble Screening

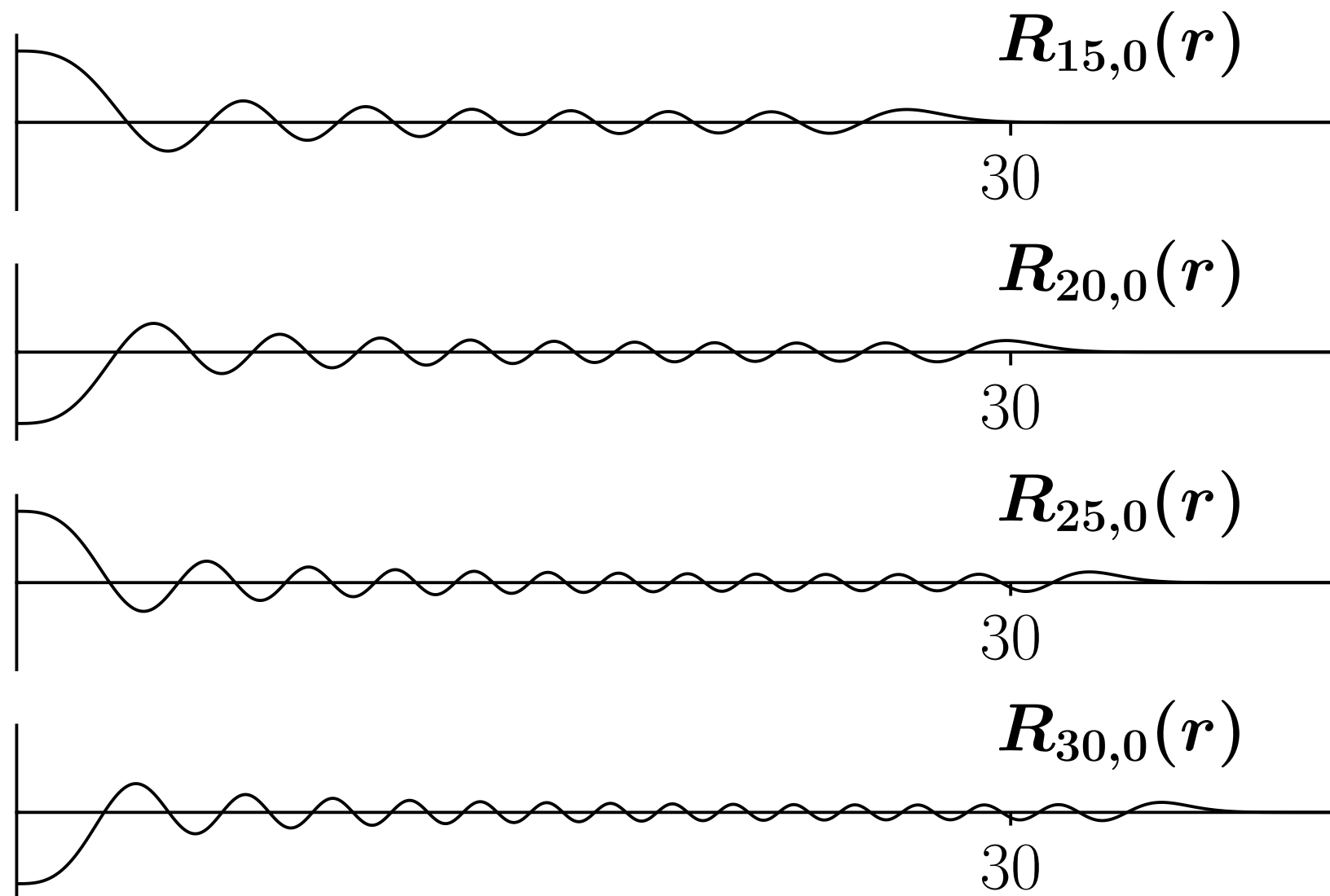


V. Perez-Nueno et al., (2012) J. Chem. Inf. Model.
52(8) 1948-1961

Shape Equations in 3D

GTO: shape $R_{nl}(r) = N_{nl}^{(q)} e^{-\rho/2} \rho^{l/2} L_{n-l-1}^{(l+1/2)}(\rho)$ $\rho = r^2/q$, $q = 20$

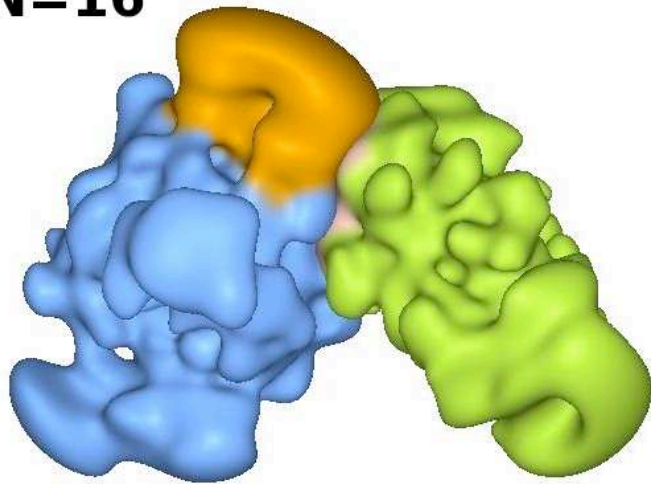
Orthogonality: $\int_0^\infty R_{nl}(r) R_{n'l}(r) r^2 dr = \delta_{nn'}$



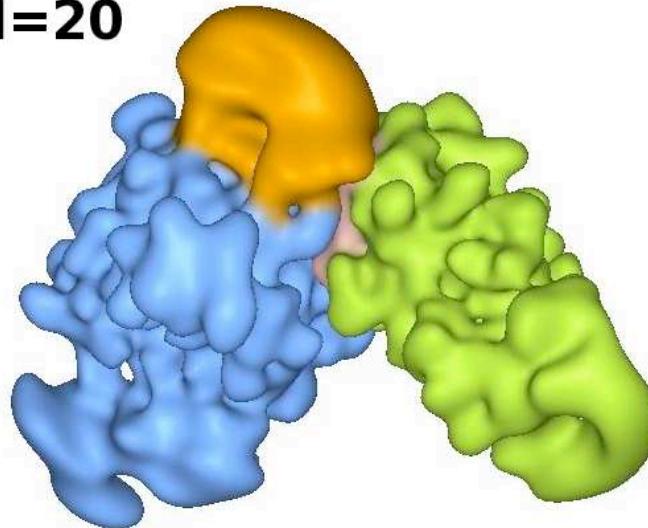
Ritchie & Kemp (2000) Proteins 39 178–194

3D Shape Reconstruction – CAPRI T21: Orc1/Sir1

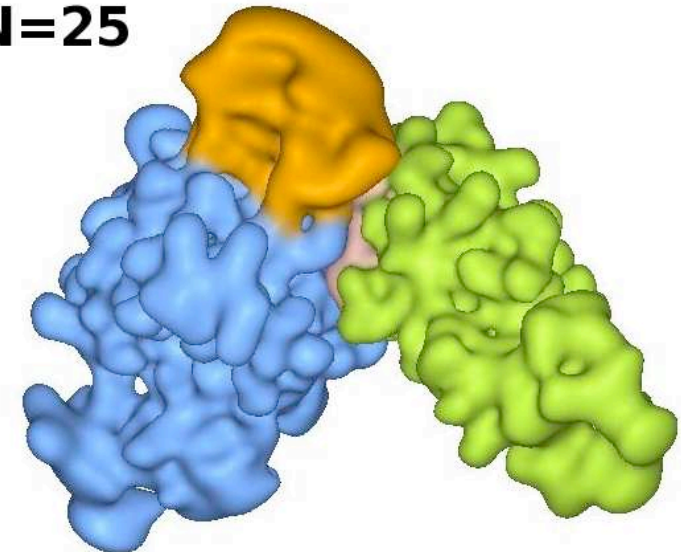
N=16



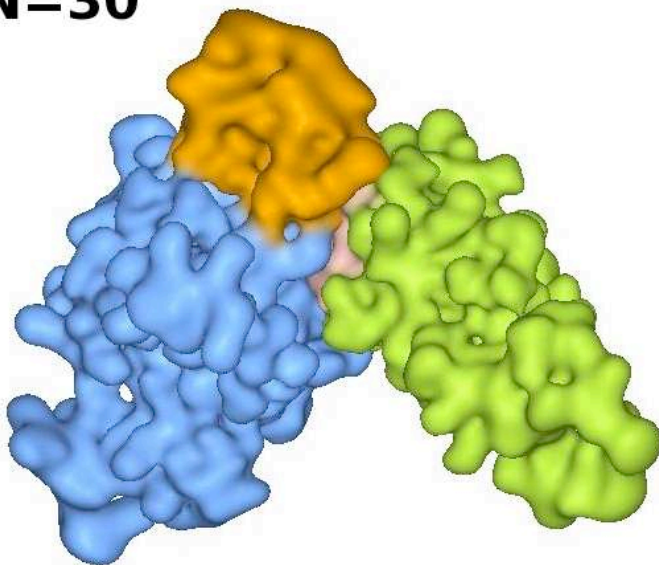
N=20



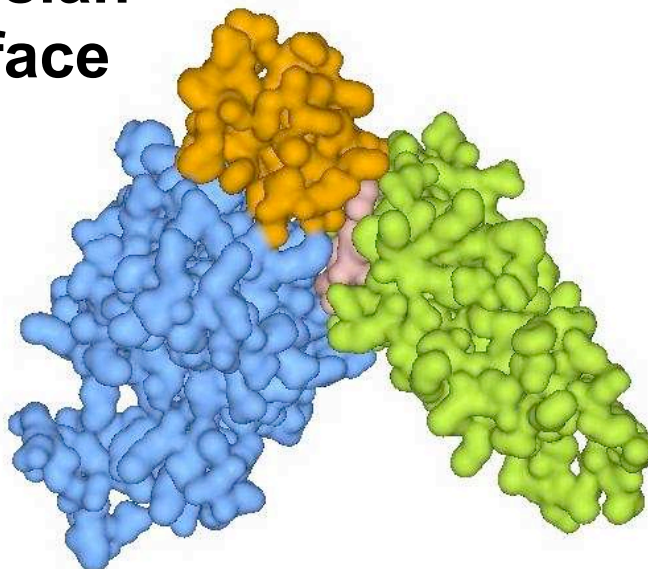
N=25



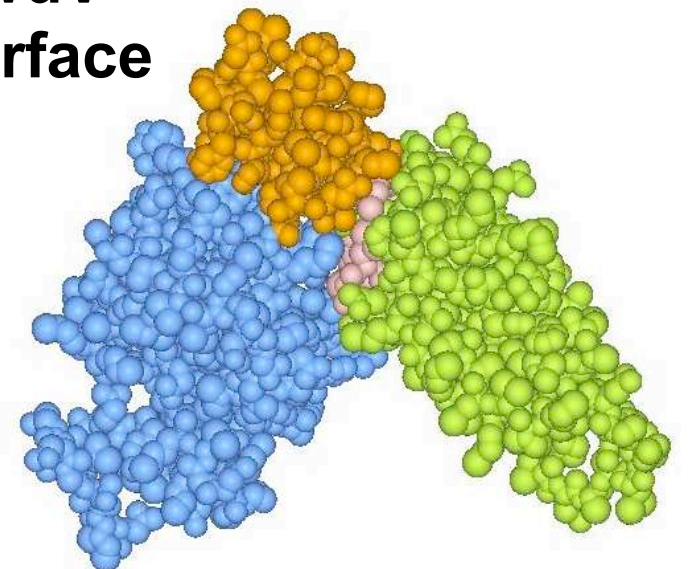
N=30



**Gaussian
surface**

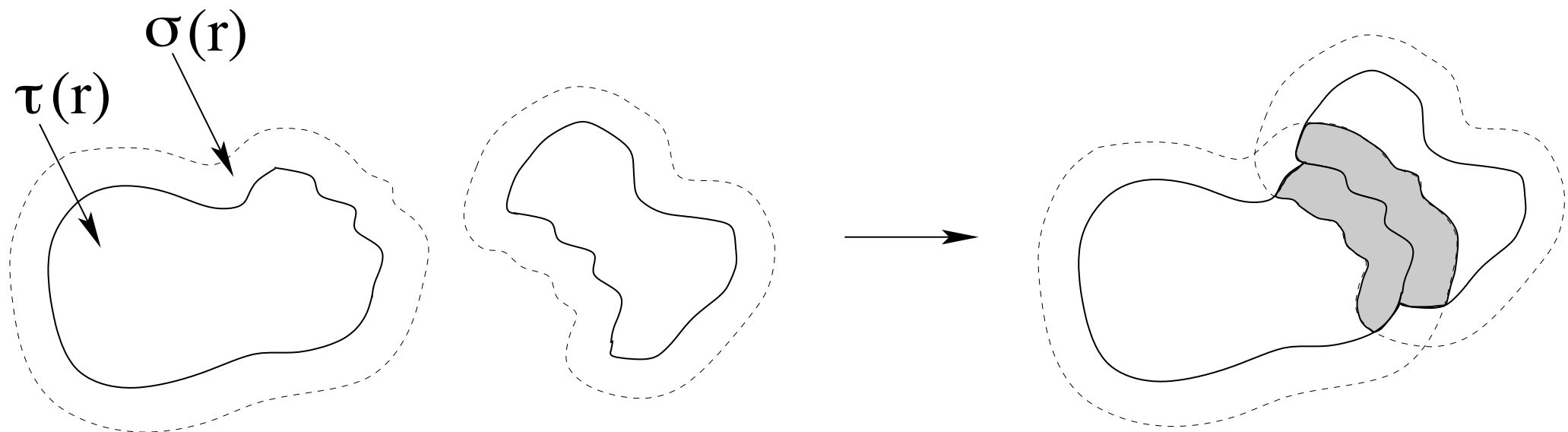


**wdV
surface**



DW Ritchie (2008) Curr. Prot. Pep. Sci. 9(1) 1-15

FFT-based Spherical Polar pair-wise docking



densities:

$$\sigma(\underline{r}) = \sum_{nlm}^N a_{nlm}^{\sigma} R_{nl}(r) y_{lm}(\theta, \phi) \quad \tau(\underline{r}) = \sum_{nlm}^N a_{nlm}^{\tau} R_{nl}(r) y_{lm}(\theta, \phi)$$

expansion
coefficients:

$$a_{n,l,m}^{\sigma} = \int \sigma(\underline{r}) R_{nl}(r) y_{lm}(\theta, \phi) dV$$

$$a_{n,l,m}^{\tau} = \int \tau(\underline{r}) R_{nl}(r) y_{lm}(\theta, \phi) dV$$

favorable:

$$\int (\sigma_A(\underline{r}_A) \tau_B(\underline{r}_B) + \tau_A(\underline{r}_A) \sigma_B(\underline{r}_B)) dV = \sum_{nlm} a_{n,l,m}^{\sigma} b_{n,l,m}^{\tau} + b_{n,l,m}^{\sigma} a_{n,l,m}^{\tau}$$

non-favorable:

$$\int \tau_A(\underline{r}_A) \tau_B(\underline{r}_B) dV = \sum_{nlm} a_{n,l,m}^{\tau} b_{n,l,m}^{\tau}$$

result:

$$S_{AB} = \int (\sigma_A \tau_B + \tau_A \sigma_B - Q \tau_A \tau_B) dV \quad \text{Penalty Factor: } Q = 11$$

Ritchie & Kemp (2000) Proteins 39 178–194

The rigid-body “docking equation”

“docking equation” – $A(\underline{x}) \longleftrightarrow B(\underline{x})$

translation operator – $\hat{T}(x, y, z)$ rotation operator – $\hat{R}(\alpha, \beta, \gamma)$

$$A(\underline{r}) \longleftrightarrow \hat{T}(x, y, z) \hat{R}(\alpha, \beta, \gamma) B(\underline{r})$$

The aim is to find six parameters $(x, y, z, \alpha, \beta, \gamma)$

such that the overlap integral, or the “docking score”

$$S = \int A(\underline{r})^* [\hat{T}(x, y, z) \hat{R}(\alpha, \beta, \gamma) B(\underline{r})] d\underline{r} \quad \text{is maximised}$$

$$A(\underline{r}) = \sum_{nlm} A_{nlm} R_{nl}(r) Y_{lm}(\theta, \phi) \quad B(\underline{r}) = \sum_{nlm} B_{nlm} R_{nl}(r) Y_{lm}(\theta, \phi)$$

rotation of coefficients :

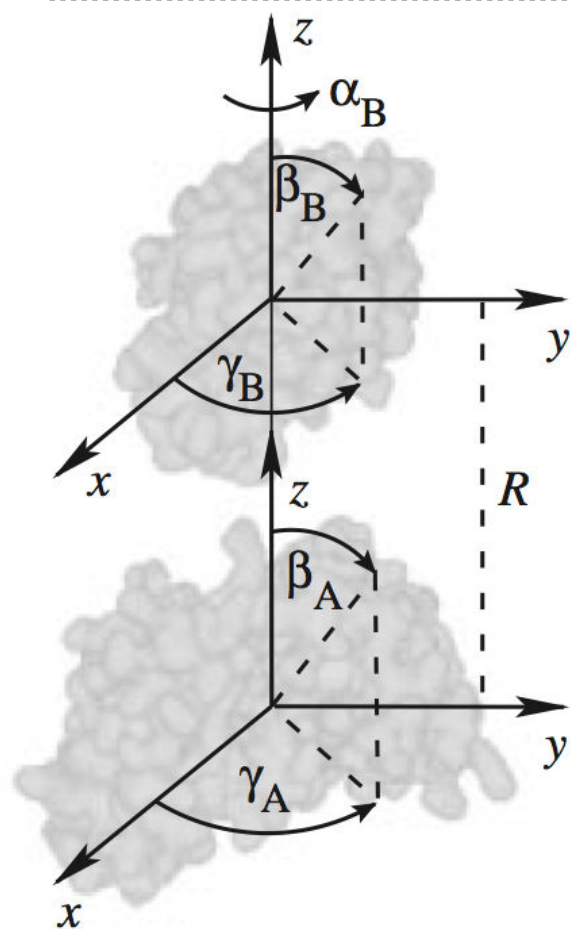
$$\hat{R}(\alpha, \beta, \gamma) B_{nlm} = \sum_{m'} D_{mm'}^{(l)}(\alpha, \beta, \gamma) B_{nlm'}$$

translation of coefficients :

$$\hat{T}(0, 0, z) B_{nlm} = \sum_{n'l'm} \sum_{n'l'} T_{nl, n'l'}^{(|m|)}(z) B_{n'l'm}$$

D.W. Ritchie. J. Appl. Cryst. 2005

The exploration of the search space



RB “docking score” :

$$S(r; \beta_A, \gamma_A, \alpha_B, \beta_B, \gamma_B)$$

1 translation and 5 rotations to adjust

$$R \in [0 : 1 : 40 \text{ \AA}]$$

discretised to enable exhaustive search

$$\alpha \in [0 : 7.5 : 360^\circ]$$

$$(\beta, \gamma) \in [0 : 7.5 : 180^\circ]$$

Truncated expressions for f and g using Polar Fourier correlation

$$S(r, \beta_A, \gamma_A, \beta_B, \gamma_B, \alpha_B) = \text{DFT}^{-1} [\text{DFT}(R(\beta_A, \gamma_A) \mathbf{f}_A) \cdot \text{DFT}(T_z(r) R(\beta_B, \gamma_B) \mathbf{g}_B)]_{\alpha_B}$$

$$S(r, \beta_A, \beta_B, \gamma_A, \gamma_B, \alpha_B) = \text{DFT}^{-1} [\text{DFT}(R(\beta_A) \mathbf{f}_A) \cdot \text{DFT}(T_z(r) R(\beta_B) \mathbf{g}_B)]_{\gamma_A, \gamma_B, \alpha_B}$$

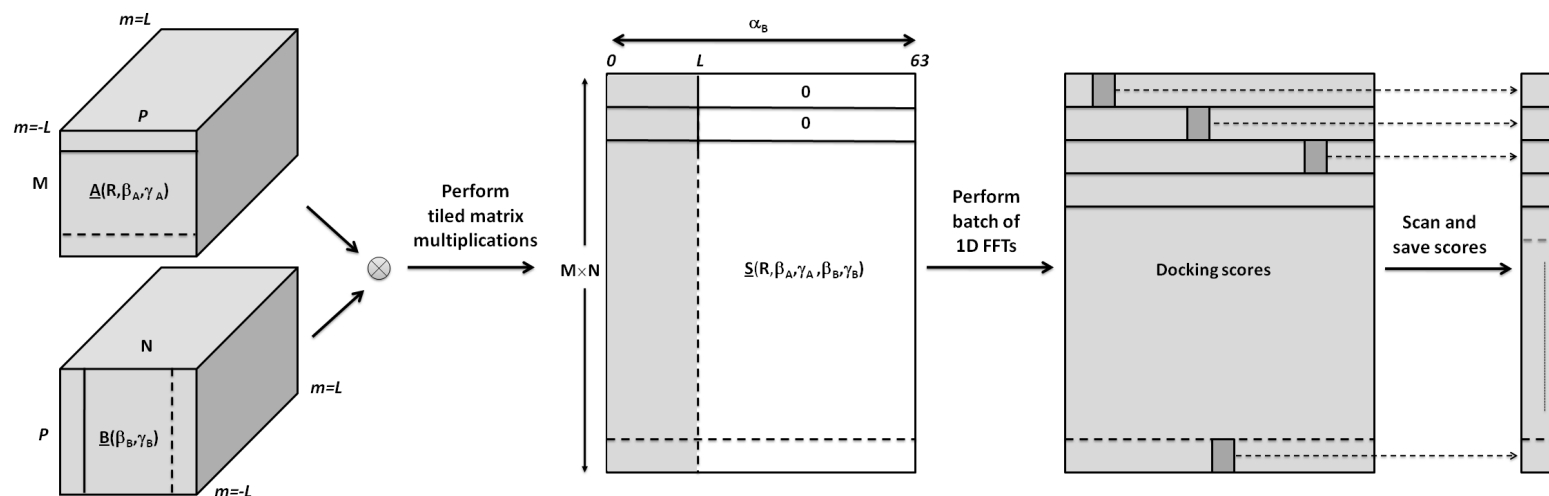
about 10^9 conformations in ~ 1 min on a laptop
equivalent to a $1 \mu\text{s}$ MD simulation $\sim$ days - months

GPU Implementation – Perform Multiple FFTs

- Calculate multiple 1D FFTs of the form:

$$S_{AB}(\alpha_B) = \sum_m e^{-im\alpha_B} \sum_{nl} A_{nlm}^\sigma(R, \beta_A, \gamma_A) \times B_{nlm}^\tau(\beta_B, \gamma_B)$$

- Cross-multiply transformed A with rotated B coefficients
- Perform batch of 1D FFTs using cuFFT and save best orientations



- 3D FFTs in $(\alpha_B, \beta_B, \gamma_B)$ can be calculated in a similar way...

D.W. Ritchie, V. Venkatraman (2010). Bioinformatics, 26, 2398-2405

SAM – Symmetry Assembler

Docking :

$$\hat{T}_z(D) \hat{R}(\alpha, \beta, \gamma) A(\underline{r}) \longleftrightarrow \hat{R}_y(\omega) \hat{T}_z(D) \hat{R}(\alpha, \beta, \gamma) A(\underline{r})$$

↑ ↑ ↑ ↑
free parameters

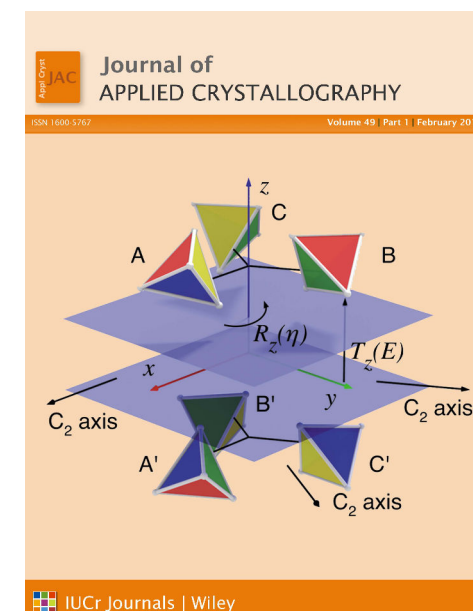
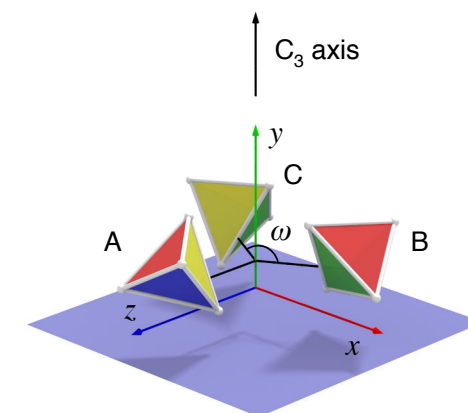
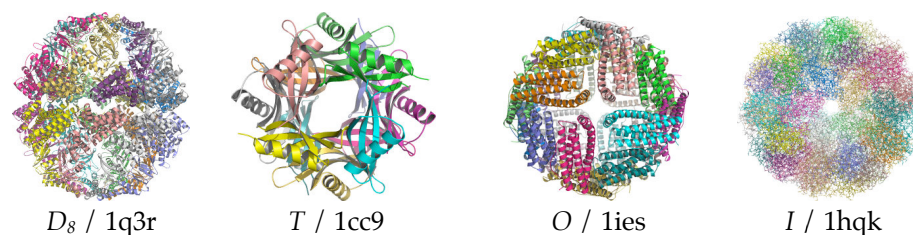
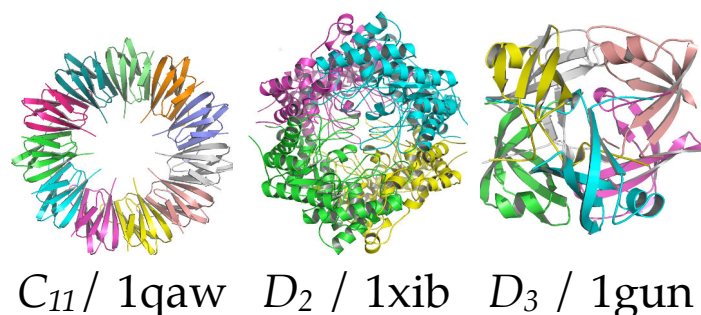
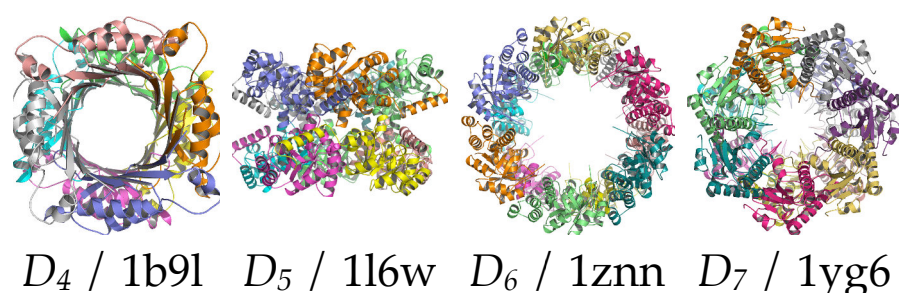
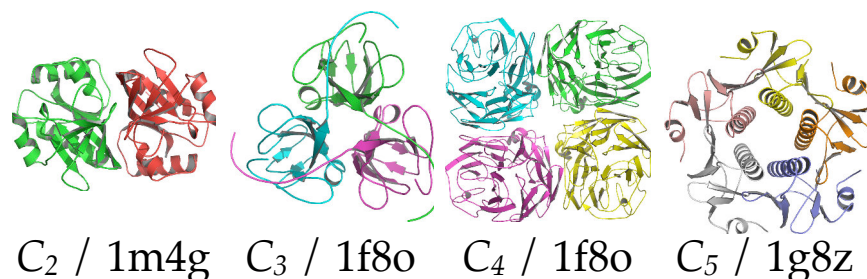
↑
fixed

↑ ↑ ↑ ↑
free parameters

maximizes the
“docking score”

$$S(\omega; D; \alpha, \beta, \gamma) = \sum_{nlmp} e^{-i(p-m)\alpha} d_{mp}^{(l)}(\omega) A'_{nlp} A''_{nlm}^*$$

- 1D FFT in α
- takes ~20 seconds on a laptop for C_n groups



D.W. Ritchie and S. Grudinin. J. Appl. Cryst. 2016

The future of FFT-based shape matching / docking

- New polynomial expansions and interconversions between them

$$H_n(\rho) = (-1)^n e^{\rho^2} \frac{d^n}{d\rho^n} e^{-\rho^2} \quad T_{nm}(\Delta) = N_n N_m \pi^{1/2} e^{-\Delta^2/4} \sum_{k=0}^{\min(n,m)} (-1)^{m-k} 2^k k! \binom{n}{k} \binom{m}{k} \Delta^{n+m-2k}.$$

- New stable relations for polynomials of very high order

$$(j_1 - m_1)(j_2 + m_2) G \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} =$$

$$[j_3(j_3 + 1) - j_2(j_2 + 1) - j_1(j_1 + 1) - 2(m_1 + 1)(m_2 - 1)] G \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 + 1 & m_2 - 1 & m_3 \end{pmatrix}$$

$$- (j_1 + m_1 + 2)(j_2 - m_2 + 2) G \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 + 2 & m_2 - 2 & m_3 \end{pmatrix}.$$

Ritchie, D., 2018. Whole Number Recursion Formulae for High Order Clebsch-Gordan Coupling Coefficients.

- New spherical-Bessel-based bases sets for FFT-based docking

$$B_{lm}^{(n)}(\underline{r}) = \sqrt{\frac{2}{\pi}} \beta_n j_l(\beta_n r) y_{lm}(\theta, \phi)$$

Ritchie, D., 2019. High Angular Resolution FFT Protein Docking Using Spherical Wave Basis Functions.

- FFT-based docking with evolution grids, new orientation-independent shape invariants, FFT-based docking of flexible shapes, exhaustive FFT-based search of protein domains and many more!

In my mailbox I've found 1,840 e-mails sent to Dave...

Jan 23, 2012:
Hi Sergei,

I am trying to see how to exploit symmetry in FFTs etc. I think you know more about this than me?

I found this old paper by Lynn Ten Eyck. Is this something like what you already have? I am still trying to understand the basic concepts ("reciprocal space", "Miller indices", ...). I think I "understand" the low level steps, but I don't yet really "see" it all well enough to be able to use it properly.

Cheers,

Dave

Feb 15, 2012:
Hi Dave,

thank you! I switched to the proposed notation for spatial transform operators and everything is getting cleaner! Could you please look at the last "docking equation" and say if it can advance further?

Cheers,
Sergei

On 10/25/2012 05:49 PM, Sergei Grudinin wrote:
[Hi Dave,](#)

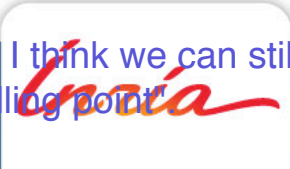
[I've looked into our symmetry equations. It's possible to express it as either 1D or 2D translational correlation. For rotational correlations, it's NOT possible to choose the twist angle as the correlation coordinate. It can be separated in a combination with some other rotations, which do not lead to the Fourier correlation, I'll write it in a more detail later. Hopefully, we will find a different generalised rotational coordinate.](#)

[Cheers,](#)
[Sergei](#)

On Oct 26, 2012, at 9:32 PM, Dave Ritchie <Dave.Ritchie@inria.fr> wrote:
[Hi Sergei,](#)

OK, I am convinced. For C_n symmetry, to generate symmetry-related poses, a rotation must be applied to each monomer before a translation (as we had on the white-board). I suppose it follows that this also applies to D_n . Furthermore, I believe we must apply a 3D rotation to each monomer before doing a translation (i.e. it is not possible to "delay" applying one rotation). This leaves just one translational degree of freedom which must be applied last. In other words: I agree with you that the only possible FFT dimension is perpendicular to the principal rotational symmetry axis of C_n or D_n .

[But I think we can still do the whole calculation efficiently in a single basis \(Hermite or Gauss-Laguerre\). So this could still be our computational advantage and "selling point"](#)



Sergei Grudinin

On Nov 1, 2012, at 8:57 PM, Dave Ritchie <Dave.Ritchie@inria.fr> wrote:
Hi Sergei,

I think I have cracked it. I think I can expose a 3D rotational FFT with one z-translation. Or equally a series of 1D FFTs for better GPU performance.

I was making a few edits to your equations to try to follow the Euler convention more closely when it hit me. Your working on the last section reminded me of the trick about factoring y-rotations in polar coordinates. And your final conclusion is correct. Basically, to go any further one has to write out our famous starting equation (your equation 13 or similar) and then expand everything as complex exponentials and collect like terms. Apart from some basic "house-keeping", it should be straight-forward. I will try to write it all out properly tomorrow.

Dave

On Dec 30, 2012, at 2:30 PM, Dave Ritchie <Dave.Ritchie@loria.fr> wrote:
Hi Sergei,

I have been doing some experiments to simulate the cost of FFT-based symmetry docking. Even the fastest method will take several CPU-hours per symmetry type. **So basically, I do not think that symmetry-constrained FFTs will be feasible.** On the other hand, I think it will be very straight-forward to do unconstrained FFT docking using Hex (or any other program for that matter), and then applying a symmetry filter to the solutions.

Please see sections 5.2, 5.5, and especially 5.6, and let me know what you think?

Cheers,

Dave

On 02/01/13 10:05, Dave Ritchie wrote:
Hi Sergei,

That looks great! **You have cracked it!**

Doing the explicit translation of coefficients is much faster than the reduced translation/overlap matrix. My old machine can do about 60 translations/second/cpu. So now we should be looking at a matter of minutes for the whole FFT calculation!

Cheers

Dave

On Jan 2, 2013, at 5:04 PM, Dave Ritchie <Dave.Ritchie@loria.fr> wrote:
Hi Sergei,

Sorry, **I am starting to have a doubt.**

When we did the original factorisation with beta1 and beta2, there was no requirement that the initial translation was along the z axis, even though we set it up that way.

The factorisation started from

$$T(x,y,z)^{-1}.Ry(\omega).T(x,y,z) = R2.Tz(D).R1$$

The next argument was that because we are in the xz plane, this can be reduced to

$$T(0,0,z)^{-1}.Ry(\omega).T(0,0,z) = Ry(\beta_2).Tz(D).Ry(\beta_1).$$

Now, since we are still working in the xz plane, my doubt is whether it is really possible to solve for x and z which can give

$$T(x,0,z)^{-1}.Ry(\omega).T(x,0,z) = Ry(\omega).Tz(D).Ry(0).$$

...loosely speaking, we add one degree of freedom, and expect it to satisfy two criteria (i.e. $\beta_2=\omega$ and $\beta_1=0$).

I tried writing out the simultaneous equations, but I cannot get a solution.

What do you think? If we are to use the "new alpha" method, we have to be able place the FFT solutions back in the global coordinate system - i.e. we have to be able to solve for x, and z above?

Cheers,

Dave

On Jan 2, 2013, at 8:02 PM, Dave Ritchie <Dave.Ritchie@loria.fr> wrote:
Hi Sergei,

Please forget my previous e-mail. It proves nothing!

On the other hand, I still think the new alpha method is good. Here is a much simpler development. We start from:

$$Ry(\omega).Tz(\delta).R(\alpha,\beta,\gamma).B \leftrightarrow Tz(\delta).R(\alpha,\beta,\gamma).A$$

Pull the alpha rotations to the left because they commute with Tz

$$Ry(\omega).Rz(\alpha).Tz(\delta).R(0,\beta,\gamma).B \leftrightarrow Rz(\alpha).Tz(\delta).R(0,\beta,\gamma).A$$

Multiply both sides by $Rz(\alpha)^{-1}$

$$Rz(\alpha)^{-1}.Ry(\omega).Rz(\alpha).Tz(\delta).R(0,\beta,\gamma).B \leftrightarrow Rz(\alpha).Tz(\delta).R(0,\beta,\gamma).A$$

Substitute for rotated/translated A and B (which are of course equal)

$$Rz(\alpha)^{-1}.Ry(\omega).Rz(\alpha).Tz(\delta).B' \leftrightarrow Tz(\delta).A'$$

Proceed with new alpha route.... :-)

Dave

Hi Dave,

I've shot a trimer :) Top view (viewer 1, CCW):



Bottom view (viewer 2, CW):



Now, if I apply a π alpha rotation I obtain the opposite arrangement (but the same trimer!). Attention, my rotation axis connects the geometrical centre with the centre of the first monomer!:

Top view (viewer 1, CW):



Bottom view (viewer 2, CCW):



The relative trimer's arrangement is the same after changing the alpha angle by π . Do I miss something?

Sergei

Hi Dave,

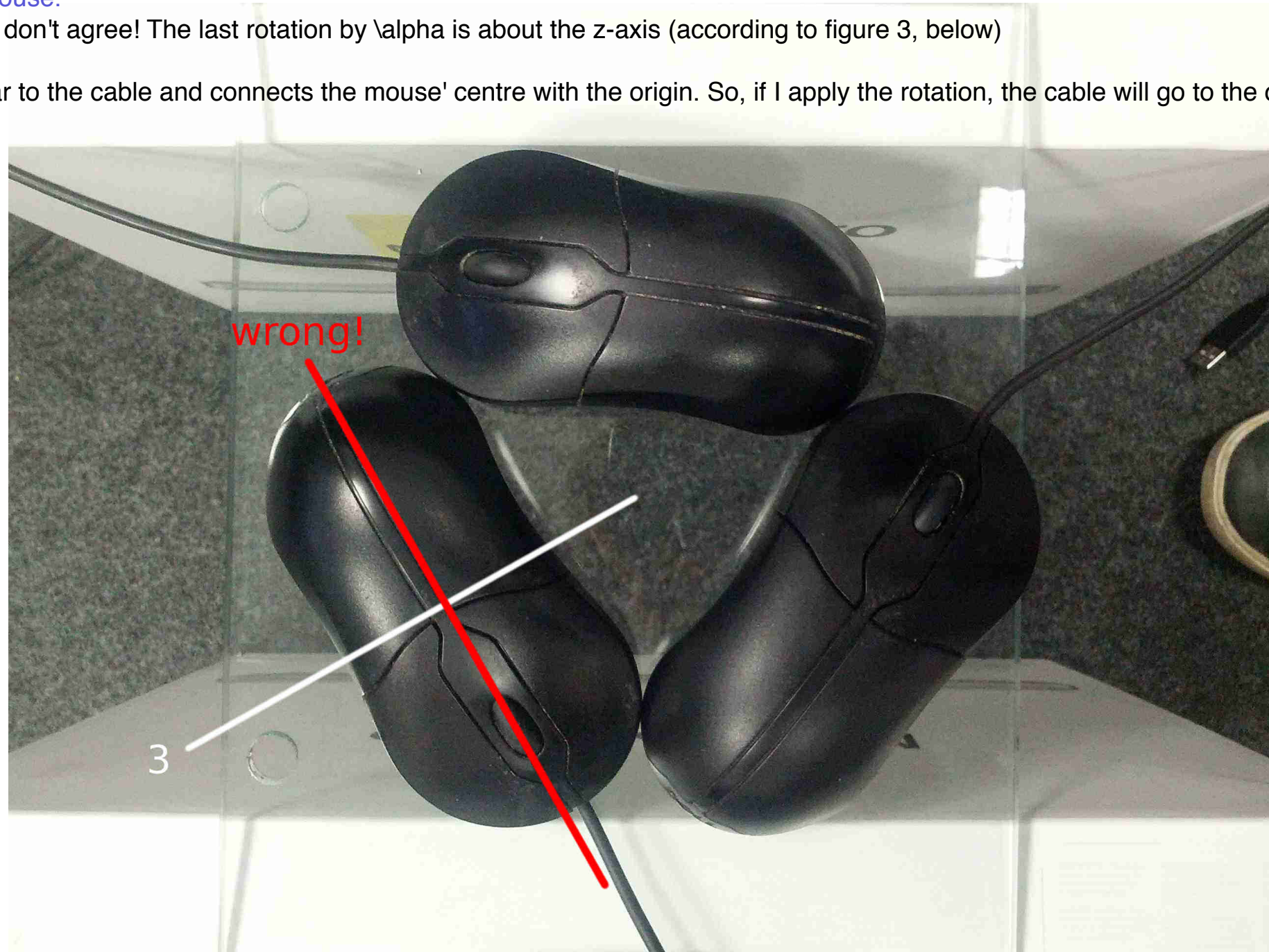
Hi Sergei,

Nice pictures! But I can see that you only took two photos to save time! ;-)
The pictures from the bottom were of a very bad quality, but I did them too!

In picture #1, the cable of the top mouse goes to the top left corner. If you rotate the mouse by π about its axis, the cable will still go to the top left but you will see the underside of the mouse.

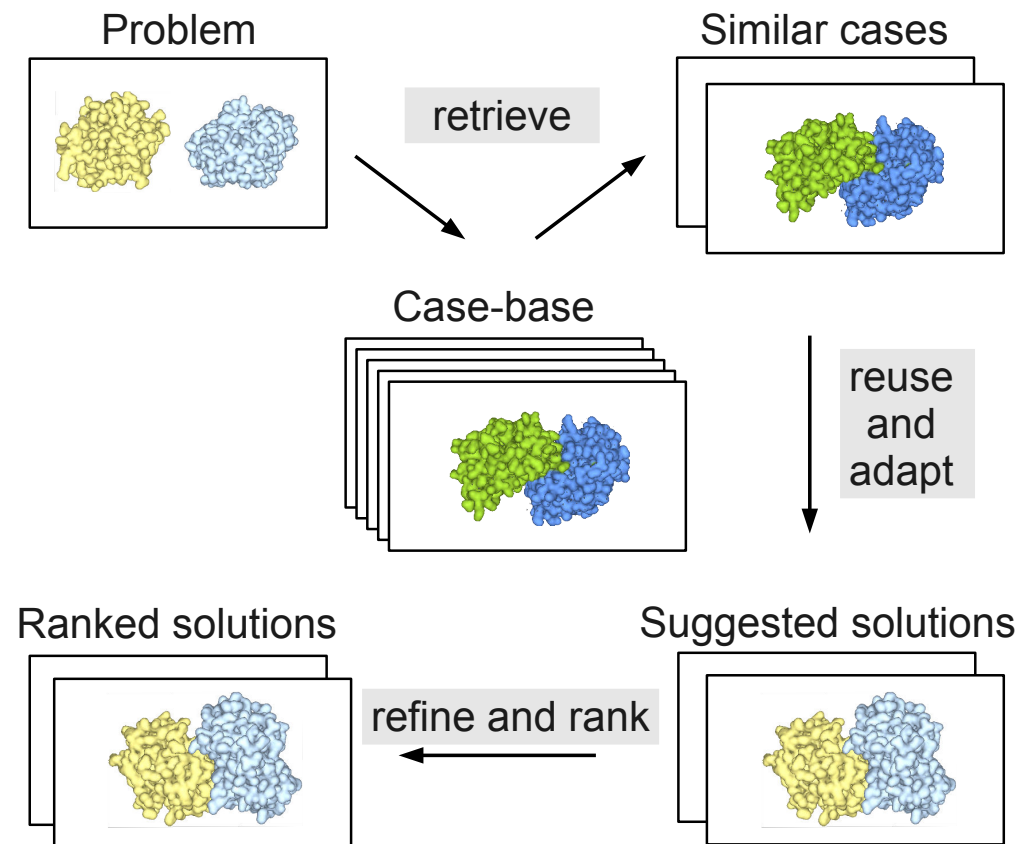
OK, now I see why we don't agree! The last rotation by α is about the z-axis (according to figure 3, below)

, which is perpendicular to the cable and connects the mouse' centre with the origin. So, if I apply the rotation, the cable will go to the opposite direction!



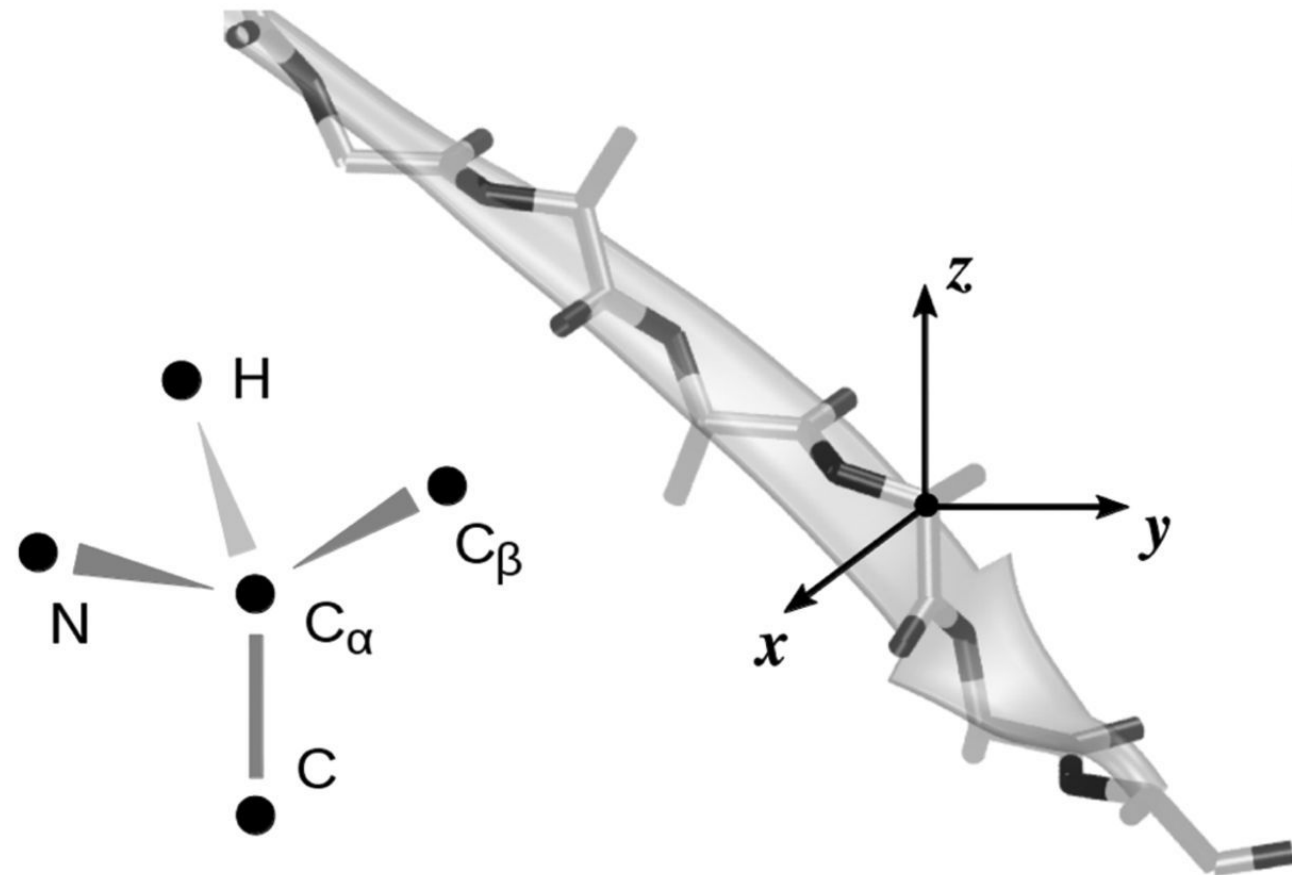
FFT-free shape matching

- Case-based reasoning for 3D protein complexes



A.W. Ghoorah et al. *Bioinformatics*, 27:20, 2820-2827 (2011)

- A fast protein structure alignment and database search program
- Extended to flexible structure alignments, and multiple structure alignments



D.W. Ritchie et al., *Bioinformatics*, 28:24, 3274–3281 (2012)
D.W. Ritchie, *Bioinformatics*, 32:17, 2650–2658 (2016)

EROS-DOCK : protein–protein docking using exhaustive branch-and-bound rotational search

Maria Elisa Ruiz Echartea et al, *Bioinformatics*, 2019, btz434, <https://doi.org/10.1093/bioinformatics/btz434>