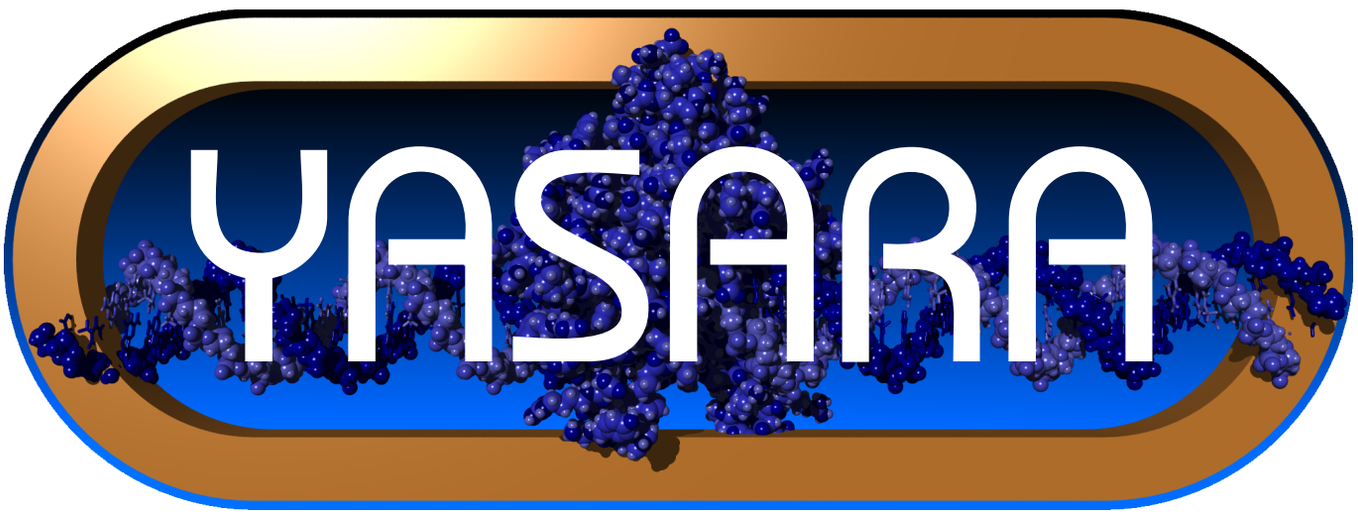



# Science Manual

Elmar Krieger

Cover: YAMI, Yet Another Mascotti, a truly hypothetical protein.


1st edition 2004, original PhD thesis "The last mile of the protein folding problem - A pilgrim's staff and skid-proof boots" (ISBN 90-9018508-9).
2nd edition 2010, with hard-cover and updated references.
3rd edition 2017, including additional articles published since 2004, renamed to "YASARA Science Manual", new DINA4 layout created by Dipl.-Ing. Kornel Ozvoldik using LibreOffice 5.2.3.3 and PyPDF2, printed by Frick Digitaldruck, www.online-druck.biz.

Since the first line of YASARA source code was written back in 1993 on an Intel 386SX PC with 0.02 GHz clock frequency and 0.002 GB RAM, about $1.4*10^{15}$ liters of water flowed down the Danube river here in Vienna on their way to the Black Sea – and 150000 lines of YASARA source code flowed right into /dev/null – the Linux equivalent of a black hole. This is the direct consequence of the compulsion to steadily improve YASARA's feature set, accuracy, performance and usability. I would like to thank you a lot for choosing YASARA and helping to keep the source flowing, and I hope that YASARA will be an equally big help for your own research.

Having started Thanksgiving, this page is an excellent place to acknowledge YASARA's supervisory board in chronological order: Prof. Günther Koraimann and Prof. Andreas Kungl paved its way in the early days at the University of Graz in Austria, by turning YASARA version 0.99 for MSDOS into my master's thesis in 1999. In 2000, Prof. Gert Vriend, head of the Center for Molecular and Biomolecular Informatics (CMBI) at the University of Nijmegen in the Netherlands, kindly opted to become YASARA's patron. As the author of the all-time classic WHAT IF molecular modeling program (cited 5500 times), he knew exactly how to let modeling software flourish, and consequently my feelings were rather mixed when I submitted the 1st edition of this Science Manual as my PhD thesis in 2004, ending my time as Gert's PhD student in Nijmegen. I am also indebted to the members of the PhD committee for bravely reading the oevre: Prof. Herman J.C Berendsen, Prof. Jaap Heringa and Prof. Henk Stunnenberg. Back in Austria, the ties with the CMBI remained strong, partly thanks to the WHAT IF/YASARA joint distribution - the Twinset, partly thanks to the great students at the CMBI who worked with and on YASARA, but mostly because of Gert's reluctance to stop the continuous help. 1000 thanks Gert!

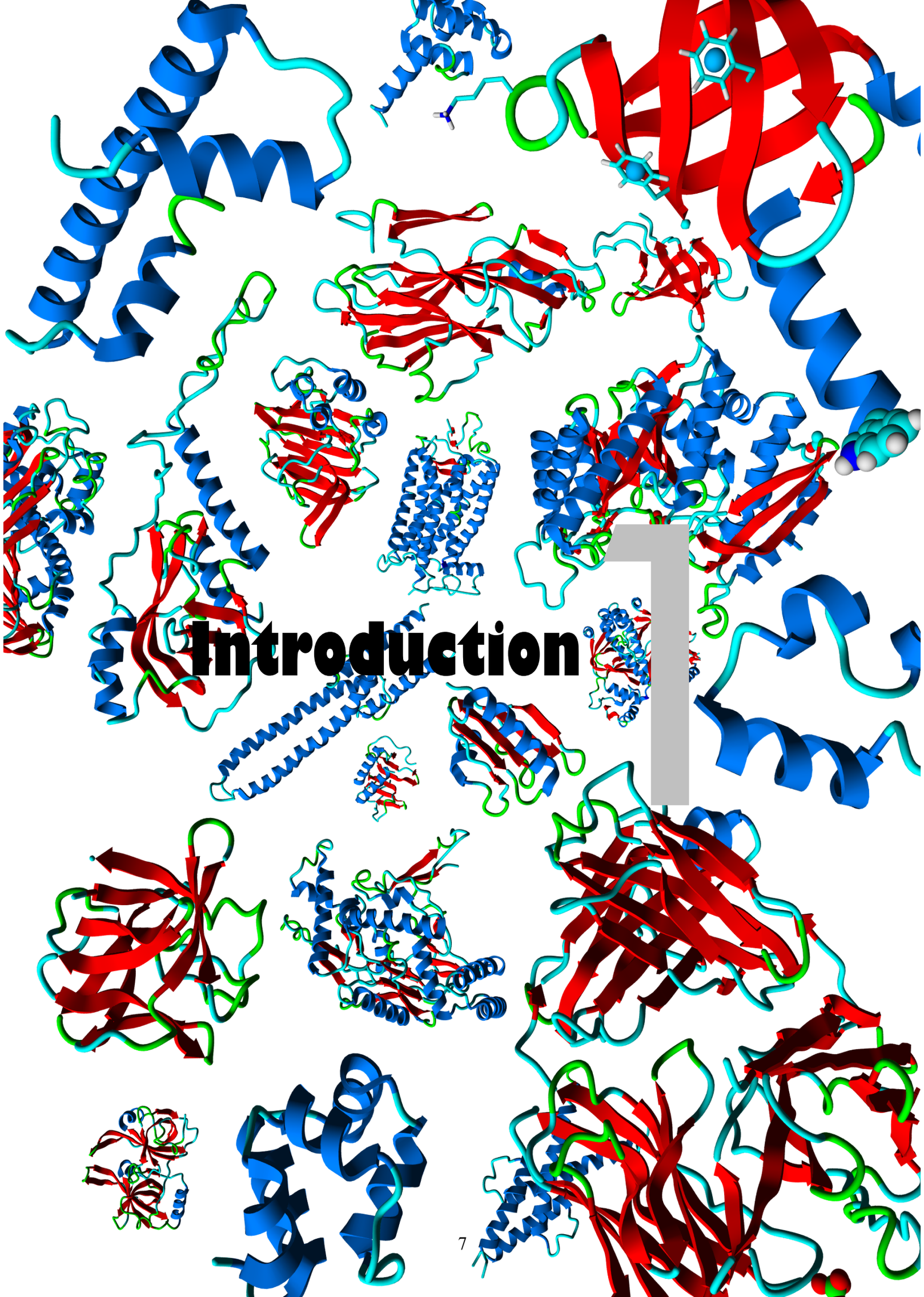
Last but not least I would like to thank YASARA's COO Dipl.-Ing. Kornel Ozvoldik for layouting this manual, and I want to praise all YASARA users for the endless flow of great ideas and extremely helpful feedback. The list of YASARA's heroes who contributed one way or another can be found in the Help menu at 'About YASARA'.

I hope that you will enjoy working with YASARA, and please don't hesitate to send your feedback...

With kind regards,

# Table of Contents

# The force fields



# 1 Introduction

# About this manual

This book describes YASARA's scientific background, it contains all the articles we published about or using YASARA. Instructions how to work with YASARA can be found in the user manual, either by clicking *Help > Show user manual* directly in YASARA, or by opening the PDF file yasara/doc/YASARA*Manual.pdf.

Structural bioinformatics is a research discipline with an ambitious goal: to move experiments from the real world into the computer, ex vivo in silicum, where they should be conducted in a faster and easier way, cost less and require the help of fewer animals. It is needless to say that we are still very far from reaching that goal. The rational design of drugs has not really lived up to the high expectations yet, and none of the major pharmaceutical companies can get away without robotic in vitro screening of real compounds. Why can virtual experiments not catch up with the real ones?

Most applications start with atomic models (e.g. an enzyme with various drug candidates bound in the active site) and end with a simple question: which model is closest to reality? If it was possible to answer this question reliably, one could make random changes to the model and arrive at the correct answer. Today, the common approach is to calculate the model energies and gamble that the one with the lowest energy is also the most realistic model - just like in nature, where the configuration with the lowest energy is the most probable one. The main reason why this does not work well is the limited accuracy of today's energy functions. The difference between nature's real energy function and our partly empirical approximations is just too large. Consequently, a main focus during YASARA development has always been to improve these functions (also known as 'force fields') in order to arrive at more accurate predictions.

This manual contains four main chapters. This first chapter provides a general introduction to protein structure prediction[1,11], since proteins are the central objects of interest in structural bioinformatics and an ideal test-case for energy function accuracy. The focus is on homology modeling, because when a prediction is already very close to the real structure, the requirements on the energy functions are maximal and it becomes very hard to improve the model (the "last mile of the protein folding problem").

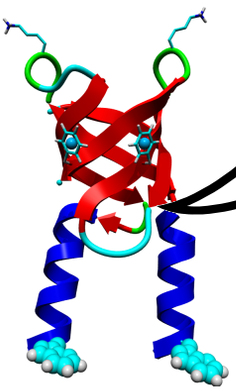
Chapter 2 deals with the algorithms that had to be developed as a foundation to support the actual work on energy functions: fast methods to visualize molecules[2] and simulate them efficiently[3], pKa prediction in periodic cells[4] and hydrogen bonding network optimization[5] to help with the reconstruction of complete crystallographic unit cells for force field parameter optimization, the PDBFinder II database[12,13] to help pick suitable structures, and Models@Home, a screensaver that delivers the required computing power by linking the PCs in a network[6].

Chapter 3 then introduces a new way of improving protein force fields: they are allowed to parameterize themselves while energy-minimizing known protein structures. First this is done *in vacuo* to obtain the NOVA force field[7], then in crystal space with explicit solvent yielding the YAMBER force fields[8], and finally the accuracy is increased further by adding knowledge-based potentials to arrive at the YASARA force fields[9].

Chapter 4 ends this manual by showing the actual applications of the newly developed methods: how YASARA was used in direct collaborations with experimental research groups to help answer their questions. We investigated mutations in the DJ-1 protein associated with parkinsonism[14,15], in gamma actin[16], myosin VIIa[17] and myosin 15A[18] causing hearing loss, in MPP family members[19,20] and RPGRIP1[21] causing blindness, in fibroblast growth factor 14 leading to cerebellar ataxia[22], and in p63 linked to ADULT syndrome[23]. In addition we modeled the interaction of heparin with interleukin-8[10], the peroxisomal targeting signal receptor Pex5p[24], the HIV-1 receptor on dendritic cells DC-SIGN[25], the Vps29 retromer component[26], and the ATPases Drg1[27], ATP7B[28] and ATP8B1[29]. A larger joint-project with Prof. Jan Koenderink focused on multidrug resistance protein 4[30,31] and the gastric H,K-ATPase: an essential salt bridge[32], the potassium binding pocket[33] and a potential ouabain binding site[34].YASARA's NMR module helped to determine the structure of Nore1[35], analyze the information content of NMR restraints[36,37] and validate NMR ensembles[38-40].


1. Krieger, E., Nabuurs, S. B. & Vriend, G. Homology modeling. *Methods Biochem Anal* **44**, 509-23 (2003).
2. Krieger, E. & Vriend, G. YASARA View - molecular graphics for all devices - from smartphones to workstations. *Bioinformatics* **30**, 2981-2982 (2014).
3. Krieger, E. & Vriend, G. New ways to boost molecular dynamics simulations. *J.Comp.Chem.* **36**, 996-1007 (2015).
4. Krieger, E., Nielsen, J. E., Spronk, C. A. E. M. & Vriend, G. Fast empirical pKa prediction by Ewald summation. *J Mol Graph Model* **25**, 481-486 (2006).
5. Krieger, E., Dunbrack, R. L., Hooft, R. W. W. & Krieger, B. Assignment of protonation states in proteins and ligands: combining pKa prediction with hydrogen bonding network optimization. *Methods Mol.Biol.* **819**, 405-421 (2012).
6. Krieger, E. & Vriend, G. Models@Home: distributed computing in bioinformatics using a screensaver based approach. *Bioinformatics* **18**, 315-318 (2002).
7. Krieger, E., Koraimann, G. & Vriend, G. Increasing the precision of comparative models with YASARA NOVA - a self-parameterizing force field. *Proteins* **47**, 393-402 (2002).
8. Krieger, E., Darden, T., Nabuurs, S. B., Finkelstein, A. & Vriend, G. Making optimal use of empirical energy functions: force field parameterization in crystal space. *Proteins* **57**, 678-683 (2004).
9. Krieger, E., Joo, K., Lee, J., Lee, J., Raman, S., Thompson, J., Tyka, M., Baker, D. & Karplus, K. Improving physical realism, stereochemistry and side-chain accuracy in homology modeling: four approaches that performed well in CASP8. *Proteins* **77, Suppl. 9**, 114-122 (2009).
10. Krieger, E., Geretti, E., Brandner, B., Goger, B., Wells, T. N. & Kungl, A. J. A structural and dynamical model for the interaction of interleukin-8 and glycosaminoglycans: support from isothermal fluorescence titrations. *Proteins* **54**, 768-775 (2004).
11. Venselaar, H., Joosten, R. P., Vroling, B., Baakman, C. A., Hekkelman, M. L., Krieger, E. & Vriend, G. Homology modelling and spectroscopy, a never-ending love story. *Eur.Biophys.J.* **39**, 551-563 (2010).

12. Joosten, R. P., te Beek, T. A., Krieger, E., Hekkelman, M. L., Hooft, R. W. W., Schneider, R., Sander, C. & Vriend, G. A series of PDB related databases for everyday needs. *Nucleic Acids Res.* **39**, D411-D419 (2011).
13. Touw, W. G., Baakman, C., Black, J., te Beek, T. A., Krieger, E., Joosten, R. P. & Vriend, G. A series of PDB-related databanks for everyday needs. *Nucleic Acids Res.* **43**, D364-D368 (2015).
14. Bonifati, V., Rizzu, P., van Baren, M. J., Schaap, O., Breedveld, G. J., Krieger, E., Dekker, M. C., Squitieri, F., Ibanez, P., Joosse, M., van Dongen, J. W., Vanacore, N., van Swieten, J. C., Brice, A., Meco, G., van Duijn, C. M., Oostra, B. A. & P., H. Mutations in the DJ-1 gene associated with autosomal recessive early-onset parkinsonism. *Science* **299**, 256-259 (2003).
15. Bonifati, V., Rizzu, P., Squitieri, F., Krieger, E., Vanacore, N., van Swieten, J. C., Brice, A., van Duijn, C. M., Oostra, B., Meco, G. & Heutink, P. DJ-1( PARK7), a novel gene for autosomal recessive, early onset parkinsonism. *Neurol Sci* **24**, 159-60 (2003).
16. van Wijk, E., Krieger, E., Kemperman, M. H., de Leenheer, E. M. R., Huygen, P. L. M., Cremer, C. W. R. J., Cremers, F. P. M. & Kremer, H. A mutation in the gamma actin 1 (ACTG1) gene causes autosomal dominant hearing loss. *J Med Genet* **40**, 879-84 (2004).
17. Luijendijk, M. W. J., van Wijk, E., Krieger, E., Bischoff, A. M. L. C., Cremers, C. W. R. J., Cremers, F. P. M., Kremer, H., Pennings, R. J. E., Weekamp, H., Cruysberg, J. R. M., Huygen, P. L. M. & Brunner, H. G. Identification and molecular modeling of a mutation in the motor head domain of myosin VIIA in a family with autosomal domi-nant hearing impairment (DFNA11). *Hum Genet* **115**, 149-156 (2004).
18. Kalay, E., Uzumcu, A., Krieger, E., Caylan, R., Uyguner, O., Ulubil-Emiroglu, M., Erdol, H., Kayserili, H., Hafiz, G., Baserer, N., Heister, A. J., Hennies, H. C., Nurnberg, P., Basaran, S., Brunner, H. G., Cremers, C. W., Karaguzel, A., Wollnik, B. & Kremer, H. MYO15A (DFNB3) mutations in Turkish hearing loss families and functional modeling of a novel motor domain mutation. *Am.J.Med.Genet.A* **143A**, 2382-2390 (2007).
19. Kantardzhieva, A., Gosens, I., Alexeeva, S., Punte, I. M., Versteeg, I., Krieger, E., Neefjes-Mol, C. A., den Hollander, A. I., Letteboer, S. J. F., Klooster, J., Cremers, F. P. M., Roepman, R. & Wijnholds, J. MPP5 Recruits MPP4 to the CRB1 Complex in Photoreceptors. *Invest Ophthalmol Vis Sci* **46**, 2192-2201 (2005).
20. Gosens, I., van Wijk, E., Kersten, F. F., Krieger, E., van der Zwaag, B., Marker, T., Letteboer, S. J., Dusseljee, S., Peters, T., Spierenburg, H. A., Punte, I. M., Wolfrum, U., Cremers, F. P., Kremer, H. & Roepman, R. MPP1 links the Usher protein network and the Crumbs protein complex in the retina. *Hum.Mol.Genet.* **16**, 1993-2003 (2007).
21. Roepman, R., Letteboer, S. J., Arts, H. H., van Beersum, S. E., Lu, X., Krieger, E., Ferreira, P. A. & Cremers, F. P. M. Interaction of nephrocystin-4 and RPGRIP1 is disrupted by nephronophthisis or Leber congenital amaurosis-associated mutations. *Proc.Natl.Acad.Sci.USA* **102** (2005).
22. van Swieten, J. C., Brusse, E., de Graaf, B. M., Krieger, E., van de Graaf, R., de Koning, I., Maat-Kievit, A., Leegwater, P., Dooijes, D., Oostra, B. A. & P., H. A mutation in the fibroblast growth factor 14 gene is associated with autosomal dominant cerebellar ataxia. *Am.J.Hum.Genet.* **72**, 191-199 (2003).
23. Duijf, P. H., Vanmolkot, K. R., Propping, P., Friedl, W., Krieger, E., McKeon, F., Dotsch, V., Brunner, H. G. & van Bokhoven, H. Gain-of-function mutation in ADULT syndrome reveals the presence of a second transactivation domain in p63. *Hum Mol Genet* **11**, 799-804 (2002).
24. Boteva, R., Koek, A., Visser, N. V., Visser, A. J., Krieger, E., Zlateva, T., Veenhuis, M. & van der Klei, I. Fluorescence analysis of the Hansenula polymorpha peroxisomal targeting signal-1 receptor, Pex5p. *Eur J Biochem* **270**, 4332-8 (2003).
25. Geijtenbeek, T. B., van Duijnhoven, G. C., van Vliet, S. J., Krieger, E., Vriend, G., Figdor, C. G. & van Kooyk, Y. Identification of different binding sites in the dendritic cell-specific receptor DC-SIGN for intercellular adhesion molecule 3 and HIV-1. *J Biol Chem* **277**, 11314-20 (2002).
26. Damen, E., Krieger, E., Nielsen, J. E., Eygensteyn, J. & van Leeuwen, J. E. The human Vps29 retromer component is a metallo-phosphoesterase for a cation-independent mannose 6-phosphate receptor substrate peptide. *Biochem.J.* **398**, 399-409 (2006).
27. Loibl, M., Klein, I., Prattes, M., Schmidt, C., Kappel, L., Zisser, G., Gungl, A., Krieger, E., Pertschy, B. & Bergler, H. The drug diazaborine blocks ribosome biogenesis by inhibiting the AAA-ATPase Drg1. *J.Biol.Chem.* **289**, 3913-3922 (2014).
28. van den Berghe, P. V., Stapelbroek, J. M., Krieger, E., de Bie, P., van de Graaf, S. F., de Groot, R. E., van Beurden, E., Spijker, E., Houwen, R. H., Berger, R. & Klomp, L. W. Reduced expression of ATP7B affected by Wilson disease-causing mutations is rescued by pharmacological folding chaperones 4-phenylbutyrate and curcumin. *Hepatology* **50**, 1783-1795 (2009).
29. van der Velden, L. M., Stapelbroek, J. M., Krieger, E., van den Berghe, P. V., Berger, R., Verhulst, P. M., Holthuis, J. C., Houwen, R. H., Klomp, L. W. & van de Graaf, S. F. Folding defects in P-type ATP 8B1 associated with hereditary cholestasis are ameliorated by 4-phenylbutyrate. **51**, 286-296 (2010).
30. El-Sheikh, A. A., van den Heuvel, J. J., Krieger, E., Russel, F. G. & Koenderink, J. B. Functional role of arginine 375 in transmembrane helix 6 of multidrug resistance protein 4 (MRP4/ABCC4). *Mol Pharmacol* **74**, 964-971 (2008).
31. Wittgen, H. G., van den Heuvel, J. J., Krieger, E., Schaftenaar, G., Russel, F. G. & Koenderink, J. B. Phenylalanine 368 of multidrug resistance-associated protein 4 (MRP4/ABCC4) plays a crucial role in substrate-specific transport activity. *Biochem.Pharmacol.* **84**, 366-373 (2012).
32. Koenderink, J. B., Swarts, H. G. P., Willems, P. H. G. M., Krieger, E. & De Pont, J. J. H. H. M. A conformational specific interhelical salt bridge is essential for the E2 preference of gastric H,K-ATPase. *J Biol Chem* **279**, 16417-16424 (2004).
33. Swarts, H. G., Koenderink, J. B., Willems, P. H., Krieger, E. & De Pont, J. J. H. H. M. Asn792 Participates in the Hydrogen Bond Network Around the K+-binding Pocket of Gastric H,K-ATPase. *J Biol Chem* **280**, 11488-11494 (2005).
34. Qiu, L. Y., Krieger, E., Schaftenaar, G., Swarts, H. G., Willems, P. H., De Pont, J. J. H. H. M. & Koenderink, J. B. Reconstruction of the Complete Ouabain-binding Pocket of Na,K-ATPase in Gastric H,K-ATPase by Substitution of Only Seven Amino Acids. *J Biol Chem* **280**, 32349-32355 (2005).
35. Harjes, E., Harjes, S., Wohlgemuth, S., Muller, K. H., Krieger, E., Herrmann, C. & Bayer, P. GTP-Ras Disrupts the Intramolecular Complex of C1 and RA Domains of Nore1. *Structure* **14**, 881-888 (2006).
36. Nabuurs, S. B., Spronk, C. A., Krieger, E., Maassen, H., Vriend, G. & Vuister, G. W. Quantitative evaluation of experimental NMR restraints. *J Am Chem Soc* **125**, 12026-34 (2003).
37. Nabuurs, S. B., Krieger, E., Spronk, C. A. E. M., Nederveen, A. J., Vriend, G. & Vuister, G. W. Definition of a new information-based per-residue quality parameter. *J Biomol NMR* **33**, 123-134 (2005).
38. Spronk, C. A., Nabuurs, S. B., Bonvin, A. M., Krieger, E., Vuister, G. W. & Vriend, G. The precision of NMR structure ensembles revisited. *J Biomol NMR* **25**, 225-34 (2003).
39. Spronk, C. A. E. M., Nabuurs, S. B., Krieger, E., Vriend, G. & Vuister, G. W. Validation of protein structures derived by NMR-spectroscopy. *Progress in NMR spectroscopy* **45**, 315-337 (2004).
40. Doreleijers, J. F., Sousa da Silva, A. W., Krieger, E., Nabuurs, S. B., Spronk, C. A., Stevens, T. J., Vranken, W. F., Vriend, G. & Vuister, G. W. CING: an integrated residue-based structure validation program suite. *J.Biomol.NMR* **54**, 267-283 (2012).



Normally time-intensive experiments are required to determine a protein structure. Two main techniques are available: X-ray diffraction on protein crystals, and nuclear magnetic resonance spectroscopy. The first is very accurate, but crystals can be hard to grow and protein structures may differ slightly in the crystal and in the living cell. The second allows to solve structures in their actual environment but yields mainly information about interatomic distances and dihedral angles that must be converted to a 3D structure. This embedding step does not always work out perfectly. Predicting a protein structure is much faster than solving it experimentally, but therefore the accuracy is often also much lower. This chapter describes how structure prediction works.

# Introduction to homology modeling

Elmar Krieger, Sander B. Nabuurs and Gert Vriend

*Methods of Biochemical Analysis* **44**, 509-523 (2003)

## Introduction

The ultimate goal of protein modeling is to predict a structure from its sequence with an accuracy that is comparable to the best results achieved experimentally. This would allow to safely use rapidly generated *in silico* protein models in all the contexts where today only experimental structures provide a solid basis: structure-based drug design, analysis of protein function, interactions, or antigenic behavior, and rational design of proteins with increased stability or novel functions. In addition, protein modeling is the only way to obtain structural information if experimental techniques fail. Many proteins are simply too large for NMR analysis and cannot be crystallized for X-ray diffraction.

There are two major approaches to 3D structure prediction: one builds proteins from scratch, while the other one tries to modify known structures to arrive at the answer. The second approach is called *homology modeling* or *comparative modeling* and is considerably easier. It is based on two major observations:

The structure of a protein is uniquely determined by its amino acid sequence[1]. Knowing the sequence should at least in theory suffice to obtain the structure.

During evolution, the structure is more stable and changes much slower than the associated sequence, so that similar sequences adopt practically identical structures, and distantly related sequences still fold into similar structures. This relationship was first identified by Chothia & Lesk[2] and later quantified by Sander & Schneider[3]. Thanks to the exponential growth of the Protein Data Bank, Rost could recently derive a more precise limit for this rule, shown in Figure 1[4]. As long as the length of two sequences and the percentage of identical residues fall in the region marked as "safe", the two sequences are practically guaranteed to adopt a similar structure.

Imagine that we want to know the structure of sequence A (150 amino acids long, Figure 2). We compare sequence A to all the sequences of known structures stored in the PDB (using for example BLAST), and luckily find a sequence B (300 amino acids long) containing a region of 150 amino acids that match sequence A with 50% identical residues. As this match ("alignment") clearly falls in the safe zone (Figure 1), we can simply take the known structure of sequence B (the "template"), cut out the fragment corresponding to the aligned region, mutate those amino acids that differ between sequences A and B, and finally arrive at our model for structure A. Structure A is called the "target" and is of course not known at the time of modeling.

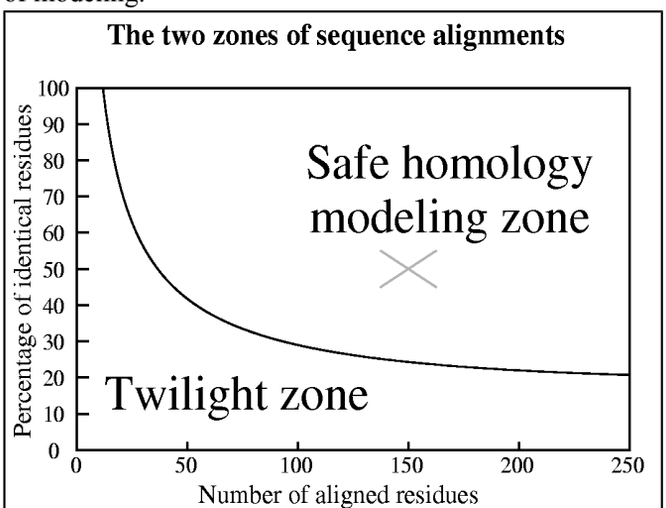



**Figure 1:** The two zones of sequence alignments. Two sequences are practically guaranteed to fold into the same structure if their length and percentage sequence identity fall into the region marked as "safe". An example of two sequences with 150 amino acids, 50% of which are identical, is shown (gray cross).

In practice, homology modeling is a multi-step process which can be summarized as follows:

(1) Template recognition and initial alignment
(2) Alignment correction
(3) Backbone generation
(4) Loop modeling

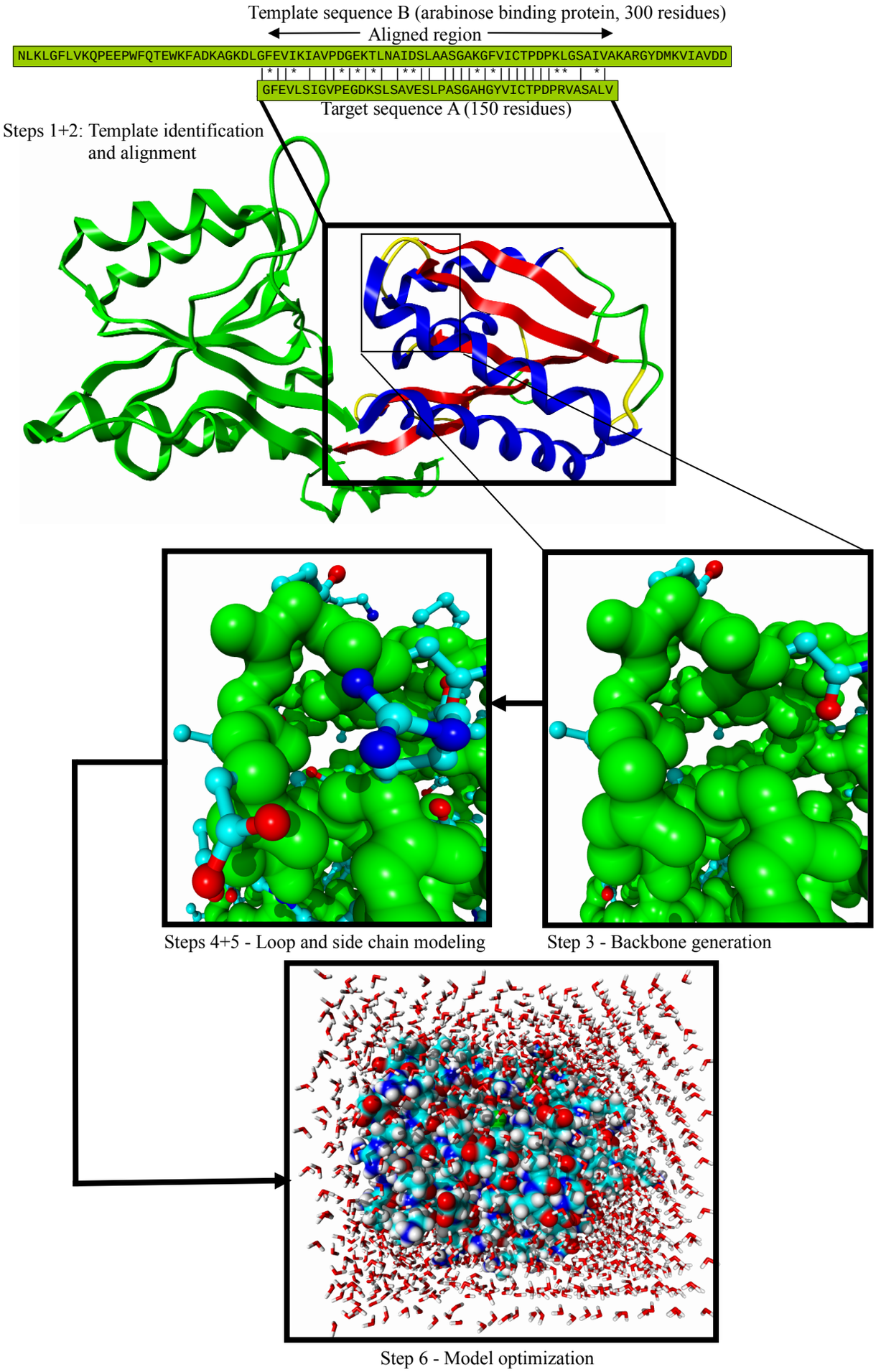
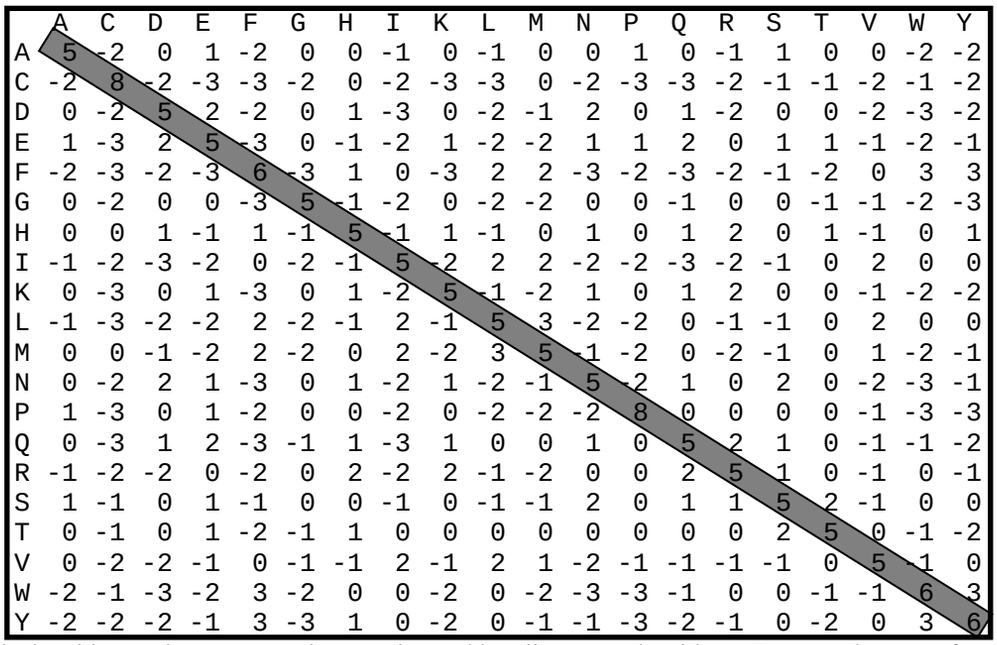



**Figure 2:** The steps to homology modeling. The fragment of the template (arabinose binding protein) corresponding to the region aligned with the target sequence forms the basis of the model (including conserved side chains). Loops and missing side chains are predicted, then the model is optimized (in this case together with surrounding water molecules).



| | A | C | D | E | F | G | H | I | K | L | M | N | P | Q | R | S | T | V | W | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 5 | -2 | 0 | 1 | -2 | 0 | 0 | -1 | 0 | -1 | 0 | 0 | 1 | 0 | -1 | 1 | 0 | 0 | -2 | -2 |
| C | -2 | 8 | -2 | -3 | -3 | -2 | 0 | -2 | -3 | -3 | 0 | -2 | -3 | -3 | -2 | -1 | -1 | -2 | -1 | -2 |
| D | 0 | -2 | 5 | 2 | -2 | 0 | 1 | -3 | 0 | -2 | -1 | 2 | 0 | 1 | -2 | 0 | 0 | -2 | -3 | -2 |
| E | 1 | -3 | 2 | 5 | -3 | 0 | -1 | -2 | 1 | -2 | -2 | 1 | 1 | 2 | 0 | 1 | 1 | -1 | -2 | -1 |
| F | -2 | -3 | -2 | -3 | 6 | -3 | 1 | 0 | -3 | 2 | 2 | -3 | -2 | -3 | -2 | -1 | -2 | 0 | 3 | 3 |
| G | 0 | -2 | 0 | 0 | -3 | 5 | -1 | -2 | 0 | -2 | -2 | 0 | 0 | -1 | 0 | 0 | -1 | -1 | -2 | -3 |
| H | 0 | 0 | 1 | -1 | 1 | -1 | 5 | -1 | 1 | -1 | 0 | 1 | 0 | 1 | 2 | 0 | 1 | -1 | 0 | 1 |
| I | -1 | -2 | -3 | -2 | 0 | -2 | -1 | 5 | -2 | 2 | 2 | -2 | -2 | -3 | -2 | -1 | 0 | 2 | 0 | 0 |
| K | 0 | -3 | 0 | 1 | -3 | 0 | 1 | -2 | 5 | -1 | -2 | 1 | 0 | 1 | 2 | 0 | 0 | -1 | -2 | -2 |
| L | -1 | -3 | -2 | -2 | 2 | -2 | -1 | 2 | -1 | 5 | 3 | -2 | -2 | 0 | -1 | -1 | 0 | 2 | 0 | 0 |
| M | 0 | 0 | -1 | -2 | 2 | -2 | 0 | 2 | -2 | 3 | 5 | -1 | -2 | 0 | -2 | -1 | 0 | 1 | -2 | -1 |
| N | 0 | -2 | 2 | 1 | -3 | 0 | 1 | -2 | 1 | -2 | -1 | 5 | -2 | 1 | 0 | 2 | 0 | -2 | -3 | -1 |
| P | 1 | -3 | 0 | 1 | -2 | 0 | 0 | -2 | 0 | -2 | -2 | -2 | 8 | 0 | 0 | 0 | 0 | -1 | -3 | -3 |
| Q | 0 | -3 | 1 | 2 | -3 | -1 | 1 | -3 | 1 | 0 | 0 | 1 | 0 | 5 | 2 | 1 | 0 | -1 | -1 | -2 |
| R | -1 | -2 | -2 | 0 | -2 | 0 | 2 | -2 | 2 | -1 | -2 | 0 | 0 | 2 | 5 | 1 | 0 | -1 | 0 | -1 |
| S | 1 | -1 | 0 | 1 | -1 | 0 | 0 | -1 | 0 | -1 | -1 | 2 | 0 | 1 | 1 | 5 | 2 | -1 | 0 | 0 |
| T | 0 | -1 | 0 | 1 | -2 | -1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 5 | 0 | -1 | -2 |
| V | 0 | -2 | -2 | -1 | 0 | -1 | -1 | 2 | -1 | 2 | 1 | -2 | -1 | -1 | -1 | -1 | 0 | 5 | -1 | 0 |
| W | -2 | -1 | -3 | -2 | 3 | -2 | 0 | 0 | -2 | 0 | -2 | -3 | -3 | -1 | 0 | 0 | -1 | -1 | 6 | 3 |
| Y | -2 | -2 | -2 | -1 | 3 | -3 | 1 | 0 | -2 | 0 | -1 | -1 | -3 | -2 | -1 | 0 | -2 | 0 | 3 | 6 |

**Figure 3:** A typical residue exchange or scoring matrix used by alignment algorithms. Because the score for aligning residues A and B is normally the same as for B and A, this matrix is symmetric.

(5) Side chain modeling
(6) Model optimization
(7) Model validation

At almost all the steps choices have to be made. The modeler can virtually never be sure to make the best ones, and thus a large part of the modeling process consists of serious thought about how to gamble between multiple seemingly similar choices. A lot of research has been spent on 'teaching' the computer how to make these decisions, so that homology models can be built fully automatically. Currently, this allows to construct models for about 25% of the amino acids in a genome, thereby supplementing the efforts of structural genomics projects[5,6]. For the remaining ~75% of a genome, no template with a known structure is available (or cannot be detected easily), and one must use fold recognition , *ab initio* folding techniques , or simply an experiment to obtain structural data. While automated model building provides high throughput, the evaluation of these methods during CASP indicated that human expertise is still helpful, especially if the alignment is close to the twilight zone [7].

## Step 1 - Template recognition and initial alignment

In the safe homology modeling zone (Figure 1), the percentage identity between the sequence of interest and a possible template is high enough to be detected with simple sequence alignment programs like BLAST[8] or FASTA[9].

To identify these hits, the program compares the query sequence to all the sequences of known structures in the PDB using mainly two matrices:

**(1) A residue exchange matrix** (Figure 3). The elements of this 20*20 matrix define the likelihood that any two of the 20 amino acids ought to be aligned. It is clearly seen that the values along the diagonal (representing conserved residues) are highest, but one can also observe that exchanges between residue types with similar physicochemical properties (for example F->Y) get a better score than exchanges between residue types that widely differ in their properties.

**(2) An alignment matrix** (Figure 4). The axes of this matrix correspond to the two sequences to align, and the matrix elements are simply the values from the residue exchange matrix (Figure 3) for a given pair of residues. During the alignment process, one tries to find the best path through this matrix, starting from a point near the top left, and going down to the bottom right. To make sure that no residue is used twice, one must always take at least one step to the right and one step down. A typical alignment path is shown in Figure 4. At first sight, the dashed path in the bottom right corner would have led to a higher score. However, it requires to open an additional gap in sequence A (Gly of sequence B is skipped). By comparing thousands of sequences and sequence families, it became clear that the opening of gaps is about as unlikely as at least a couple of non-identical residues in a row. The jump roughly in the middle of the matrix on the other hand is justified, because after the jump we earn lots of points (5,6,5) which would have been (1,0,0) without the jump. The alignment algorithm therefore subtracts an "opening penalty" for every new gap and a much smaller "gap extension penalty" for every residue that is skipped in the alignment. The gap extension penalty is smaller simply because one gap of three residues is much more likely than three gaps of one residue each.

In practice, one just feeds the query sequence to one of the countless BLAST servers on the web, selects to search the PDB, and obtains a list of hits - the modeling templates and corresponding alignments (Figure 2).

## Step 2 - Alignment correction

Having identified one or more possible modeling templates using the fast methods described above, it is time to

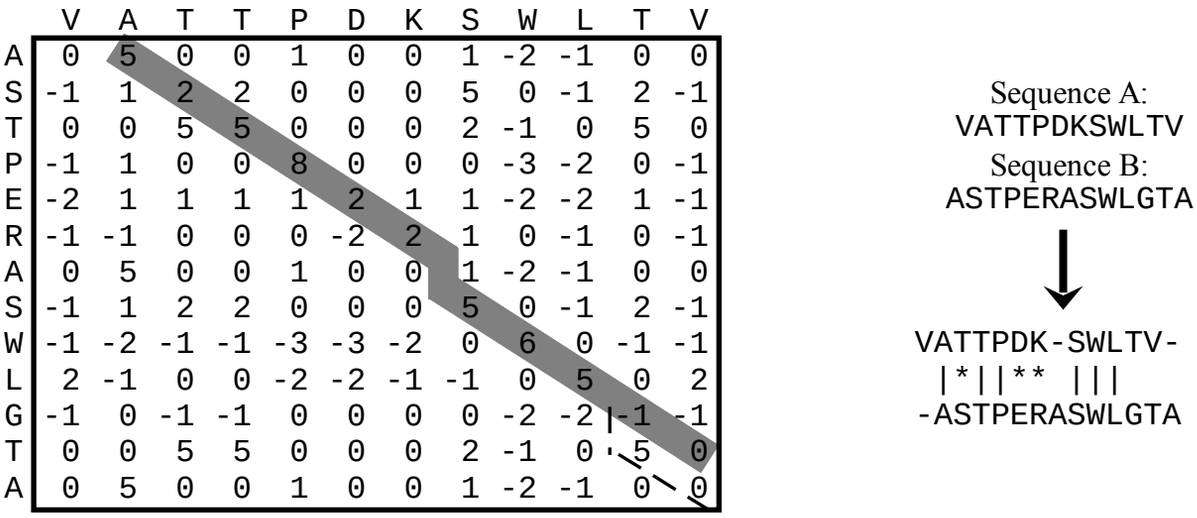



**Figure 4:** The alignment matrix for the sequences VATTPDKSWLTV and ASTPERASWLGTA, using the scores from Figure 3. The optimum path corresponding to the alignment on the right side is shown in gray. Residues with similar properties are marked with a star '*'. The dashed line marks an alternative alignment that scores more points but requires to open a second gap.

consider more sophisticated methods to arrive at a better alignment.

Sometimes it may be difficult to align two sequences in a region where the percentage sequence identity is very low. One can then use other sequences from homologous proteins to find a solution. A pathological example is shown in Figure 5: suppose you want to align the sequence LTLTLTLT with YAYAYAYAY. There are two equally poor possibilities, and only a third sequence, TYTYTYTYT, which aligns easily to both of them can solve the issue.

The example above introduced a very powerful concept called "multiple sequence alignment". Many programs are available to align a bunch of related sequences, for example CLUSTALW[10], and the resulting alignment contains a lot of additional information. Think about an Ala->Glu mutation. Relying on the matrix in Figure 3, this exchange always gets a score of 1. In the three dimensional structure of the protein, it is however very unlikely to see such an Ala->Glu exchange in the hydrophobic core, but on the surface this mutation is perfectly normal. The multiple sequence alignment implicitly contains information about this structural context. If at a certain position only exchanges between hydrophobic residues are observed, it is highly likely that this residue is buried. To consider this knowledge during the alignment, one uses the multiple sequence alignment to derive position specific scoring matrices, also called "profiles"[11,12].

When building a homology model, we are in the fortunate situation of having an almost perfect profile - the known structure of the template. We simply know that a certain alanine sits in the protein core and must therefore not be aligned with a glutamate. Multiple sequence alignments are nevertheless useful in homology modeling, for example to place deletions (missing residues in the model) or insertions (additional residues in the model) only in areas where the sequences are strongly divergent. A typical example for correcting an alignment with the help of the template is shown in Figures 6 and 7. While a simple sequence alignment gives the highest score for the wrong answer (alignment 1 in Figure 6), a simple look at the structure of the template reveals that alignment 2 is correct, because it leads to a small gap, compared to a huge hole associated with alignment 1.

## Step 3 - Backbone generation

When the alignment is ready, the actual model building can start. Creating the backbone is trivial for most of the model: one simply copies the coordinates of those template residues that show up in the alignment with the model sequence (Figure 2). If two aligned residues differ, only the backbone coordinates (N,CA,C and O) can be copied. If they are the same, one can also include the side chain (at least for not too flexible side chains, rotamers tend to be conserved).

Experimentally determined protein structures are not perfect (but still better than models in most cases). There are countless sources of errors, ranging from poor electron density in the X-ray diffraction map to simple human errors when preparing the PDB file for submission. A lot of work has been spent on writing software to detect these errors (correcting them is even harder), and the current count is at more than 10.000.000 problems in the 17.000 structures deposited in the PDB by the end of 2001. It is obvious that a straightforward way to build a good model is to choose the template with the fewest errors (the PDBREPORT database[13] at www.cmbi.ru.nl/pdbreport can be very helpful). But what if two templates are available, and each has a poorly determined region, but these regions are not the same? One should clearly combine the good parts of both templates in one model - an approach known as "multiple template modeling". (The same applies if the alignments between the model sequence and possible templates show good matches in different regions). Although in principle simple (and done by automated modeling servers like Swiss-Model[6]), it is hard in practice to achieve results that are really closer to the true structure than all the templates.

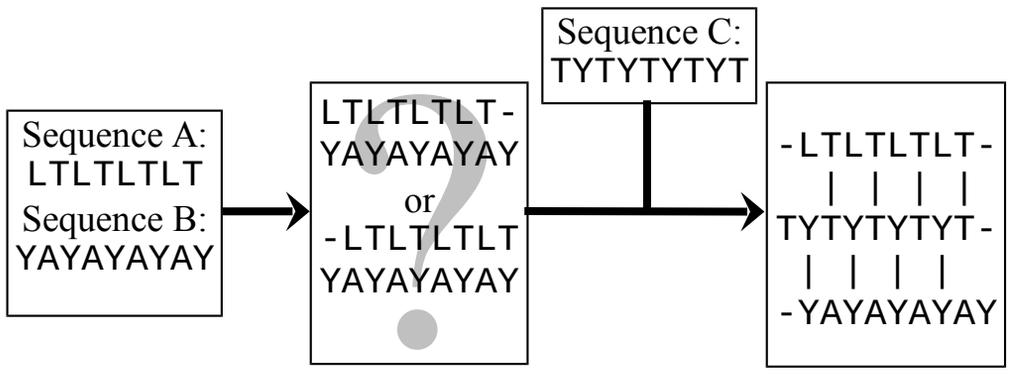



**Figure 5**: A pathological alignment problem. Sequences A and B are impossible to align, unless one considers a third sequence C from a homologous protein.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Template | PHE | ASP | ILE | CYS | ARG | LEU | PRO | GLY | SER | ALA | GLU | ALA | VAL |
| Model(bad) 1 | PHE | ASN | VAL | CYS | ARG | ALA | PRO | --- | --- | --- | GLU | ALA | ILE |
| Model(good) 2 | PHE | ASN | VAL | CYS | ARG | --- | --- | --- | ALA | PRO | GLU | ALA | ILE |

**Figure 6:** Example of a sequence alignment where a three-residue deletion must be modeled. While the first alignment appears better when considering just the sequences (a matching proline at position 7), a look at the structure of the template leads to a different conclusion (Figure 7).

## Step 4 - Loop modeling

In the majority of cases, the alignment between model and template sequence contains gaps. Either gaps in the model sequence (deletions as shown in Figures 6 and 7) or in the template sequence (insertions). In the first case one simply omits residues from the template, creating a hole in the model that must be closed. In the second case, one takes the continuous backbone from the template, cuts it, and inserts the missing residues. Both cases imply a conformational change of the backbone. The good news is that conformational changes usually don't happen within regular secondary structure elements. It is therefore safe to shift all insertions or deletions in the alignment out of helices and strands, placing them in loops and turns. The bad news is that these changes in loop conformation are notoriously hard to predict (the big unsolved problem in homology modeling). To make things worse, even without insertions or deletions do we often find quite different loop conformations in template and target. Three main reasons can be identified:

(1) Surface loops tend to be involved in crystal contacts, leading to a significant conformational change between template and target.

(2) The exchange of small to bulky side chains underneath the loop pushes it aside.

(3) The mutation of a loop residue to proline or from glycine to any other residue. In both cases, the new residue must fit into a more restricted area in the Ramachandran plot, which most of the time requires conformational changes of the loop.

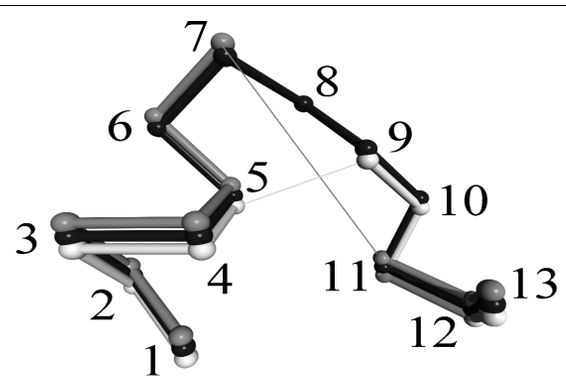



**Figure 7:** Correcting an alignment based on the structure of the modeling template (Cα-trace shown in black). While the alignment with the highest score (dark gray, also in Figure 6) leads to a gap of 7.5Å between residues 7 and 11, the second option (white) creates only a tiny hole of 1.3Å between residues 5 and 9. This can easily be accommodated by small backbone shifts. (The normal Cα-Cα distance of 3.8Å has been subtracted).

There are two main approaches to loop modeling:

**(1) Knowledge based**: one searches the PDB for known loops with endpoints that match the residues between which the loop has to be inserted, and simply copies the loop conformation. All major molecular modeling programs and servers support this approach (e.g. 3D-Jigsaw[14], Insight[15], Modeller[16], Swiss-Model[6] or WHAT IF[17]).

**(2) Energy based**: as in true *ab initio* fold prediction, an energy function is used to judge the quality of a loop. Then this function is minimized, using Monte Carlo[18] or molecular dynamics techniques[19] to arrive at the "best" loop conformation. Often the energy function is modified ("smoothed") to facilitate the search[20].

At least for short loops (up to 5-8 residues), the various methods have a reasonable chance of predicting a loop conformation that superposes well on the true structure. As mentioned above, surface loops tend to change their conformation due to crystal contacts. So if the prediction is made for an isolated protein and then found to differ from the crystal structure, it might still be correct.

## Step 5 - Side chain modeling

When we compare the side chain conformations ("rotamers") of residues that are conserved in structurally similar proteins, we find that they often have similar $\chi_1$-angles (i.e. the torsion angle about the $C_\alpha$-$C_\beta$ bond). It is therefore possible to simply copy conserved residues entirely from the template to the model (see also step 3) and achieve a higher accuracy than by copying just the backbone and re-predicting the side chains. In practice, this rule of thumb holds only at high levels of sequence identity, when the conserved residues form networks of contacts. When they get isolated (<35% sequence identity), the rotamers of conserved residues may differ in up to 45% of the cases[21].

Practically all successful approaches to side chain placement are at least partly knowledge based: They use libraries of common rotamers, extracted from high resolution X-ray structures. The various rotamers are tried successively and scored with a variety of energy functions. Intuitively, one might expect rotamer prediction to be computationally demanding due to the "combinatorial explosion": The choice of a certain rotamer automatically affects the rotamers of all neighboring residues, which in turn affect their neighbors and so on. With 100 residues and on average ~5 rotamers per residue, one would already end up at $5^{100}$ different combinations to score - that's about a 1 with 70 zeros. A lot of research has been spent on the development of methods to make this enormous search space tractable[22]. The number of combinations is in fact so large, that even nature could not try all of them during the folding process. This already indicates that there must exist mechanisms to shrink down the search-space.

Beside the trivial fact that copying conserved rotamers from the template often splits up the protein into distinct regions where rotamers can be predicted independently, the key to handling the "combinatorial explosion" lies in the protein backbone: Certain backbone conformations strongly favor certain rotamers (allowing for example a hydrogen bond between side chain and backbone) and thus greatly reduce the search-space. For a given backbone conformation, there may be only one strongly populated rotamer which can be modeled right away, thereby providing an anchor for surrounding, more flexible side chains. An example for a backbone conformation that favors two different tyrosine rotamers is shown in Figure 8. These "position-specific rotamer libraries" are widely used today[23-25]. To build such a library, one takes high resolution structures and collects all stretches of 3 to 7 residues (depending on the method) with a given amino acid in the center. To predict a rotamer, the corresponding backbone stretch in the template is superposed on all the collected examples, and the possible side chain conformations are selected from the best backbone matches[26].

Further evidence that the combinatorial problem of rotamer prediction is far smaller than originally believed was found recently: Xiang and Honig first removed one single side chain from known structures and re-predicted it. In a second step, they removed all the side chains and added them again using the same simple search strategy. Surprisingly, it turned out that the accuracy was only marginally higher in the much easier first case[27].

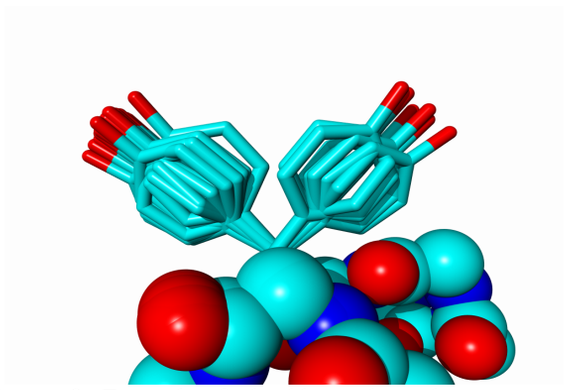


**Figure 8:** Example of a backbone-dependent rotamer library. The current backbone conformation (space-filling display) favors two different rotamers for Tyrosine (sticks) which appear about equally often in the database.

The prediction accuracy is usually quite high for residues in the hydrophobic core where >90% of all $\chi_1$-angles fall within ±20° from the experimental values, but much lower for residues on the surface where the percentage is often even below 50%. There are two reasons for that:

(1) Experimental reasons: flexible side chains on the surface tend to adopt multiple conformations, which are additionally influenced by crystal contacts. So even experiment cannot provide one single "correct answer".

(2) Theoretical reasons: the energy functions used to score rotamers can easily handle the hydrophobic packing in the core (mainly Van der Waals interactions), but are not accurate enough to get the complicated electrostatic interactions on the surface right, including hydrogen bonds with water molecules and associated entropic effects.

It is important to note that the prediction accuracies given in most publications cannot be reached in real-life applications. This is simply due to the fact that the methods are evaluated by taking a known structure, removing the side chains and re-predicting them. The algorithms thus rely on the "correct" backbone, which is not available in homology modeling: the backbone of the template often differs significantly from the target. The rotamers must thus be predicted based on a "wrong" backbone - and prediction accuracies tend to be lower in this case.

## Step 6 - Model Optimization

The problem just mentioned above leads to a classical "chicken and egg" situation: to predict the side chain rotamers with high accuracy, we need the correct backbone, which in turn depends on the rotamers and their packing. The common approach to such a problem is an iterative one: predict the rotamers, then the resulting shifts in the backbone, then the rotamers for the new backbone, and so on, until the procedure converges. This boils down to a sequence of rotamer prediction and energy minimization steps. The latter use the methods from the loop modeling

step above, but this time they must be applied to the entire protein structure, not just an isolated loop. This requires an enormous accuracy in the energy function, because there are many more paths leading away from the answer (the target structure) than towards it. That is why energy minimization must be used carefully. At every minimization step, a few big errors (like "bumps", i.e. too short atomic distances) are removed while at the same time many small errors are introduced. When the big errors are gone, the small ones start accumulating and the model moves away from the target (Figure 9).

**Energy minimization of homology models**

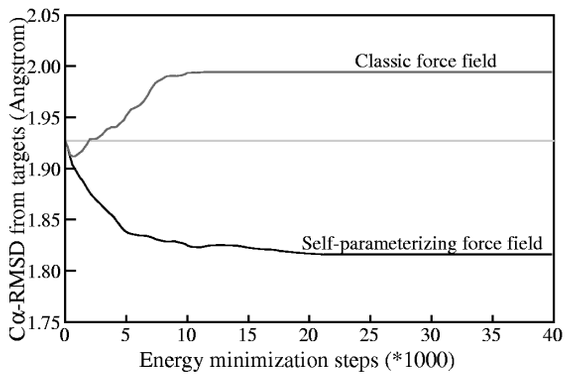



**Figure 9:** The average RMSD between models and targets during an extensive energy minimization of 14 homology models with two different force fields. Both force fields improve the models during the first ~500 energy minimization steps but then the small errors sum up in the classic force field and guide the minimization in the wrong direction, away from the target while the self-parameterizing force field goes in the right direction. To reach experimental accuracy, the minimization would have to proceed all the way down to ~0.5 Å which is the uncertainty in experimentally determined coordinates.

As a rule of thumb, today's modeling programs therefore either restrain the atom positions and/or apply only a few hundred steps of energy minimization. In short, model optimization does not work until energy functions ("force fields") get more accurate. Two ways to achieve that are currently being pursued:

(1) Quantum force fields: protein force fields must be fast to handle these large molecules efficiently, energies are therefore normally expressed as a function of the positions of the atomic nuclei only. The continuous increase of computer power has now finally made it possible to apply methods of quantum chemistry to entire proteins, arriving at more accurate descriptions of the charge distribution[28]. It is however still difficult to overcome the inherent approximations of today's quantum chemical calculations. Attractive Van der Waals forces are for example so hard to treat, that they must often be completely omitted. While providing more accurate electrostatics, the overall accuracy achieved is still about the same as in the "classical" force fields.

(2) Self-parameterizing force fields: the accuracy of a force field depends to a large extent on its parameters (e.g. Van der Waals radii, atomic charges). These parameters are usually obtained from quantum chemical calculations on small molecules and fitting to experimental data, following elaborate rules[29]. By applying the force field to proteins, one implicitly assumes that a peptide chain is just the sum of its individual small molecule building blocks - the amino acids. Alternatively, one can just state a goal - e.g. "improve the models during an energy minimization" - and then let the force field "parameterize itself" while trying to optimally fulfill this goal[30]. This leads to a computationally rather expensive procedure: Take initial parameters (for example from an existing force field), change a parameter randomly, energy minimize models, see if the result improved, keep the new force field if yes, otherwise go back to the previous force field. With this procedure, the force field accuracy increases enough to go into the right direction during an energy minimization (Figure 9), but experimental accuracy is still far out of reach.

The most straightforward approach to model optimization is to simply run a molecular dynamics simulation of the model. Such a simulation follows the motions of the protein on a femtosecond ($10^{-15}$ s) timescale and mimics the true folding process. One thus hopes that the model will complete its folding and "home in" to the true structure during the simulation. The advantage is that a molecular dynamics simulation implicitly contains entropic effects that are otherwise hard to treat, the disadvantage is that the force fields are again not accurate enough to make it work in practice. (One must in fact be happy if the model is not "messed up" during the simulation).

## Step 7 - Model Validation

Every homology model contains errors. The number of errors (for a given method) depends on mainly two values:

(1) The number of errors in the template.

(2) The percentage sequence identity between template and target. If it is greater than 90%, the accuracy of the model can be compared to crystallographically determined structures, except for a few individual side chains [2,32]. From 90% down to 50% identity, the RMS error in the modeled coordinates can be as large as 1.5 Å, with considerably larger local errors. If the sequence identity drops to 25%, the alignment turns out to be the main bottleneck for homology modeling, leading to often very large errors.

As with most errors, they become less of a problem when they can be localized. Upon modeling a protease it is probably not important that a loop far away from the active site is placed incorrectly. The essential step in the homology modeling process is therefore undoubtedly the verification of the model, and the estimation of the likelihood, magnitude and location of errors.

There are two principally different ways to estimate errors in a structure:

(1) Calculating the model's energy based on a molecular dynamics force field: This allows to check if the bond lengths and bond angles are within normal ranges, and if there are lots of bumps in the model (corresponding to a high Van der Waals energy). Truly es-

sential questions like "is the model folded correctly?" cannot be answered this way, because completely misfolded but well minimized models often reach the same force field energy as the target structure[33]. This is due to the fact that molecular dynamics force fields do not explicitly contain entropic terms (like hydrophobic interactions), but rely on the simulation to generate them. While this problem can be addressed by extending the force field and adding e.g. solvation terms[34] or know-ledge-based potentials[31], the major drawback is that one always obtains a single number for the entire protein and cannot trace problems down to individual residues.

**(2) Determination of normality indices** that describe how well a given characteristic of the model resembles the same characteristic in real structures. Many features of protein structures are well suited for normality analysis. Most of them are directly or indirectly based on the analysis of contacts, either inter-residue contacts, or contacts with water. Some published examples are:

(1) General checks for the normality of bond lengths, bond- and torsion angles[35,36] are good checks for the quality of experimentally determined structures, but are less suitable for the evaluation of models because the better model building programs simply do not make this kind of errors.

(2) Inside/outside distributions of polar and apolar residues can be used to detect completely misfolded models[37].

(3) Packing rules have been derived for structure evaluation[38].

(4) Atomic contacts that are not abundant in the protein structure database are good indicators of local model building problems[39]. If a contact between two residue fragments has the same distance and orientation as a contact that occurs often in the database of known structures, then a high score is given. If a contact in the model seems rather unique, a low score is given. This 'quality control' of local packing has proven to be a powerful tool for the detection of abnormal structures.

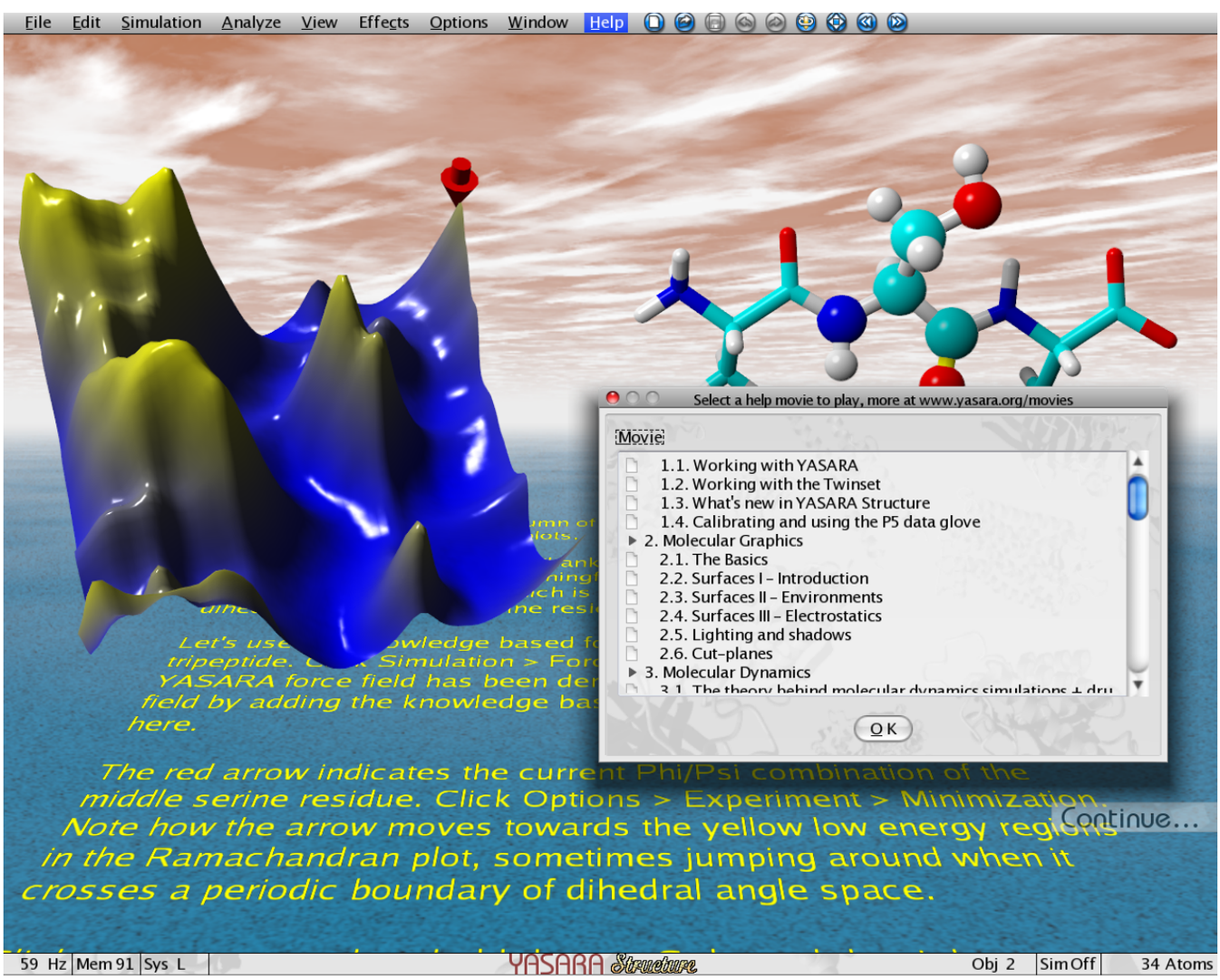
Most methods used for the verification of models can also be applied to experimental structures (and hence the templates used for model building). A detailed verification is essential when trying to derive new information from the model, either to interpret or predict experimental results or plan new experiments.

As a summary, it is safe to say that homology modeling is unfortunately not as easy as stated in the beginning. Ideally, homology modeling uses threading to improve the alignment, *ab initio* folding to predict the loops and molecular dynamics simulations with a perfect force field to home in to the "true" structure. Doing all that correctly will keep researchers busy for a long time, leaving lots of fascinating discoveries to good old experiment.

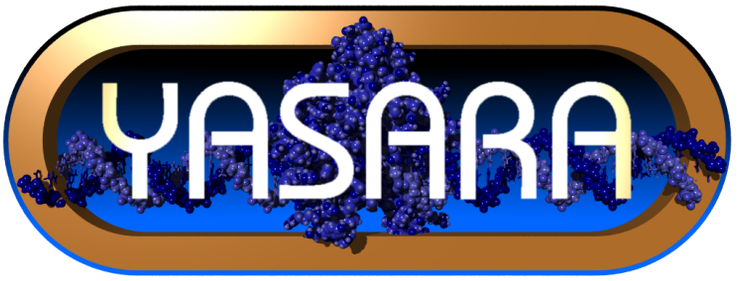
*Acknowledgements:* We thank Rolando Rodriguez, Sander Nabuurs, Chris Spronk, Rob Hooft, Chris Sander, Glay Chinea, Enzo de Filippis, Hans Doeberling and his team, Brigitte Altenberg, Karina Krmoian for stimulating discussions and practical help. We apologize to the numerous crystallographers who made all this work possible by depositing structures in the PDB for not referring to each of the 16000 very important articles describing these structures.

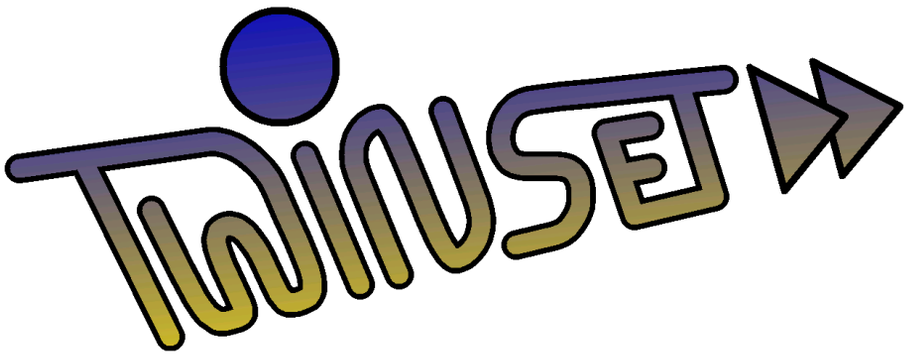
1. Epstain, C. J., Goldberger, R. F. & Anfinsen, C. B. *Cold Spring Harbor Symp.Quant.Biol.* **28**, 439 (1963).
2. Chothia, C. & Lesk, A. M. The relation between the divergence of sequence and structure in proteins. *EMBO J.* **5**, 823-836 (1986).
3. Sander, C. & Schneider, R. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins* **9**, 56-68 (1991).
4. Rost, B. Twilight zone of protein sequence alignments. *Protein Eng.* **12**, 85-94 (1999).
5. Sanchez, R. & Sali, A. ModBase: a database of comparative protein structure models. *Bioinformatics* **15**, 1060-1061 (1999).
6. Peitsch, M. C., Schwede, T. & Guex, N. Automated protein modelling - the proteome in 3D. *Pharmacogenomics* **1**, 257-266 (2000).
7. Fischer, D., Barret, C., Bryson, K., Elofsson, A., Godzik, A., Jones, D., Karplus, K. J., Kelley, L. A., MacCallum, R. M., Pawowski, K., Rost, B., Rychlewski, L. & Sternberg, M. J. E. CAFASP-1: Critical assessment of fully automated structure prediction methods. *Proteins, Suppl.* **3**, 209-217 (1999).
8. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J.Mol.Biol.* **215**, 403-410 (1990).
9. Pearson, W. R. Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol.* **183**, 63-98 (1990).
10. Thompson, J. D., Higgins, D. G. & Gibson, T. J. ClustalW: improving the sensitivity of progressive multiple sequence alignments through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **22**, 4673-4680 (1994).
11. Taylor, W. R. Identification of protein sequence homology by consensus template alignment. *J.Mol.Biol.* **188**, 233-258 (1986).
12. Dodge, C., Schneider, R. & Sander, C. The HSSP database of protein structure-sequence alignments and family profiles. *Nucleic Acids Res.* **26**, 313-315 (1998).
13. Hooft, R. W. W., Vriend, G., Sander, C. & Abola, E. E. Errors in protein structures. *Nature* **381**, 272-272 (1996).
14. Bates, P. A. & Sternberg, M. J. E. Model building by comparison at CASP3: using expert knowledge and computer automation. *Proteins, Suppl.* **3**, 47-54 (1999).
15. Dayringer, H. E., Tramontano, A. & Fletterick, R. J. Interactive program for visualization and modelling of proteins, nucleic acids and small molecules. *J.Mol.Graph.* **4**, 82-87 (1986).
16. Sali, A. & Blundell, T. L. Comparative protein modelling by satisfaction of spatial restraints. *J.Mol.Biol.* **234**, 779-815 (1993).
17. Vriend, G. WHAT IF - A molecular modeling and drug design program. *J.Mol.Graph.* **8**, 52-56 (1990).
18. Simons, K. T., Bonneau, R., Ruczinski, I. & Baker, D. Ab initio structure prediction of CASP III targets using ROSETTA. *Proteins, Suppl.* **3**, 171-176 (1999).
19. Fiser, A., Do, R. K. & Sali, A. Modeling of loops in protein structures. *Protein Sci.* **9**, 1753-1773 (2000).
20. Tappura, K. Influence of rotational energy barriers to the conformational search of protein loops in molecular dynamics and ranking the conformations. *Proteins* **44**, 167-179 (2001).
21. Sanchez, R. & Sali, A. Evaluation of comparative protein structure modeling by MODELLER-3. *Proteins, Suppl.* **1**, 50-58 (1997).
22. Desmet, J., De Maeyer, M., Hazes, B. & Lasters, I. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature* **356**, 539-542 (1992).
23. Stites, W. E., Meeker, A. K. & Shortle, D. Evidence for strained interactions between side-chains and the polypeptide backbone. *J.Mol.Biol.* **235**, 27-32 (1994).
24. Dunbrack, R. L. J. & Karplus, M. Conformational analysis of the backbone dependent rotamer preferences of protein side chains. *Nat.Struct.Biol.* **5**, 334-340 (1994).

25. de Filippis, V., Sander, C. & Vriend, G. Predicting local structural changes that result from point mutations. *Protein Eng.* **7**, 1203-1208 (1994).
26. Chinea, G., Padron, G., Hooft, R. W. W., Sander, C. & Vriend, G. The use of position specific rotamers in model building by homology. *Proteins* **23**, 415-421 (1995).
27. Xiang, Z. & Honig, B. Extending the accuracy limits of prediction for side-chain conformations. *J.Mol.Biol.* **311**, 421-430 (2001).
28. Liu, H., Elstner, M., Kaxiras, E., Frauenheim, T., Hermans, J. & Yang, W. Quantum mechanics simulation of protein dynamics on long timescale. *Proteins* **44**, 484-489 (2001).
29. Wang, J., Cieplak, P. & Kollman, P. A. How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules? *J.Comp.Chem.* **21**, 1049-1074 (2000).
30. Krieger, E., Koraimann, G. & Vriend, G. Increasing the precision of comparative models with YASARA NOVA - a self-parameterizing force field. *Proteins* **47**, 393-402 (2002).
31. Sippl, M. J. Calculation of conformational ensembles from potentials of mean force. *J.Mol.Biol.* **213**, 859-883 (1990).
32. Sippl, M. J. Recognition of errors in three dimensional structures of proteins. *Proteins* **17**, 355-362 (1993).
33. Novotny, J., Rashin, A. A. & Bruccoleri, R. E. Criteria that discriminate between native proteins and incorrectly folded models. *Proteins* **4**, 19-30 (1988).
34. Holm, L. & Sander, C. Evaluation of protein models by atomic solvation preference. *J.Mol.Biol.* **225**, 93-105 (1992).
35. Morris, A. L., MacArthur, M. W., Hutchinson, E. G. & Thorton, J. M. Stereochemical quality of protein structure coordinates. *Proteins* **12**, 345-364 (1992).
36. Czaplewski, C., Rodziewicz-Motowidlo, S., Liwo, A., Ripoll, D. R., Wawak, R. J. & Scheraga, H. A. Molecular simulation study of cooperativity in hydrophobic association. *Protein Sci.* **9**, 1235-1245 (2000).
37. Baumann, G., Frommel, C. & Sander, C. Polarity as a criterion in protein design. *Protein Eng.* **2**, 329-334 (1989).
38. Gregoret, L. M. & Cohen, F. E. Novel method for the rapid evaluation of packing in protein structures. *J.Mol.Biol.* **211**, 959-974 (1990).
39. Vriend, G. & Sander, C. Quality control of protein models: Directional atomic contact analysis. *J.Appl.Cryst.* **26**, 47-60 (1993).

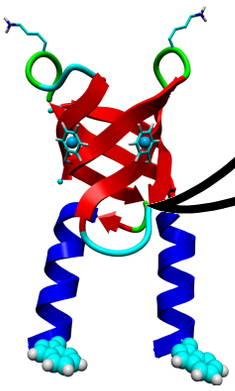




**Space filler:** Screenshot of YASARA's help movie 4.2 "Knowledge-based potentials"




# 2 The algorithms

Protein structures that have been solved experimentally are deposited at the Protein Data Bank (www.rcsb.org), where they are stored in flat text files called "PDB files". A PDB file contains the Cartesian coordinates of the protein atoms, residue names and numbers, and a lot of additional information about the experiment, literature references etc. Before one can really work with the proteins, the PDB file has to be parsed and converted to a three-dimensional representation of the protein that can be shown on screen. This is known as Molecular Graphics and can be a quite complicated task, especially if performance is important and hundreds of thousands of atoms need to be drawn quickly.

# YASARA View – molecular graphics for all devices – from smartphones to workstations

Elmar Krieger and Gert Vriend

*Bioinformatics* **30**, 2981-2982 (2014)

## Abstract

**Summary:** Today's graphics processing units (GPUs) compose the scene from individual triangles. Since about 320 triangles are needed to approximate a single sphere - an atom - in a convincing way, visualizing larger proteins with atomic details requires tens of millions of triangles, far too many for smooth interactive frame rates. We describe a new approach to solve this 'molecular graphics problem', which shares the work between GPU and multiple CPU cores, generates high quality results with perfectly round spheres, shadows and ambient lighting, and requires only OpenGL 1.0 functionality, without any pixel shader Z-buffer access (a feature which is missing on most mobile devices).

**Availability:** YASARA View, a molecular modeling program built around the visualization algorithm described here, is freely available (including commercial use) for Linux, MacOS, Windows and Android (Intel) from www.YASARA.org

## Introduction

In 1966, Cyrus Levinthal pioneered molecular graphics at the MIT, when he set up the first interactive wire-frame display of a protein on a monochrome oscilloscope[1]. Since then, molecular graphics has made tremendous progress, mostly thanks to the video game industry, that induced the rise of GPUs. Today many different molecular visualizers are available, e.g. VMD[2], Chimera[3], PyMol[4] or QuteMol[5] each using different tricks to boost rendering performance and quality. We describe an algorithm that can cope with two specific difficulties: first, it does not depend on high-end shader tricks and thus works on smartphones too. And second, it does not require expensive precalculation steps that depend on atom positions. It can thus visualize moving atoms, allowing to perform interactive molecular dynamics simulations on smartphones and tablets.

## Methods

The general idea is very simple and has been used ever since texture mapping* became part of 3D graphics: if an object is too complex (like the 960 triangles required to draw a single water molecule in Figure 1A) it is replaced with 'impostors', i.e. fewer triangles that have precalculated textures attached which make them look like the original object. So instead of drawing 320 triangles to create one, still somewhat edgy atom, we simply draw the precalculated image of a perfectly round atom. Since textures may be partly transparent, this image can be drawn as a simple square (transparent in the corners), which requires just two triangles.

In practice, many different images of atoms are needed, since atoms can have different colors and sizes. Regarding colors, we use blue, magenta, red, yellow, green, cyan and grey, and blend any two of them with a variable blending factor to support color gradients. Regarding sizes, the precalculated images can be shrunk on the fly during texture mapping, but the shrinking procedure reduces the image quality. That's why multiple smaller images of each atom are stored as well. Changing the texture during rendering reduces performance and consequently all these atom images are squeezed into a single texture of size 1024x1024, which is shown in Figure 1B. When the user changes the position of the light source, this texture is updated from a collection of 200 different views, pre-rendered with www.POVRay.org. For stereo graphics, a second texture is used, that has the atoms pre-rendered from a slightly shifted point of view.

* *texture mapping* means that a triangle is not rendered with a single color, but an image (the texture) is attached to it instead. For each of the three triangle vertices, the programmer can specify the corresponding 2D coordinates in the texture, and the hardware interpolates in between.

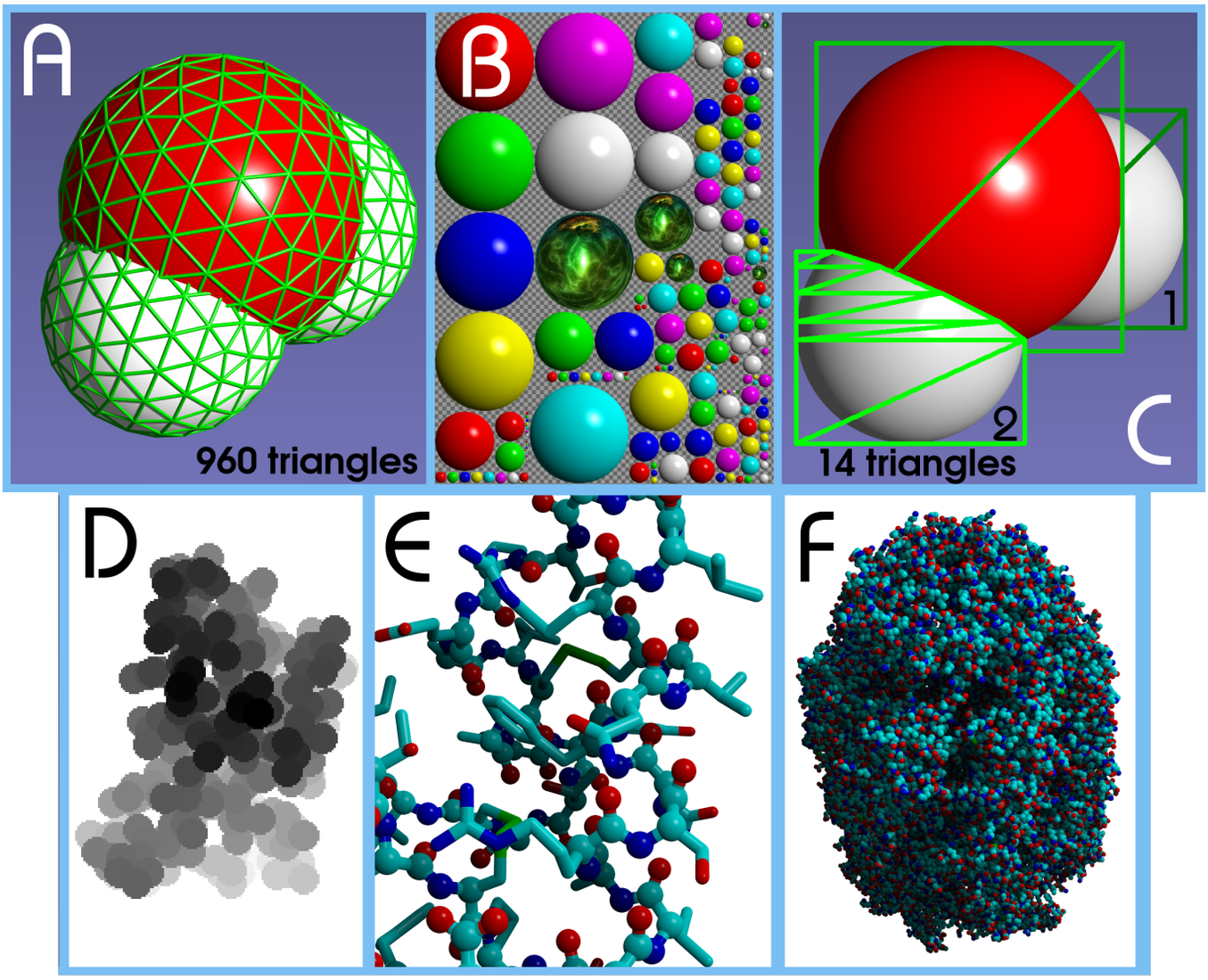



**Figure 1:** A water molecule rendered classically with 960 triangles (A) or quickly using texture mapping and precalculated impostors arranged in a single texture (B). The texture contains ray-traced images of spheres with various colors (two of which are blended with a variable factor to create other colors and color gradients) and various sizes (so called 'mipmaps', which reduce aliasing artifacts). The spheres coated with a stellar nebula are used to draw atoms selected by the user. The grey checkerboard indicates transparent pixels. Using texture (B), the water molecule in (A) can be drawn quickly using just 14 triangles (C). Low-res depth map of 1CRN to calculate shadows (D), balls&sticks of 1CRN (E) and space filling display of 1AON (F).

This straightforward approach has not been routinely used in the past for the following reason: when the GPU draws a pixel, it stores its Z-value (the distance from the view-plane) in the Z-buffer, and then never again draws a pixel at this location, unless it has a smaller Z-value. When spheres are modeled with lots of triangles as in Figure 1A, each pixel has the right Z-value associated, so that the spheres intersect correctly thanks to the Z-buffer. With our shortcut however, each sphere consists of just two triangles that are parallel to the view-plane, and each pixel of the sphere image thus has the same Z-value. Consequently, the spheres do not intersect at all, instead the closer one completely occludes the more distant one. This is fine when drawing non-intersecting spheres (in sticks and balls&sticks visualization styles), but obviously goes very wrong with a space filling style. The logical solution would be to adjust the pixel Z-values on the fly during rendering (with a so called 'pixel shader'), but this approach is either slow (because the hardware can no longer perform an early Z-test to discard pixels) or not supported at all (e.g. mobile devices based on OpenGL ES lack this feature, and PowerVR GPUs don't even have a Z-buffer). The algorithm described here therefore takes a different route, it shares the work between CPU and GPU according to the following recipe, which can easily be distributed over multiple CPU cores (a very detailed 20 page step-by-step recipe has been included as supplementary material):

(1) The CPU transforms the atom coordinates from object space to screen space and immediately discards atoms that are off-screen.

(2) For each atom i, the CPU creates a temporary Z-buffer that includes atom i and all the more distant atoms k which can influence the shape of atom i by intersection, i.e. those atoms whose sphere image touches atom i and who are closer along Z than their own radius Rk. The atoms k could be found quickly with a neighbor search grid, but it turns out that the trivial approach to just look at covalently bound atoms is good enough.

(3) Finally, the CPU loops over the pixel lines in the temporary Z-buffer of atom i, checks which lines are af-

fected by intersections and emits a number of triangles that trace these intersections. The principle is clarified in Figure 1C, which shows how to draw a water molecule with just 14 instead of 960 triangles.

(4) If atoms are shown as sticks or balls&sticks, cylinders need to be drawn that connect the atoms (Figure 1E). To reduce the polygon count, only the front side of the cylinders is drawn, using between 2 and 18 triangles, depending on the distance from the viewer. Cylinders always use the same texture as the atoms (Figure 1B), which ensures visual continuity.

(5) Shadows and ambient lighting are calculated per atom, not per pixel. The CPU first draws a low resolution depth map of the scene where atoms have a diameter of just 15 pixels (Figure 1D), either seen from the position of the light source (shadows) or from the six main directions (ambient lighting). Then it integrates the amount of light reaching each atom (i.e. the fraction of pixels not occluded by closer ones) and darkens the atom accordingly (using either GL_EXT_fog_coord or multi-texturing).

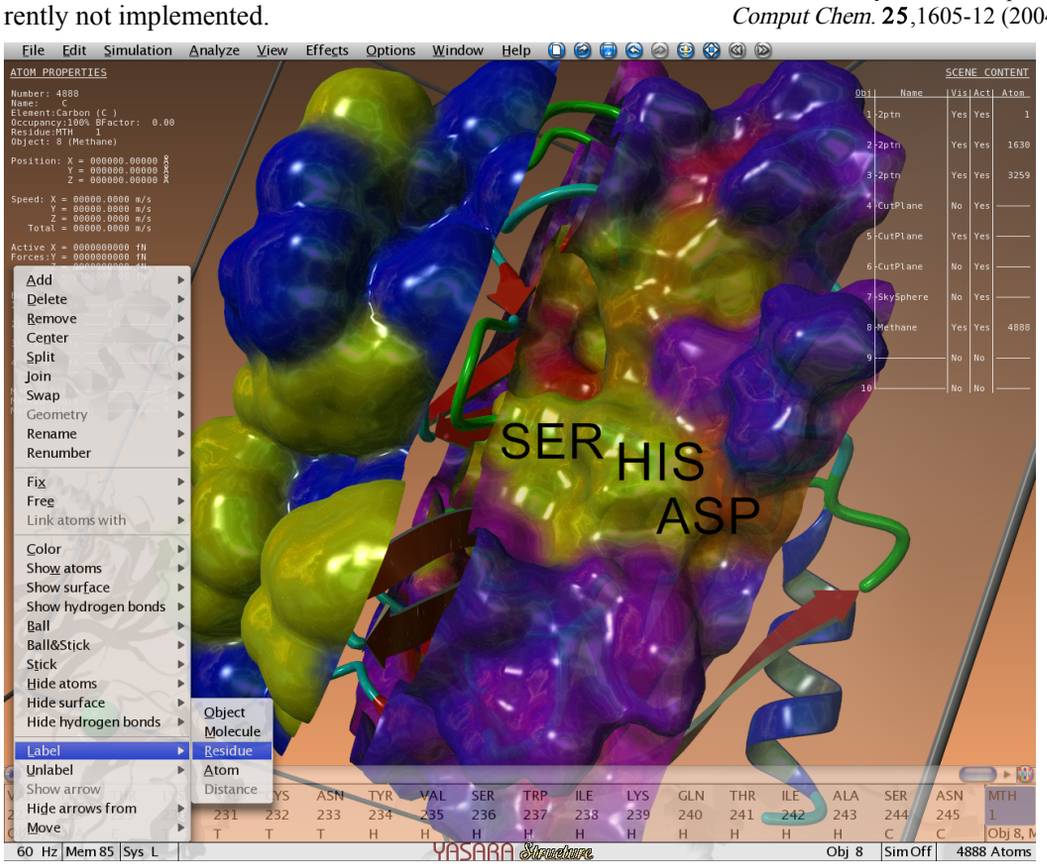
This fast way of drawing molecules also has three limitations compared to the classic approach: First, atom colors must be mixed from two of the standard colors present in the texture (Figure 1B), which allows to create most useful colors, but not all colors. Second, the maximum atom size on screen is limited to the largest atom size in the texture (currently 256*256 pixels), unless one wants to use lower quality upscaled atoms. To prevent atoms from getting too small, YASARA therefore restricts its window size to Full HD, but we plan to double the texture size to 2048*2048 soon, covering 4K and similar hires displays. And third, drawing transparent atoms is not straightforward and currently not implemented.

## Results

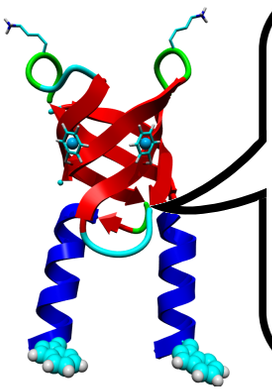
A visualization example for the chaperonin GroEL/ES (1AON, 58884 atoms) with real-time shadows is shown in Figure 1F. On a 240 EUR Motorola Razr i smartphone (Intel Atom Z2480@2GHz with two threads and PowerVR SGX 540, 960x540 pixels, Android 4) the algorithm reaches 4-12 frames per second, depending on the number of atoms on screen (or 5-30 fps with ambient lighting but no shadows). This is about 10x as fast as other popular apps (which do however not support shadows). On a Windows 8 tablet with the faster Atom Z2760 CPU@1.8 GHz, four threads, 1366x768 pixels and PowerVR SGX 545, the frame rate ranges from 8 to 15 fps (12-30 fps without shadows, about 6x as fast as others). On a high-end workstation, the frame rate is usually above the refresh rate of the screen (60 Hz) for all but the largest structures (ribosomes etc.). We separately tested the usability for interactive MD (not in YASARA View) and obtained 4 fps on the Motorola Razr i for DHFR in water (23788 atoms), 7.9 A VdW cutoff and PME electrostatics, just enough to pull atoms around.

*Acknowledgements:* YASARA View is part of the NewProt project (www.newprot.eu) that is funded by the European Commission within its FP7 Programme, under the thematic area KBBE-2011-5 with contract number 289350. We also thank the YASARA users for their invaluable feedback and financial support.

1. Levinthal, C. Molecular model-building by computer. *Scientific American* **214**, 42-52 (1966).
2. Humphrey, W., Dalke, A., Schulten, K. VMD: visual molecular dynamics. *J.Mol.Graph.* **14**, 27-28 (1996).
3. Pettersen, E.F., Goddard, T.D., Huang C.C., Couch, G.S., Greenblatt, D.M., Meng, E.C., Ferrin, T.E. UCSF Chimera - a visualization system for exploratory research and analysis. *J Comput Chem.* **25**,1605-12 (2004).
4. DeLano, W.L. The case for open-source software in drug discovery. *Drug Discov.Today* **10**, 213-7 (2005).
5. Tarini, M., Cignoni, P., Montani, C. Ambient Occlusion and Edge Cueing for Enhancing Real Time Molecular Visualization. *IEEE Transactions on Visualization and Computer Graphics* 12,1237-44 (2006).




**Space filler:** Screenshot of a Ser-His-Asp catalytic triad with residues colored by sequence conservation (right half) and crystal contacts (left half) via View > Color > Residue by feature.



In a molecular dynamics simulation, the computer creates a tiny little universe, which is confined by the walls of the simulation cell. Since the walls introduce problematic boundary effects like surface tension, one usually turns them into 'teleporters', so that atoms leaving the cell on one side come back on the other side (periodic boundaries). Time passes in small steps of 1-5 femtoseconds. In each step, the computer uses the current force field (like AMBER) to calculate the forces acting on the atoms, obtains the resulting acceleration from F=m*A, calculates the change of velocity and moves the atoms forward. Various tricks to run these simulations faster are described in this chapter.



# New ways to boost molecular dynamics simulations

Elmar Krieger and Gert Vriend

*J.Comput.Chem.* **36**, 996-1007 (2015)


## Abstract

We describe a set of algorithms that allow to simulate dihydrofolate reductase (a common benchmark) with the AMBER all-atom force field at 160 nanoseconds/day on a single Intel Core i7 5960X CPU (no GPU, 23788 atoms, PME, 8.0 Å cutoff, correct atom masses, reproducible trajectory, CPU with 3.6 GHz, no turbo boost, 8 AVX registers). The new features include a mixed multiple time-step algorithm (reaching 5 fs), a tuned version of LINCS to constrain bond angles, the fusion of pair list creation and force calculation, pressure coupling with a 'densostat', and exploitation of new CPU instruction sets like AVX2. The impact of Intel's new transactional memory, atomic instructions, and sloppy pair lists is also analyzed.

The algorithms map well to GPUs and can automatically handle most PDB files including ligands. An implementation is available as part of the YASARA molecular modeling and simulation program from www.YASARA.org.


## Introduction

Molecular simulations with empirical force fields like AMBER[1], CHARMM[2] or OPLS[3] are enjoying a phase of enthusiastic interest, thanks to the arrival of personal supercomputers, i.e. graphics processing units (GPUs) that can accelerate science equally well as video games. As shown by AceMD[4] and OpenMM[5], classical force fields are ideally suited for GPUs, because the calculations mainly require single precision floating point operations - which are the GPU's home game.

While the slow transfer of data between CPU and GPU initially led to the development of programs that perform all computations on the GPU and let the CPU run idle, this trend seems to reverse recently. CPU and GPU are increasingly often fused on the same chip with unified memory (AMD Kaveri, Intel Iris), solving the data transfer problem. Additionally, modern CPUs contain powerful vector instructions sets (SSE, AVX), which are too valuable to be left unused. Consequently, the GROMACS team recently achieved a very high MD performance by using CPU and GPU in parallel. We therefore believe that MD simulations are best approached with a capable 'home base' on the CPU, which can handle the countless complications in real-life applications (like knowledge-based force fields[6], X-ray[7] and NMR refinement[8]) and offloads tasks to the GPU when beneficial. In this work, we focus on this home base and describe a number of algorithms to generally improve simulation performance, and we benchmark them on a single Intel Core i7 CPU with AVX2. Most of the algorithms are equally well suited to accelerate simulations using multiple CPUs and GPUs.

While simulation performance is usually considered less important than accuracy (which we focused on previously[6,9]), only fast simulations allow an important accuracy check: whether the force field can reproduce folding and structural changes of proteins or not[10].

## Results & Discussion

### A mixed multiple time-step algorithm

Raising the integration time-step to boost MD simulation speed tends to reveal its disadvantages after a few hundred picoseconds - when the simulation suddenly blows up. Such blow-ups originate from atoms that vibrate with high speed and accidentally experience a larger than usual force, which accelerates them to the point where the distance traveled per time-step is so large, that reliable integration is no longer possible. The vibration then becomes self-enforcing, until atoms jump around randomly, 'infecting' others and the simulation explodes. For a given force, the acceleration is inversely proportional to the atom mass, which puts hydrogens most at risk. In our experience, hydrogen bond vibrations blow up if the time-step exceeds about 1.75 fs, hydrogen angle vibrations become critical at 2.5 fs (especially when the time-step for non-bonded forces is larger), and heavy atom bond vibrations around

3.5 fs.

Four solutions to deal with these vibrations are commonly used: First, one can simply increase the hydrogen masses, which slows down the vibrations[11]. Second, one can integrate the vibrations more accurately by using multiple time-steps[12]: a large time-step for the slowly varying intermolecular forces, and a smaller time-step for the quickly varying intramolecular forces (including the most critical bond and angle vibrations). The stability of this approach can be improved in various ways, e.g. with the mollified impulse method[13] used in NAMD[14]. Third, one can totally remove bond vibrations by constraining the bond lengths using algorithms like LINCS[15] or SHAKE[16]. And finally, one can remove hydrogen angle vibrations by 'virtualizing' the hydrogen atoms[11] (i.e. by treating them as dummy atoms without mass, whose position is recalculated from the heavy atoms at each step).

We combine solutions two and three in a new way, which is shown in Figure 1. While a normal multiple time-step algorithm uses the same time-step for all atoms and a certain recipe to apply the forces (e.g. the quickly varying bonded forces at each step, and the slowly varying non-bonded forces only at every other step), we mix it with a single time-step algorithm depending on the molecule type. Small molecules whose internal degrees of freedom can be removed by applying constraints are propagated with a large single time-step (up to 5 fs). Since all bonds and angles in such an internally frozen molecule are at their equilibrium values, the corresponding forces are zero and need not be calculated, only the non-bonded interactions are required. For all the other molecules, a multiple time-step algorithm is used.

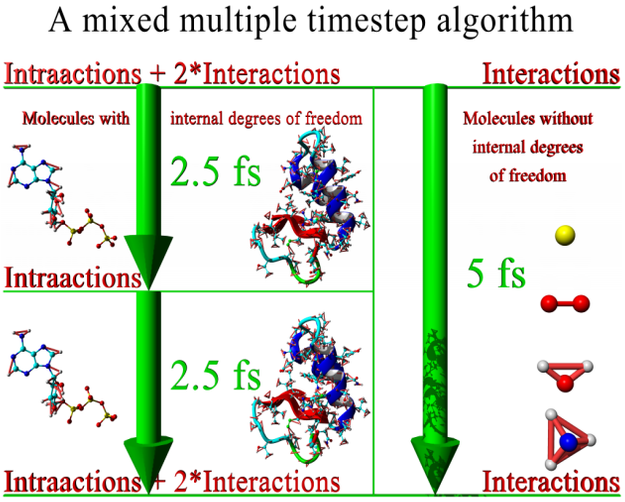



**Figure 1:** Molecular dynamics simulation by mixing a multiple time-step algorithm (left) with a normal single time-step algorithm (right) depending on the molecule type. The multiple time-step algorithm uses a pulsed approach, which calculates bonded intraactions at each 2.5 fs step, and adds the non-bonded interactions (scaled with 2) only every other step, in sync with the 5 fs single time-step algorithm.

This approach has three advantages: First, it is easy to implement, while virtual hydrogens are rather complicated to handle (they require elaborate code for each hydrogen configuration, so that it is often not possible to simulate organic molecules 'out of the box', especially if they contain less common hydrogen configurations). Second, it does not require to change hydrogen masses (like the virtual hydrogens with zero mass or the heavy hydrogens). While it is certainly true that the effect of changing hydrogen masses is either small (compared to the errors inherent in empirical force fields) or completely absent (when looking at thermodynamic properties, which do not depend on atom masses[11]), we simply consider it convenient not having to think about the potential impact on a case-by-case basis. And third, it improves performance compared to normal multiple time-step algorithms, which need to move all atoms in several costly integration steps. As shown in Figure 1, the majority of atoms (typically waters) require only a single integration step.

Care must be taken when choosing the multiple time-step recipe because of its impact on energy conservation and simulation accuracy. In an extensive comparison study, Grubmueller et al.[17] analyzed several different multiple time-step schemes, some of which even extrapolate the non-bonded forces from the current and previous forces. In comparison, our setup shown in Figure 1 is rather simple: there are no distance classes (all the non-bonded Van der Waals and Coulomb interactions are calculated together with a 5 fs time-step), and there is only a single step in between (when just bond, angle and dihedral intraactions are calculated). For this simple case, we found that the method they named DC-i yielded the most stable trajectories: the non-bonded forces are doubled in the even steps, and totally ignored in the odd steps in between (also called the 'impulse method'[17] or 'Verlet-I'[18]). Adding non-bonded forces every second step is still in the safe range of the impulse method, which has as advantage that it always uses exact forces that match the atom positions.

## A tuned version of LINCS to constrain bond angles

Figure 1 illustrates our goal to integrate bonded intraactions with a 2.5 fs time-step, and this means that vibrations of bonds and angles involving hydrogens need to be constrained. We use the very elegant LINCS algorithm[15], which employs a power series expansion to invert the constraint coupling matrix and to determine how to move the atoms such that all constraints are satisfied. Unfortunately, this fast approximate inversion only works as long as the simplified coupling matrix (which has zeroes along the diagonal) is sparse enough, because all absolute eigenvalues must be smaller than 1. When two constraints involve the same atom, the corresponding element in the coupling matrix becomes non-zero, so the sparsity shrinks as the connectivity between constraints grows. Consequently, the LINCS authors noted that their approach works fine for constraining bonds (also in rings), but adding angle constraints quickly raises the eigenvalues above 1, which creates a need for virtual hydrogen sites. Our alternative solution to this problem works as follows: Thanks to the mul-

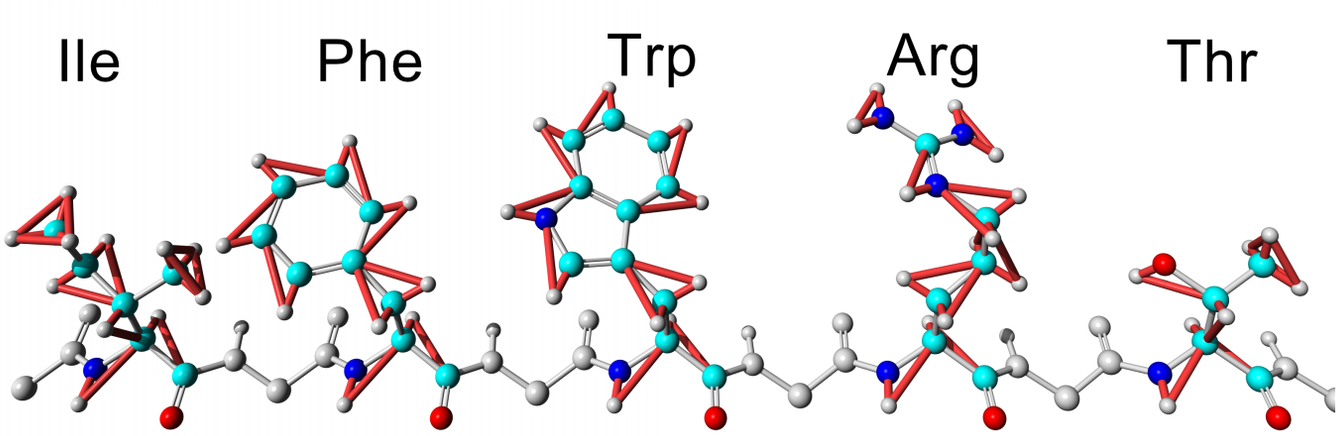



**Figure 2**: Placement of constraints to reduce the degrees of freedom of hydrogen atoms and enable larger time-steps. The constrained distances and angles are shown for five exemplary amino acids (orange in the electronic version). The algorithm that decides which angles to constrain such that the coupling matrix can still be inverted quickly is not specific for amino acids (see Materials & Methods).



tiple time-step algorithm, we can integrate the bonded interactions with a 2.5 fs time-step (instead of the overall 5 fs time-step). With 2.5 fs, bonds between heavy atoms are not yet critical and do not need to be constrained, which increases the sparsity of the simplified coupling matrix almost to the point where enough hydrogen angle constraints can be added to permit stable simulations (Figure 2). Constraining all hydrogen angles is not possible, but also not necessary, since a single angle constraint per hydrogen is usually enough. We wrote 'almost' and 'usually', because there is one exception: if a heavy atom has three hydrogens bound (e.g. $-CH_3$, $-NH_3^+$ groups), adding constraints in a way that treats each atom equally yields a tetrahedron of six constraints (three bonds and three angles). The largest eigenvalue of the corresponding simplified coupling matrix is unfortunately 1.35. We therefore implemented a version of LINCS that handles this special case by inverting the 6x6 coupling matrix exactly. This requires only a few hundred CPU cycles and has no noteworthy impact on performance. A heavy atom with four hydrogens (e.g. methane) on the other hand is easy to handle again, because one can simply add two angle constraints between pairs of hydrogens (largest eigenvalue 0.82). The algorithm to decide which angles to constrain is explained in the Materials & Methods section.

One might wonder why two constraints per hydrogen are enough – after all, they cannot prevent vibrations perpendicular to the plane spanned by the constraints. The reason is that with a 2.5 fs time-step, not all directions are critical yet – mostly those where other hydrogens separated by four covalent bonds are close by and exert strong forces. These critical directions are protected with constraints, for example by placing angle constraints along a chain of CH2 groups, instead of constraining just the H-C-H angle (as shown for the Arg side-chain in Figure 2).

Additional angle constraints generally yield larger eigenvalues, which in turn require to increase the accuracy of the LINCS algorithm to keep the constraints satisfied. Apart from trivial adjustments (like doubling the LINCS expansion order), we had to tweak the algorithm for single precision calculations. Water molecules are handled with the analytic SETTLE algorithm[19].

A convenient aspect of our approach is that coupled constraints form small groups only. Since bonds between heavy atoms are not constrained, these groups don't extend over the entire protein, they usually don't even cross residue boundaries (Figure 2). Consequently, special considerations regarding workload distribution (like those described for P-LINCS[20]) are not needed when parallelizing the algorithm.

## Mixing pair list creation and force calculation

The algorithms described here have been implemented in our molecular modeling and simulation program YASARA[21]. While there is an optional text mode interface to be run on servers, a major goal has always been to visualize the simulation on screen, allowing to dive into the system and pull atoms interactively. When we implemented this feature in 1997, CPUs were rather weak, and in order to provide a smooth interactive MD experience, YASARA did not use pair lists (i.e. arrays containing the non-bonded interaction partners for each atom). A simulation with pair lists consists of a slow step (which includes pair list creation) and a series of fast steps (using the pair lists). Such an alteration of slow and fast steps caused stutter on the screen during interactive MD runs. To ensure that each simulation step takes an equal amount of time, a grid-based neighbor search was done at each step, intertwined with the non-bonded force calculation, so that no pair lists were needed. To maximize the performance of the grid-search, the grid cubes should be small enough to provide a decent approximation of the cutoff sphere, and at the same time large enough to avoid useless tests of empty cubes. We obtained best performance using a grid spacing of cutoff/3 for cutoffs below 9.5 Å, and cutoff/4 above. Figure 3 shows a neighbor-search example for an 8 Å cutoff, i.e. the search space extends seven cubes along each axis. Compared to a cubic neighbor-search volume of 7*7*7 cubes, four cubes can be skipped in each of the eight corners, reducing the search space by 10%.

Apart from the constant execution time (which was im-

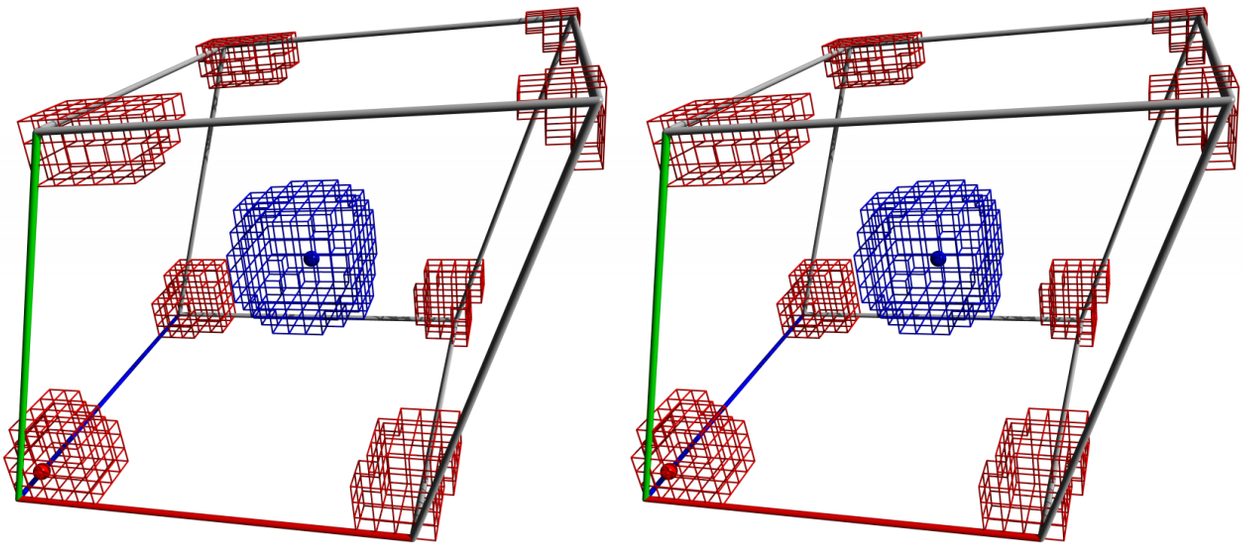


**Figure 3**: Two examples of grid-based neighbor search in a triclinic simulation cell shown in stereo. The grid spacing is 'non-bonded force cutoff/3', so that seven cubes need to be searched along each axis to find the neighbors of the atom in the center, but four cubes can be skipped in each of the eight corners. The atom in the bottom left corner is close to three periodic cell boundaries, so cubes on the other sides must be considered too, resulting in a rather complex search pattern, which can of course be precalculated. .

portant in 1997, but has only a marginal impact on today's hardware) an algorithm that works without pair lists has two performance advantages: First, it is trivial to make the algorithm store pair lists on the fly to be used in the next steps. The resulting pair list-based algorithm executes faster than the usual approach of first creating the pair lists and then calculating the forces, which requires to load the atom coordinates twice, possibly from slow main memory. And second, if the user wants to update the pair list at every step, one can totally skip the pair list creation.

Why would a user choose to update the pair list at every step? Apart from the interactive visualization purposes mentioned above, for example to make sure that no interactions within the cutoff distance are missed. While the literature often mentions an idealized pair list, which is created for a somewhat larger distance **cutoff+x** and updated whenever the first atom traveled further than **x/2**, this approach is rather slow in practice, because it updates the pair list more frequently than really needed. As noted by the GROMACS team (user manual 4.6, chapter 3.4.2), it is much more efficient to just use the pair list for a certain number of steps, eventually missing some interactions within the cutoff while the atoms move. The energy drift associated with such 'sloppy pair lists' can be reduced at will by increasing the pair list cutoff used during neighbor search, calculated for example from this empirical formula:

$$Cutoff_{PairList} = Cutoff_{Force} + \frac{\sqrt{UpdateFrequency - 1} * TimeStep * Temp * 0.001}{M} \quad (1)$$

Our formula estimates how far particles with an averag velocity proportional to 'Temp/M' travel between pair list updates that are done every 'UpdateFrequency' steps, given the current simulation 'TimeStep'. 'M' is the average particle mass in the simulation, water molecules are treated as single particles. We assume that particles get either closer or further away at each step, which boils down to a one-dimensional random walk, for which the expected travel distance is proportional to the square root of the number of steps. The empirical proportionality constant 0.001 Å*Dalton/(Kelvin*fs) must be chosen as small as possible (since it increases the pair list cutoff and thus reduces the performance) and as large as necessary to reduce the energy drift to an acceptable value.

## Evaluation of simulation accuracy

Molecular dynamics simulations can be very sensitive to the protocol and algorithms used, especially when events that occur on longer time scales are investigated, like protein folding or membrane formation. Every new approximation made to gain performance must therefore be analyzed with great care. Our solution to this problem is trivial – we avoid new approximations. The methods described here either allow to calculate the same forces faster (and are thus not approximations), or recycle old approximations that have become common scientific practice. Any problems with these approaches would thus have been discovered by now. These approximations are: reducing the degrees of freedom of hydrogen atoms (e.g. with LINCS[15] and virtual hydrogen sites[11]), sloppy pair lists (used and tested extensively in GROMACS[22]), a time-step of 5 fs for the non-bonded interactions (the default in GROMACS when using virtual hydrogen sites), and the Verlet-I[18] multiple time-step algorithm (used extensively in NAMD[14]). Most simulations were run with an 8 Å cutoff for Van der Waals- and direct space Coulomb forces, since this is a common choice, also when posting DHFR benchmark results (e.g. by the AMBER[23] and OpenMM[5] developers). This does not imply that an 8 Å cutoff is ideally suited for all purposes[24].

The first thing to check is that the actual implementation is reliable and not suffering from problematic energy drifts, especially since it makes extensive use of single pre-

| Cutoff (Å) | Pair list update frequency | Interaction time step (fs) | Intraaction time step (fs) | Constraints | Energy drift ($k_BT$/ns) |
|---|---|---|---|---|---|
| 9.6 | 1 | 1.0 | 1.0 | no | 0.009 |
| 9.0 | 1 | 1.0 | 1.0 | no | 0.010 |
| 8.0 | 1 | 1.0 | 1.0 | no | 0.011 |
| 8.0 | 1 | 2.5 | 1.25 | no | 0.005 |
| 8.0 | 1 | 5 | 2.5 | H bonds and angles | 0.018 |
| 8.3 / 8.0 | 10 | 5 | 2.5 | H bonds and angles | -0.006 |

**Table 1**: Energy drift per nanosecond and degree of freedom during a simulation of DHFR. Time-steps for non-bonded interactions and bonded intraactions are listed separately. The first two rows list values in the 9 Å cutoff range to facilitate comparison with drifts reported for other MD programs[22]. The last row uses sloppy pair lists updated every 10 steps, and a larger pair list cutoff of 8.3 Å obtained using equation 3 with an average particle mass of 14.55 Dalton.



cision calculations, which are much faster on CPUs as well as GPUs (Table 1). Intuitively, one would expect the energy drift to increase with each approximation made, but this is not the case. Drifts are expressed per nanosecond and not per integration step, so if a significant part of the drift is caused by the integration procedure itself, then one can reduce the drift by increasing the time-step (since fewer integration steps are needed per nanosecond). This allows a mixed multiple time-step integrator with 2.5 fs (0.005 in row 4) to outperform a single time-step integrator with 1 fs (rows 1-3). Of course this principle no longer holds when the time-step is increased further and starts to dominate the drift (row 5). It is also noteworthy that inaccuracies do not always cause positive drifts. Sloppy pair lists for example cause a negative drift, which can be adjusted at will by shifting the pair list cutoff. The resulting small negative drift in row 6 is thus due to cancellation of errors, and could be made zero or positive by increasing the 8.3 Å pair list cutoff, i.e. the empirical constant '0.001' in formula 1. One should thus emphasize the importance of listing energy drifts in a step-wise manner, while enabling the various acceleration methods, so that the real accuracy is obvious and no cancellation of errors goes unnoticed.

The various approximations listed above have been described and validated in separate articles, using different methodologies. To facilitate a direct comparison, we tested each approach with an accuracy benchmark described previously[9]: simulating 25 protein crystals with the AMBER03 force field[23] and calculating the average RMSD from the starting structures. Using complete crystallographic unit cells ensures that all forces giving rise to the X-ray structures are present, and RMSDs really depend on simulation accuracy and not on differences between crystal and solution environments[9]. The reference simulation (Figure 4, blue) was run at the temperature of the experimental structure determination (as specified in the PDB header, on average 176K) with PME electrostatics[25], a 10.5 Å cutoff for Van der Waals (VdW) and direct space electrostatic forces, a single 1 fs time-step and no constraints. Raising the temperature to the standard 298K heavily increased the average RMSD during the last quarter of the simulation from 0.55 Å (blue) to 0.83 Å (magenta). As expected, the commonly used MD approximations had no significant impact on the RMSD: reducing the cutoff to 8.0 Å (red), additionally increasing the time-step to 2.5 fs and using the impulse method (1.25 fs for bonded intraactions, orange), adding hydrogen constraints and doubling the time-step to 5 fs (green, Figure 1), and updating the pair list only every 10th step (cyan).

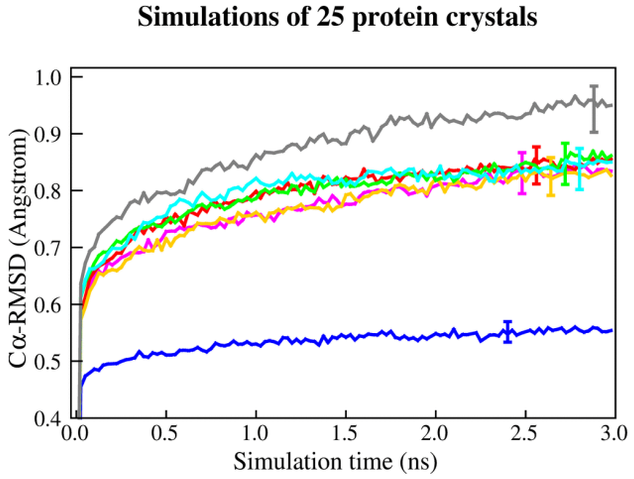



**Figure 4**: Accuracies of seven simulation protocols, measured as the average Cα RMSD of 25 protein crystal simulations with the AMBER03 force field. Error bars indicate the average and standard deviation of the RMSD during the last quarter of the simulation, averaged over the 25 simulations. The blue simulations are run at the temperature of experimental structure determination (on average 176K), while the others are run at 298K. The reference protocol (blue,magenta) uses PME electrostatics, a 10.5 Å VdW and direct space electrostatic cutoff, a 1 fs time-step, no constraints, and no pair list. The other protocols improve performance with a 8.0 Å cutoff (red), additionally a 2.5 fs time-step (orange), or a 5 fs time-step with constraints (green) and also sloppy pair lists (cyan) and have no significant impact on the RMSD, unlike a further cutoff reduction to 5.0 Å (gray). The actual performance of these protocols is shown in Figure 8.

The simulations in Figure 4 are computationally expensive, because they involve 175 different trajectories, some with very slow protocols. Extending the simulation time would not improve the benchmark result, since the RMSD from the starting structures is only a useful accuracy indicator during the initial phase of a simulation – in the long

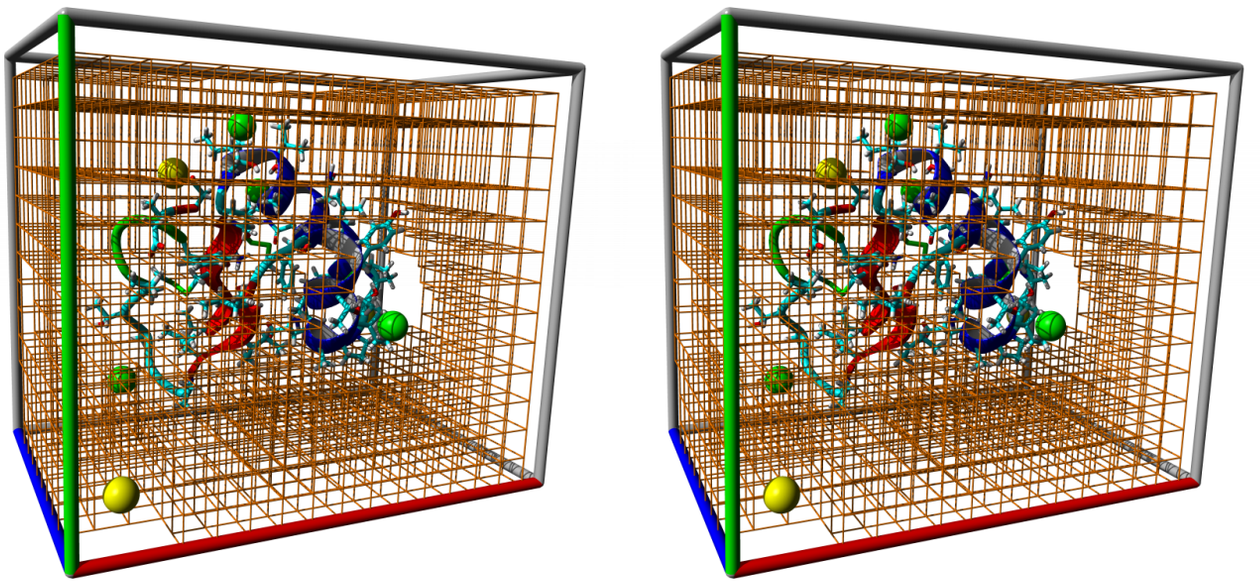


**Figure 5**: Stereo image of crambin, ions, and the grid of cubes that contain just water and can be used to quickly calculate the current water density to rescale the cell with a 'densostat'. The grid was obtained by excluding the 3*3*3=27 cubes around each non-water atom, which mostly covers the first hydration shell. The ions shown correspond to a physiological NaCl concentration of 0.9% (154 mM). At this concentration, ~78% of the solvent cubes can be included in the density calculation. The fraction of useful cubes drops slowly with increasing NaCl concentration, reaching 50% at about 4.5% NaCl (770 mM). We also tested the exclusion of larger parts of the hydration shells (57 cubes around each non-water atom, i.e. the big cube of 3*3*3 cubes above, plus a cross of 5 cubes on each of its six sides), and found that the accuracy of the density calculation did not improve significantly, while considerably fewer cubes could be used (70% at 0.9% NaCl, reaching 50% already at ~2.3% NaCl). In case of very high salt concentrations or mixed solvents, where not enough water cubes are available, one can of course run a reference simulation with a barostat to measure the density of the entire solvent and plug that value into the densostat, which is then based on solvent cubes instead of water cubes.

term, proteins would undergo temporal partial unfolding[26] and randomize the RMSDs. Fortunately, problems with simulation accuracy tend to show up early, as demonstrated for a cutoff reduction to 5.0 Å (gray).

## Pressure coupling without virial calculation

The most common way to run an MD simulation is the 'real-life' NPT ensemble, where the number of particles, the pressure and the temperature are kept constant. While the current temperature can simply be calculated from the atom velocities, the pressure is not trivial to handle. The most common approach is to calculate the pressure 'P' using a formula derived from the Clausius virial theorem:

$$P = \frac{2}{3 * Volume} * \left( KineticEnergy + \frac{1}{2} * \sum_{i=1}^{Atoms} Position_i * Force_i \right) \quad (2)$$

Unfortunately, the resulting pressures fluctuate strongly (by hundreds of atmospheres at each step), and even if the time average pressure is used to rescale the cell, one arrives at densities that are a bit off. For example the density of water at 298K using PME electrostatics was reported[27] to be 0.979 g/ml instead of the expected 0.997 g/ml. Apart from changing the water model[27], these discrepancies can be dealt with in two ways: One favors the density and applies corrections to the pressure (e.g. to account for the truncation of attractive VdW forces at the cutoff[28]), getting closer to the right density. The other favors the pressure and argues against corrections, because they might have a negative impact (if the cutoff for VdW interactions makes waters 'happy' at 0.979 g/ml, then compressing them to 0.997 g/ml could make them 'feel stressed').

Our approach does not choose a side, but lets the user decide. It is based on an assumption that is implicit in all pressure coupling protocols with cell rescaling - that pressure is not a localized property, but spreads through the cell. The pressure in the left half of the cell is the same as in the right half, the pressure in the solute is the same as in the solvent. This implies that one only needs to know the solvent pressure, which allows to take the shortcut shown in Figure 5. Having placed a grid in the simulation cell, all grid cubes that contain just water atoms (and have 26 neighbors with just water atoms) are tagged. Then the masses of the water atoms in the tagged cubes are summed and divided by the volume to obtain the current water density. The grid is simply the neighbor search grid with ~2.6 Å spacing (Figure 3). Contrary to the virial, the water density shows only small variations (whenever atoms change a cube), so that it's enough to calculate it every 10 steps and then average 50 measurements to obtain a stable result that can be plugged into a 'densostat' (similar to a Berendsen barostat[29]), which defines a scaling factor 'S' for the atom coordinates to reach the desired water density:

$$S = \max\left(0.999, \min\left(1.001, \sqrt[3]{1 + C * \left(\frac{Density_{measured}}{Density_{set}} - 1\right)}\right)\right) \quad (3)$$

The 'max' and 'min' functions make sure that the cell is never scaled by more than 0.1%, even if the user chooses a large coupling strength 'C' or if the density difference is large (we found 0.1% to be a reasonable limit to avoid a temperature rise caused by scaled bond lengths). The time

| Force field | Pressure coupling | Random seed | Average B-factor (Å$^2$) | Average B-factor difference from first row (Å$^2$) | Cell volume (Å$^3$) with 23788 atoms |
|---|---|---|---|---|---|
| AMBER03 | Barostat | 0 | 78 | 0±0 | 240645±218 |
| AMBER03 | Barostat | 1 | 60 | –18±59 | 240700±220 |
| AMBER03 | Densostat | 0 | 83 | 5±48 | 240560±342 |
| AMBER99 | Barostat | 0 | 104 | 26±58 | 240744±223 |

**Table 2**: Influence of force field, pressure coupling method and random number seed on the B-factors and cell volume extracted from 150 ns simulations of DHFR in solution at 298K (at room temperature and without crystal packing, B-factors are higher than those found in X-ray structures).

average density is used so that fluctuations are not artificially suppressed (which can be a problem with weak coupling methods, especially if the simulated system is small[30]).

Our 'densostat' has two advantages: First, it lets simulations run about 8% faster. The reason is that the virial calculation - even though it looks fast and simple in equation 1 - requires special care when handling forces that cross periodic boundaries, which effectively pulls it into the inner loops of the force calculation. And second, the densostat makes it trivial to reach exactly the right density (if that's desired). Those who prefer the right pressure can simply run a reference simulation of water with a barostat, and their favorite cutoff and temperature, and use the resulting water density as the densostat target value.

The disadvantage is obvious: since the density is a scalar, the densostat fails when the pressure cannot be expressed as a scalar, i.e. when the three values along the trace of the pressure tensor deviate from each other. This happens when the solute spans the entire cell, so that solvent molecules cannot travel freely to spread the pressure uniformly. The most common examples are proteins embedded in a membrane or protein crystals. These need to be handled the classic 'virial' way, using different scaling factors for each cell axis. We do not claim that the densostat can replace the virial calculation in all the other applications of molecular dynamics simulations, but we found no influence on the dynamics of the simulated system. Table 2 shows an analysis of atomic B-factors extracted from 150 ns simulations of DHFR in solution at 298K (using the cyan protocol in Figure 4). With a barostat (Materials&Methods), the average heavy atom B-factor was 78 Å$^2$. Running the same 150 ns simulation a second time, but with a different random number seed for the initial velocities, yielded heavy atom B-factors that differed on average by –18±59 Å$^2$. With the densostat, B-factors differed by 5±48 Å$^2$, so the densostat had no larger impact than the random number seed. However, after changing the force field from AMBER03 to AMBER99[31], B-factors differed by 26±58 Å$^2$. Also the cell volume during the last 75% of the simulation was not significantly different.

These data reflect the common observation that simulation time-scales of proteins are usually too short to reach exhaustive sampling, so that the results often depend considerably on the initial conditions. This dependence can easily mask variations in the simulation protocol, like the switch between barostat and densostat shown in Table 2.

To verify that the densostat also works for small systems, which can be sampled well and thus show a stronger dependence on the simulation protocol, we ran microsecond simulations of a well known model system - the alanine dipeptide - and extracted the free energy landscape.

Early simulations of the alanine dipeptide lasted a few nanoseconds and required tricks to enhance sampling, like

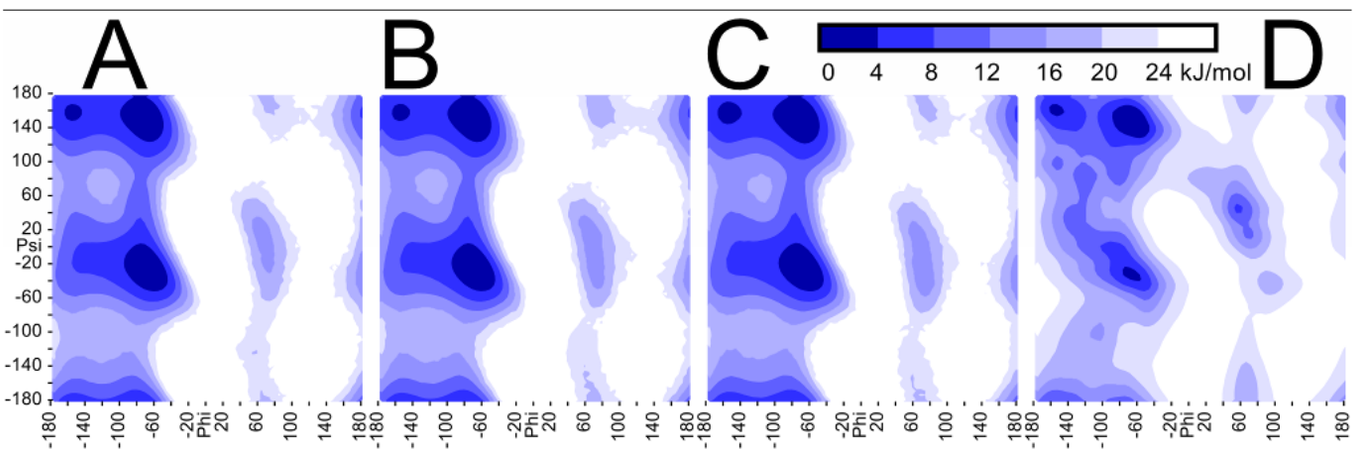



**Figure 6**: The free energy landscape of the alanine dipeptide, showing the φ/ψ map of the first alanine. **A**: Microsecond simulation at 298K with a conservative protocol (single 1 fs time-step, exact pair list updated each step, no constraints, barostat with 1 bar). **B**: The same simulation again, but with different random seed and different initial conformation of the peptide. **C**: Simulation with our fast protocol (multiple 5 fs time-step, sloppy pair list updated every 10 steps, bond- and angle constraints, densostat with 0.97 g/ml water density[27]). **D**: For comparison, the corresponding φ/ψ map of alanine, extracted from high-resolution X-ray structures in the PDB, with an additional round of smoothing to provide knowledge-based torsion forces in the YASARA force field[6]

thermodynamic integration of perturbed trajectories[32]. Fortunately, the microsecond simulations possible on today's hardware provide sufficient sampling - the plot of the φ/ψ free energy map shows only marginal dependence on the initial conditions (Figure 6, A and B). Enabling the fast simulation methods described here (including the densostat) yields comparable differences (Figure 6C). In contrast, Figure 6D shows the free energy map of alanine, extracted from high-resolution X-ray structures in the PDB. Differences are of course expected, but we note that the positions of the energy minima match well.

## Multi-threaded force calculation methods and overall performance

We evaluated the performance of the various methods by running the well-known DHFR benchmark, a simulation of dihydrofolate reductase (3158 protein atoms and 20628 water atoms) with PME electrostatics[25]. Today's CPUs can execute a steadily growing number of parallel threads, so performance depends to a large extent on the way the non-bonded force calculation is distributed. We analyzed four different approaches, all of which distribute atoms among threads by dividing the cell into 'thread regions', from left to right. Atoms are reordered such that those in the same region are stored next to each other in memory (which optimizes cache usage and non-unified memory access (NUMA)). The easiest approach is shown in Figure 7A. It is based on a single table of atomic force vectors, one for each atom, which is shared among the threads. Each thread calculates and adds the forces acting on its atoms. If an interacting atom belongs to the same thread, the force is immediately subtracted (i.e. added in reverse direction) there too. Otherwise the force needs to be calculated twice (once by each thread), because one thread cannot simply change forces belonging to other threads (which would cause a 'race condition').

With the most conservative simulation parameters (10.5 Å VdW and direct space electrostatic cutoff, 1 fs time-step), using the AVX instruction set and a classic virial-based barostat, method A yields 7.5 ns/day on an Intel Core i7 5960X CPU with 16 threads and 3.6 GHz (Figure 8, magenta). If the cutoff is reduced to 8.0 Å, performance increases to 13.3 ns/day (red). Activating the mixed multiple time-step without constraints (2.5 fs) yields 30.2 ns/day (127% more, orange). Constraints (5 fs time-step) almost double the speed (56.9 ns/day, green), and sloppy pair lists add 52% to 86.7 ns/day (cyan).

Method B (Figure 7B) uses two force tables. Forces are always calculated once and added in force table 1. If the interacting atom belongs to the same thread, the force is also subtracted in force table 1, otherwise it is atomically subtracted in force table 2. 'Atomic' means that the subtraction cannot be interrupted by another thread, avoiding race conditions. This is achieved by acquiring a simple spinlock (using the x86 instruction sequence 'loop: xor eax,eax/pause/lock cmpxchg/jnz loop'),performing the force subtraction, and releasing the lock. We use one lock per atom, which is placed as the fourth element of the force vector (a force vector normally needs 3*4 bytes storage, but SSE requires 16-byte alignment, so a fourth element is included for padding). Both tables need to be added in the end. Method B increases performance by 32% to 114.1 ns/day.

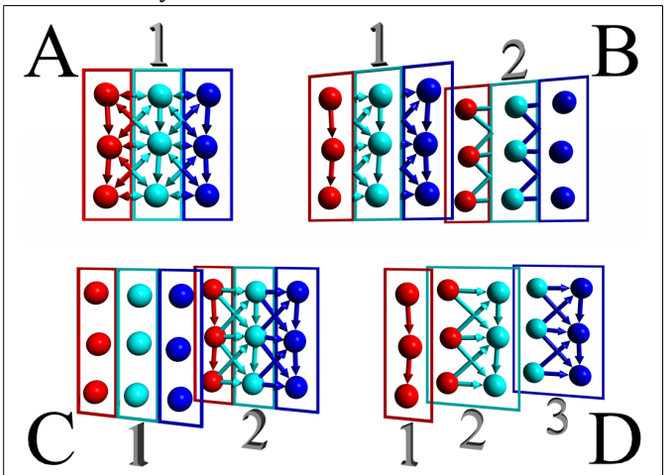



**Figure 7:** Four methods to parallelize non-bonded force calculation. In this example, three threads (red, cyan, blue) calculate the forces acting on nine atoms. Each arrow corresponds to one interaction within the cutoff range (which includes direct and diagonal neighbors). The arrow color shows which thread calculates the force, which is added to the atom pointed at. If the arrow has a single head, the force is also subtracted from the other atom (and thus needs to be calculated once only). If the arrow has two heads, the same force needs to be calculated twice (by two threads), which reduces performance. The colors of the rectangles indicate which thread owns the force table(s). **A**: Single force table, forces that cross thread boundaries need to be calculated twice. **B**: Two force tables with atomic subtractions of forces that cross thread boundaries in table 2. **C**: Two force tables using transactional memory, the first table is only needed if a transaction fails. **D**: One separate force table per thread, which need to be summed in the end.

Method C (Figure 7C) also uses two force tables and employs the new transactional memory extension TSX, introduced by Intel with the Haswell CPU architecure. TSX provides an instruction 'xbegin', which starts a memory transaction. Right afterwards, the program is allowed to behave as if no other threads were present, adding and subtracting forces in table 2 at will. When done, the program issues the 'xend' instruction, which tells the CPU to commit the transaction. If another thread just by chance tries to access the same force vectors at the same time, the transaction fails, the CPU restores the state before the transaction and executes a fallback path instead, which is simply method B. The fraction of aborted transactions depends on the number of atoms and the number of threads working on the atoms. When simulating DHFR with eight threads, 5% of the transactions fail. With increasing system size, the failure rate approaches 2%. Since TSX has a considerable overhead, it is not trivial to outperform method B. Intel recommends that a transaction should last about 400 nanoseconds, we got good results by bundling 8 to 16 force additions (and the corresponding subtractions) in a

single transaction. This yields an improvement of 5% (measured on a Core i7 4770, since Intel unfortunately disabled TSX in the Core i7 5960X due to a hardware issue).

Method D (Figure 7D) uses one force table per thread. Each thread can thus add and subtract forces without any danger of collisions. The drawback is that this approach requires more memory to store forces, and the forces for each atom have to be summed in the end. The good news is that – if the system is large compared to the number of threads - a certain atom is usually only touched by 2-4 threads, so that only 2-4 forces per atom need to be stored and summed. Method D performs best, improving simulation times by 19% to 142.7 ns/day.

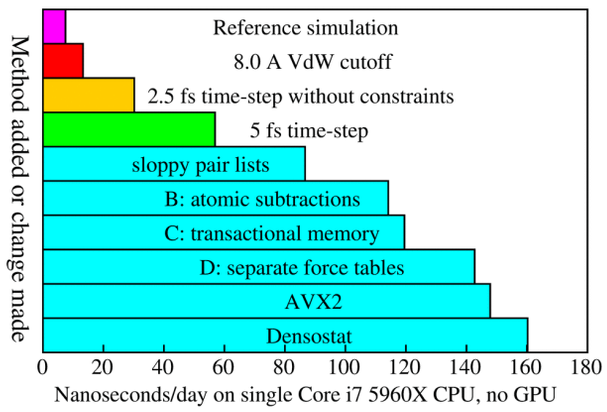



**Figure 8:** Performance of 10 simulation protocols in the DHFR benchmark. The reference protocol (top) uses a 10.5 Å cutoff, a 1 fs time-step, no constraints, no pair list, a barostat and a single force table shared among threads (Figure 7A).

Methods A and D have the inherent advantage of being reproducible, so that one can obtain the very same MD trajectory twice in a row. Methods B and C add forces in random order, and since A+B+C does not exactly equal A+C+B in floating point operations, they yield marginally different forces, which quickly causes trajectories to diverge[33].

Only time will tell which of the four methods wins on future CPU generations. Four-socket systems with Haswell Xeon CPUs may support up to 144 threads, requiring method D to store and sum so many forces per atom, that method C could run faster. Intel's new Xeon Phi 'Knights Landing' should arrive in 2015 with 288 threads and no TSX, which could make method A or B win.

The AVX2 instruction set released by Intel in 2013 helps a bit (4% to 147.8 ns/day), thanks to the doubled register space for integers, which is needed when calculating force-table lookup indices. The densostat finally adds 8% to 160.2 ns/day. The performance of this protocol for a wide range of system sizes is shown in Figure 9.

The benchmark results shown in Figures 8 and 9 have been obtained with the 32bit version of YASARA, because the 64bit version was not completely finished at the time of writing. Since 64bit operating systems happily run 32bit software, this is not an issue in practice. 64bit mode offers twice as many CPU registers, which could boost performance beyond 200 ns/day (estimated from the observation that the GROMACS 64bit version runs about 30% faster than the 32bit version).

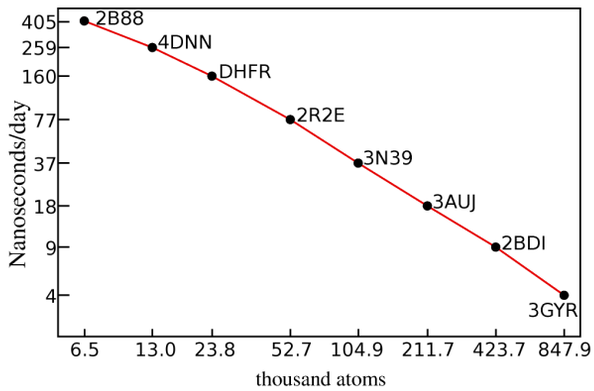



**Figure 9**: Molecular dynamics performance on a Core i7 5960X CPU for eight proteins, their PDB IDs are shown. The simulation protocol is the last (fastest) from Figure 8. Both axes use logarithmic scaling, the system size approximately doubles each step. While the structures in the middle show a linear O(N) scaling, the small structures 2B88 and 4DNN do not perform ideally, because the cost to launch and synchronize 16 threads starts to dominate the computation. For very large structures like 3GYR, the performance is reduced because the fast Fourier transform of the PME calculation has only O(N*log N) scaling and becomes the bottleneck.

*Acknowledgments:* The simulation algorithms are used for protein refinement tasks in the NewProt protein modeling project (www.newprot.eu) that is funded by the European Commission within its FP7 Programme, under the thematic area KBBE-2011-5 with contract number 289350. E.Krieger is the owner of the company that distributes YASARA, and would like to thank the YASARA users for providing invaluable feedback and financial support, Prof. Herman Berendsen for joining the PhD committee at the CMBI during the early YASARA development phase, and Dr. Gavin Seddon for his kind hardware donation.

# Materials & Methods

## Choice of programming language

To exploit the full potential of today's CPUs, one needs to make extensive use of the various vector instruction sets (e.g. SSE, AVX), where a single instruction operates on multiple data elements in parallel (SIMD approach). While compilers could in theory do that automatically, it does not work well enough in practice. Instead, the developer must write code that explicitly uses these instruction sets, either by programming directly in assembly language, or by using 'intrinsics', small C/C++ functions that operate on vector data types and map almost directly to the corresponding assembly instructions, so that the compiler has an easy job. Both approaches have disadvantages: assembly language is hard to maintain (especially with respect to local variables and register spilling), while intrinsics are rather cryptic and require to disassemble the code to check that the compiler really did what it was supposed to (which is very difficult for large functions), and both suffer from the major problem that one needs to rewrite or at least adapt the code for

each SIMD instruction set (and almost every new CPU comes with additional SIMD instructions). For a general molecular modeling application like YASARA, which uses high-performance code throughout (including molecular graphics), both approaches are impractical.

Our solution to this problem was to develop a 'meta assembly language' named PVL (portable vector language), which supports all the low-level performance tricks possible in assembly, but keeps the administration of the code nevertheless simple. Since PVL is not publicly available, we briefly describe the main features to help reproducing the results: PVL hides the complexity of the various SIMD instruction sets by providing its own simple vector data types and instructions. As a result, one needs to write the code only once, and PVL translates it to currently 16 different versions (SSE, SSE2, SSE3, SSSE3, SSE4, AVX, AVX+FMA3, AVX2, each for 32 and 64bit mode. Support for various 3DNow! combinations, e.g. SSE2+3DNow!, was dropped only recently, when AMD discontinued the latter). The different code paths can be packed into the same executable and chosen at run-time, so that a single executable runs optimally on all CPU architectures (intrinsics normally require to provide a separate executable for each CPU type). PVL can create multiple similar functions from a single parent (e.g. to calculate non-bonded forces with and without the virial, with PME or with a switching function etc.). PVL allows to define vectors with variable length to optimally fill the available registers (AVX registers can store 8 floats, while SSE registers can only store 4, and 64bit mode has twice as many registers as 32bit mode). For example, non-bonded forces are calculated for 8 atoms in parallel in 32bit SSE code, for 16 atoms in 32bit AVX code, and for 32 atoms in 64bit AVX code. PVL takes care of local variables and function parameters, addressing them via the stack pointer, so that no frame pointer is needed (reducing register shortage in 32bit mode). PVL supports position independent code (needed on Android), automatic register spilling, loop unrolling and nested functions: when using SIMD instructions, loop unrolling is not optional but obligatory (because it's very slow to access the i-th element of a vector register if 'i' is a variable). This quickly blows up the code beyond the instruction cache size (e.g. for a non-bonded force kernel, PVL creates 50000 lines of assembly code, taking ~115kb memory). Consequently, inlining functions is hopeless, and one needs a way to quickly call functions without overhead. PVL allows to embed a function inside another function, so that the callee can access all the local variables and function parameters of the caller without having to pass them explicitly.

## Choice of data structure layout

Programming for SIMD architectures involves the difficult choice between two competing approaches to arrange the data in memory: Structures of arrays and arrays of structures. The first places all data of the same type next to each other (e.g. all atom X-coordinates), so that loading a SIMD register from memory fills it with data of the same type (e.g. one AVX register with the X-coordinates of 8 atoms, one register with the Y-coordinates etc.). Operating on these data is then trivial, e.g. one would perform three multiplications and two additions to calculate 8 dot products in parallel. The second approach places all data belonging together next to each other (e.g. an atom's X, Y, Z coordinates and charge), so that loading a SIMD register from memory fills it with the data of one or more atoms (e.g. one AVX register with the positions and charges of two atoms). The second approach is far less convenient, because it requires cumbersome shuffling and 'horizontal operations' (like adding two neighbor values in the same SIMD register), e.g. one would perform one multiplication and two horizontal additions to calculate just two dot products in parallel. Nevertheless, we chose this approach in our MD algorithms, for three good reasons: first, it improves memory locality and thus cache hit rate (the position of an atom can be loaded with a single instruction from the same cache line and does not have to be gathered from three different locations). Second, current SIMD instruction sets provide good support for the difficult horizontal operations within a register, e.g. the AVX 'vdpps' instruction calculates two dot products in one shot. And third, it requires far fewer registers, which avoids expensive register spilling to memory. For example, to store the position of a water molecule, the 'structures of arrays' approach needs 9 AVX registers (to store the X/Y/Z coordinates of one O and two Hs), while the 'array of structures' approach only needs 1.5 (half a register for the oxygen position, and one register for the two hydrogen positions). Since the CPU only has 8 registers in 32bit and 16 registers in 64bit mode, this helps a lot.

## Force interpolation

Force calculation in accurate and fast MD simulations involves interpolation from lookup tables, because the treatment of longrange electrostatics with PME requires to evaluate the Gauss error function to determine the real space damping factor[25], which cannot be done fast enough. Since lookup tables 'pollute' the cache and induce slow main memory accesses, our implementation uses only four: one table with the general PME damping factors as a function of distance, and three tables with O-O, O-H and H-H forces between water molecules. Lennard Jones forces involving solute atoms are thus calculated explicitly. We use linear interpolation, which has become common practice in many popular MD programs thanks to GPUs and their built-in linear interpolation hardware (normally used for texture mapping). As described in detail previously[4], the linear interpolation error is about $1e^{-6}$. This matches the difference one gets when summing up the forces acting on an atom in a different order using single precision floats (which only have ~7 significant digits). The interpolation is performed using the AVX2 'vcvtps2dq' instruction to convert the distance to an integer index for the look-up table, two 'vgatherdps' instructions to fetch the two boundary values, 'vpand' and 'vcvtdq2ps' to calculate the two scaling factors, and one multiplication combined

with a fused multiply-add to calculate the result. The drawback of 'vgatherdps' is that it blocks three of the eight AVX registers and runs only marginally faster than manual gathering

## Algorithm used to constrain distances and angles

All bonds and selected angles involving hydrogens are constrained with a tuned variant of the LINCS method. 'Constraining the bond angle A-B-C' means that the distance between atoms A and C is constrained to sqrt(sqr(AB)+sqr(BC)-2*AB*BC*cos(ABC)), where AB, BC and ABC are the equilibrium distances and angle assigned by the force field. The tuning involves optimization for single precision calculations (next paragraph) and the handling of heavy atoms with three bound hydrogens (e.g. the $CH_3$ groups in Figure 2). In this case the six constraints (three bonds and three angles) form a tetrahedron, and the largest absolute eigenvalue of the simplified coupling matrix $\mathbf{A}$ is 1.35, so that the approximate LINCS matrix inversion $(\mathbf{1}-\mathbf{A})^{-1} = \mathbf{1}+\mathbf{A}+\mathbf{A}^2+\mathbf{A}^3+..$ fails. We therefore invert the 6x6 matrix $\mathbf{1}-\mathbf{A}$ exactly, noting that the same inverse can be used in both LINCS steps (the initial projection and the correction for rotational lengthening). We do not take advantage of the fact that $\mathbf{1}-\mathbf{A}$ is symmetric, contains a few zeroes and only ones along the diagonal, but instead simply use the fastest of Intel's SSE-optimized 6x6 matrix inversion routines (document AP-929, order number 245044-001).

To apply the constraints with sufficient accuracy (i.e. yielding a sufficiently small energy drift), we use a LINCS matrix expansion order of 8 and perform the correction for rotational lengthening three times in a row. The LINCS algorithm originally described[15] takes as input the old and new atom coordinates (obtained from the MD integrator) and then iteratively adjusts the new coordinates until the constraints are satisfied. Unfortunately single precision floating point numbers are a troublesome but unavoidable way of storing absolute atom coordinates, which get less accurate the further they are away from the origin. Every coordinate change is thus coupled with a loss of accuracy and should be avoided, which makes the many LINCS iterations required to handle angle constraints problematic. Relative coordinates on the other hand make optimal use of the 32 bits available. We therefore changed the MD integrator to provide LINCS with the old positions and the steps to the new positions instead. The steps are then adjusted by LINCS, and only added to the old positions in the end, yielding more accurate results and smaller drifts.

### Algorithm used to select constrained angles

The angles to constrain must be chosen carefully so that the eigenvalues of the simplified constraint coupling matrix stay below 1. We use a recursive algorithm, which is centered on the function FixHydrogenAngles, whose pseudo-code is given below (Figure 10).

The function FixHydrogenAngles is called first for all atoms with three bound hydrogens (-CH3, $-NH3^+$), second for all atoms with one hydrogen and two bonds (-OH, -SH), third for all atoms with two hydrogens that have at least one atom with a single and at most one atom with two hydrogens bound (this traverses along -CH2- chains, leaving maximum options for the atoms with a single hydrogen), fourth for all atoms with two hydrogens that have at most one atom with two hydrogens bound (this traverses along the remaining -CH2- chains), fifth for all atoms with a single hydrogen that have at most one atom with a single hydrogen bound (this traverses along -XH- chains in rings), and finally for all remaining atoms.

The above heuristic recipe was tuned by analyzing a large number of organic molecules, and was the easiest approach that gave optimum results (i.e. the largest number of constraints below the eigenvalue limit 1) without resorting to a global optimizer, which would have raised the complexity of the approach.

## Alanine dipeptide simulations

The alanine dipeptide was built with YASARA[21], adding acetyl- and N-methyl capping groups. The system consisted of ~3000 atoms (32 peptide atoms, 981 water molecules and three ion pairs, i.e. 0.98% NaCl). The force field was AMBER03, simulations were run at 298K with the protocols described in the caption of Figure 6. After an equilibration period of 1 ns, the current φ/ψ dihedrals were calculated every 50 fs and mapped to a 2D grid with a resolution of 5 degrees (72*72 bins), then the corresponding counter was incremented. After a microsecond, the probability in each grid bin was obtained by dividing with the total number of counts, converted to a free energy using the well known Boltzmann formula Energy = -BoltzmannConstant*298*ln(Probability), shifted so that the energy minimum was at 0, and visualized using the marching squares algorithm for seven contour levels with a spacing of 4 kJ/mol. The YASARA macro used to perform these tasks can be found in the documentation of the free YASARA View program version 15 or later, at Commands > Options > Tables > Tabulate.

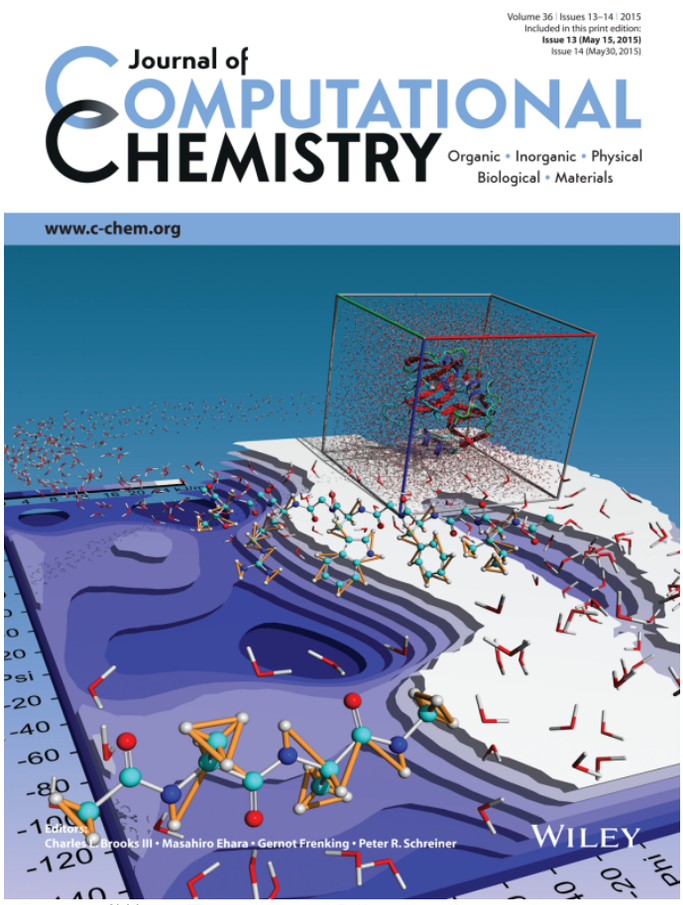
## DHFR benchmark details

All dihydrofolate reductase benchmark results were obtained by compiling and running on an Intel Core i7 5960X CPU with 3.6 GHz, the latest RedHat Enterprise Linux 7 (free CentOS version) and GCC 4.8. Turbo boost (Intel's dynamic overclocking feature) was disabled to ensure consistent timings. Hyperthreading was enabled, so that 16 threads were available. PME electrostatics were calculated with a grid spacing < 1 Å and 4$^{th}$ order B-splines. Pressure coupling was done as indicated, either based on the density (see results section), or on the pressure calculated from the virial. In the latter case, the Berendsen barostat[29] was fed with the time average pressure to avoid the suppression of fluctuations, which have been analyzed in detail for the weak coupling methods[30].

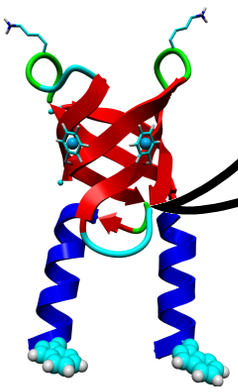
```
FixHydrogenAngles(done[],atm,lastatm)
{ // 'done' is a table that flags atoms which have already been analyzed.
  // The analysis starts at 'atm' and should not recurse to 'lastatm'.

  // First make sure that each atom is analyzed only once:
  if (done[atm]) return
  done[atm]=1

  // Maybe we will recurse to 'nextatm', but not yet
  nextatm=NONE

  // Store the bound hydrogens in 'hydtab' and their number in 'hydrogens'
  GetBoundHydrogens(hydtab,&hydrogens,atm)

  if (hydrogens==4)
  { // Handle four hydrogens like methane with two constraints
    FixAngle(hydtab[0],atm,hydtab[1])
    FixAngle(hydtab[2],atm,hydtab[3]) }

  if (hydrogens==3)
  { // Handle three hydrogens like CH3,NH3 groups with three constraints
    FixAngle(hydtab[0],atm,hydtab[1])
    FixAngle(hydtab[1],atm,hydtab[2])
    FixAngle(hydtab[2],atm,hydtab[0])
    // Don't recurse if the heavy atom bound to atm is sp3 with <=1 hydrogens
    nextatm=BoundHeavyAtom(atm)
    if (nextatm!=NONE and BoundHydrogens(nextatm)<=1 and Bonds(nextatm)>3) nextatm=NONE }

  if (hydrogens==2)
  { if (Bonds(atm)>3)
    { // Two hydrogens bound to sp3 atom. These need constraints to the next heavy atom:
      nextatm = A heavy atom bound to atm, which is not lastatm, which has <3 hydrogens,
                which does not already have an angle constraint atm-nextatm-x,
                and which has the highest score. The score is 3 if nextatm has two bound
                hydrogens (it's best if we continue along a -CH2- chain), 2 if it has 0
                hydrogens (but also OK to end at an atom without hydrogens), 1 if it has
                1 hydrogen and 3 bonds, otherwise 0. }
    if (nextatm!=NONE)
    { // Found a bound heavy atom that can take the constraints
      FixAngle(hydtab[0],atm,nextatm)
      FixAngle(hydtab[1],atm,nextatm)
      // Again, don't recurse if nextatm is sp3 with <=1 hydrogens, these are handled later
      if (BoundHydrogens(nextatm)<=1 and Bonds(nextatm)>3) nextatm=NONE }
    else
    { // Two hydrogens bound to sp2 atom (or no nextatm found), these can safely be coupled
      FixAngle(hydtab[0],atm,hydtab[1]) } }

  if (hydrogens==1)
  { // A single hydrogen gets a single constraint. Find best partner atom.
    nextatm = A heavy atom bound to atm, which has <3 hydrogens and the best score.
              The score is -(number of bound hydrogens + number of angle constraints
              nextatm-x-y)*4. If nextatm==lastatm or there already is an angle constraint
              atm-nextatm-x, then score=score-20. If nextatm is not in the same residue
              as atm, then score=score-10.
    if (nextatm!=NONE) FixAngle(hydtab[0],atm,nextatm) }
  // Recurse if it makes sense
  if (nextatm!=NONE) FixHydAngles(done,nextatm,atm) }
```

**Figure 10:** Pseudo code of the aligorithm used to restrain hydrogen bond angles

1. Duan, Y. et al. A Point-Charge Force Field for Molecular Mechanics Simulations of Proteins. *J.Comp.Chem.* **24**, 1999-2012 (2003).
2. Brooks, B. R. et al. CHARMM: the biomolecular simulation program. *J.Comp.Chem.* **30**, 1545-1614 (2009).
3. Kaminski, G., Friesner, R. A., Tirado-Rives, J. & Jorgensen, W. L. Evaluation and Reparametrization of the OPLS-AA Force Field for Proteins via Comparison with Accurate Quantum Chemical Calculations on Peptides. *J.Phys.Chem.B* **105**, 6474-6487 (2001).
4. Harvey, M. J., Giupponi, G. & De Fabritiis, G. ACEMD: Accelerated molecular dynamics simulations in the microsecond timescale. *J.Chem.Theory Comput.* **5**, 1632-1639 (2009).
5. Eastman, P. et al. OpenMM 4: A Reusable, Extensible, Hardware Independent Library for High Performance Molecular Simulation. *J.Chem.Theory Comput.* **9**, 461-469 (2013).
6. Krieger, E. et al. Improving physical realism, stereochemistry and side-chain accuracy in homology modeling: four approaches that performed well in CASP8. *Proteins* **77, Suppl. 9**, 114-122 (2009).
7. Brunger, A. *X-PLOR version 3.1: A system for X-ray crystallography and NMR* (Yale University Press,New Haven,CT, 1992).
8. Kuszewski, J., Gronenborn, A. M. & Clore, G. M. Improvements and extensions in the conformational database potential for the refinement of NMR and X-ray structures of proteins and nucleic acids. *J Magn Reson* **125**, 171-177 (1997).
9. Krieger, E., Darden, T., Nabuurs, S. B., Finkelstein, A. & Vriend, G. Making optimal use of empirical energy functions: force field parameterization in crystal space. *Proteins* **57**, 678-683 (2004).
10. Lindorff-Larsen, K. et al. Systematic validation of protein force fields against experimental data. *PLoS One* **7**, e32131 (2012).
11. Feenstra, K. A., Hess, B. & Berendsen, H. J. Improving efficiency of large time-scale molecular dynamics simulations of hydrogen-rich systems. *J.Comp.Chem.* **20**, 786-798 (1999).
12. Streett, W. B. & Tildesley, D. J. Multiple time-step methods in molecular dynamics. *Mol.Phys.* **35**, 639-648 (1978).
13. Izaguirre, J. A. et al. in *Lecture Notes in Computational Science and Engineering* 146-174 (2002).
14. Phillips, J. C. et al. Scalable molecular dynamics with NAMD. *J.Comp.Chem.* **26**, 1781-1802 (2005).
15. Hess, B., Bekker, H., Berendsen, H. J. C. & Fraaije, J. G. E. M. LINCS: a linear constraint solver for molecular simulations. *J.Comp.Chem.* **18**, 1463-1472 (1997).
16. Ryckaert, J. P., Ciccotti, G. & Berendsen, H. J. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *J.Comp.Phys.* **23**, 327-341 (1977).
17. Grubmueller, H. & Tavan, P. Multiple time step algorithms for molecular dynamics simulations of proteins: how good are they? *J.Comp.Chem.* **19**, 1534-1552 (1998).
18. Grubmueller, H., Heller, H., Windemuth, A. & Schulten, K. Generalized Verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Mol.Simul.* **6**, 121-142 (1991).
19. Shuichi, M. & Kollman, P. A. SETTLE: an analytical version of the SHAKE and RATTLE algorithm for rigid water models. *J.Comp.Chem.* **13**, 952-962 (1992).
20. Hess, B. P-LINCS: A parallel linear constraint solver for molecular simulation. *J.Chem.Theory Comput.* **4**, 116-122 (2008).
21. Krieger, E. & Vriend, G. YASARA View - molecular graphics for all devices - from smartphones to workstations. *Bioinformatics* **30**, 2981-2982 (2014).
22. Hess, B., Kutzner, C., van der Spoel, D. & Lindahl, E. GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. *J.Chem.Theory Comput.* **4**, 435-447 (2008).
23. Wang, J., Wolf, R. M., Caldwell, J. W., Kollman, P. A. & Case, D. A. Development and testing of a general AMBER force field. *J.Comp.Chem.* **25**, 1157-1174 (2004).
24. Piana, S. et al. Evaluating the effects of cutoffs and treatment of long-range electrostatics in protein folding simulations. *PLoS One* **7**, e39918 (2012).
25. Essman, U. et al. A smooth particle mesh Ewald method. *J.Chem.Phys.* **103**, 8577-8593 (1995).
26. Piana, S., Lindorff-Larsen, K. & Shaw, D. E. How robust are protein folding simulations with respect to force field parameterization. *Biophys. J.* **100**, L47-L49 (2011).
27. Price, D. J. & Brooks, C. L. r. A modified TIP3P water potential for simulation with Ewald summation. *J.Chem.Phys.* **121**, 10096-10103 (2004).
28. Lague, P., Pastor, R. W. & Brooks, B. R. Pressure-based long-range correction for Lennard-Jones interactions in molecular dynamics simulations: application to alkanes and interfaces. *J.Phys.Chem.B* **108**, 363-368 (2004).
29. Berendsen, H. J. C., Postma, J. P. M., van Gunsteren, W. F., DiNola, A. & Haak, J. R. Molecular dynamics with coupling to an external bath. *J.Chem.Phys.* **81**, 3684-3690 (1984).
30. Morishita, T. Fluctuation formulas in molecular dynamics simulations with the weak coupling heat bath. *J.Chem.Phys.* **113**, 2976-2982 (2000).
31. Wang, J., Cieplak, P. & Kollman, P. A. How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules? *J.Comp.Chem.* **21**, 1049-1074 (2000).
32. Anderson, A. G. & Hermans, J. Microfolding: conformational probability map for the alanine dipeptide in water from molecular dynamics simulations. *Proteins* **3**, 262-265 (1988).
33. Braxenthaler, M., Unger, R., Auerbach, D., Given, J. A. & Moult, J. Chaos in protein dynamics. *Proteins* **29**, 417-425 (1997).


**Space filler:** The cover of J.Comput.Chem. shows various aspects of this article. The validation of the protocol includes simulations of the enzyme dihydrofolate reductase (shown at the back) and a capped alanine dipeptide (shown at the front). The phi/psi dihedral angle distribution extracted from the latter simulation is depicted as a free energy landscape in shades of blue. One of the acceleration methods involves the LINCS algorithm, tuned to constrain not only bonds but also selected angles formed by hydrogen atoms. These constraints are shown in bright orange for the alanine dipeptide, and for a larger peptide with sequence AIAFAWARATA in the middle.



The main chapter 3 in this book describes how force field parameters can be optimized while minimizing high-resolution crystal structures. The force field learns how to recognize nature's real energy minimum and keep the protein close to it. Obviously the optimization target structures should contain the same atoms that were present during the X-ray diffraction experiment. Unfortunately PDB files are missing many atoms that are not visible in the electron density, most of which are hydrogen atoms. This chapter deals with the prediction of missing hydrogens at ionizable groups (Glu, Asp, His, Tyr, Lys) considering the pH at which the crystal was solved as well as the periodicity of the crystal cell.



# Fast empirical pKa prediction by Ewald summation

Elmar Krieger*, Jens E. Nielsen#, Chris A.E.M. Spronk* and Gert Vriend*

*J.Mol.Graph.Model.* **25**, 481-486 (2006)

** Center for Molecular and Biomolecular Informatics, University of Nijmegen, Toernooiveld 1, 6525ED Nijmegen, Netherlands*
*# Department of Biochemistry, Conway Institute, University College Dublin, Dublin 4, Ireland*

## Abstract

pKa calculations for macromolecules are normally performed by solving the Poisson-Boltzmann equation, accounting for the different dielectric constants of solvent and solute, as well as the ionic strength. Despite the large number of successful applications, there are some situations where the current algorithms are not suitable: 1) Large scale, high-throughput analysis which requires calculations to be completed within a fraction of a second, e.g. when permanently monitoring pKa shifts during a molecular dynamics simulation. 2) Prediction of pKas in periodic boundaries, e.g. when reconstructing entire protein crystal unit cells from PDB files, including the correct protonation patterns at experimental pH. Such *in silico* crystals are needed by 'self-parameterizing' molecular dynamics force fields like YAMBER, that optimize their parameters while energy-minimizing high-resolution protein crystals.

To address both problems, we define an empirical equation that expresses the pKa as a function of electrostatic potential, hydrogen bonds and accessible surface area. The electrostatic potential is evaluated by Ewald summation, which captures periodic crystal environments and the uncertainty in atom positions using Gaussian charge densities. The empirical proportionality constants are derived from 217 experimentally determined pKas, and despite its simplicity, this pKa calculation method reaches a high overall jack-knifed accuracy, and is fast enough to be used during a molecular dynamics simulation. A reliable null-model to judge pKa prediction accuracies is also presented.

## Introduction

The prediction of pKa values in proteins has made considerable progress over the last years[1,2]. The Poisson-Boltzmann equation (PBE) has become an important tool because it allows the calculation of the electrostatic potential in a heterogeneous solute-solvent system, taking into account dielectric boundaries and the ionic strength. Initial approaches to electrostatic calculations were based on rough approximations like spherical proteins[3]. The ability to solve the Poisson-Boltzmann equation for arbitrarily shaped proteins[4-6] cleared the path for a range of successful applications, such as studies of enzymatic activity[7], pH-dependent conformational changes[8] and protein stability[9-11]. These algorithms, however, are computationally expensive, and consequently led to the development of several simplified algorithms that avoid solving the PBE. Examples of these algorithms are the Debye-Hueckel approach[12] and the electrostatic screening functions[13,14].

pKa calculations have always focused on proteins in their physiological environment, matching the experimental determination of pKa values, which is also done in solution using NMR spectroscopy. However, the quality of pKa calculations depends heavily on the availability of high resolution protein structures. NMR structures of sufficient resolution are often not available, and one is forced to predict solution pKa values using X-ray structures. Much effort has been devoted to determining the regions of structural divergence, excluding residues involved in crystal contacts[15], optimizing X-ray structures[16] and incorporating information on protein flexibility[17].

## The goal is pKa prediction in protein crystals

The approach presented here has been developed due to a lack of solutions for a problem that appears paradoxical, given the facts mentioned above: the prediction of pKa values in protein crystals. Because of the crystal packing interactions, these pKas certainly differ from those measured in solution. The reason for addressing this problem becomes clear in view of recent developments in force field research. Thanks to the virtually unlimited resources provided by distributed computing systems like Models@Home[18], it became feasible to use complete proteins in-

stead of small molecules as optimization targets when fitting the force field parameters[19]. This was done by randomly changing force field parameters and running simulations on a series of protein structures to see if the parameter changes would be beneficial. Obviously, the protein structures in the optimization set should be as realistic as possible, otherwise the force field might memorize features that are just structural artifacts. This can be achieved by taking high resolution X-ray structures and reconstructing the entire unit cell, including water molecules, counter ions and all solute hydrogens. The correct placement of polar hydrogens is especially important, and in addition to optimizing the hydrogen-bond network[20], this requires the pKa values of all ionizable residues in the protein crystal and the pH at which the protein was crystallized. The force field parameters are then optimized in crystal space, so that all the interactions responsible for the experimentally observed structure can be considered, while converging at a force field like YAMBER[21]. Because crystal and solution environments obey the same laws of physics, the optimized force field can be used in both.

## Ewald summation captures the periodic environment

Electrostatic calculations in periodic crystal systems are complicated by the infinite number of interactions. A clever way of making the problem tractable is Ewald summation[22], which allows the calculation of the potential due to the N particles in the unit cell and an infinite number of periodic replicas. The method combines a rapidly converging short-range term with a long-range component evaluated in reciprocal space[23]. If the reciprocal sum is calculated using a particle-mesh approximation, the resulting Particle Mesh Ewald (PME) algorithm[24] is considerably faster than the standard Ewald method. PME is part of almost every molecular dynamics program, and forms the basis for this work. However, we only use the reciprocal space portion, which provides the solution to Poisson's equation with periodic boundaries, Gaussian charge distributions and a single dielectric constant. By ignoring the short-range term and the associated damping of the reciprocal space term at short-range, we essentially remove the long-range attribute from the reciprocal space term: it now covers all distance ranges equally, and differs from Coulomb's law only by the use of Gaussian charge densities instead of localized point charges. Smeared-out Gaussians account for the uncertainty in atom positions (which also proved beneficial for the development of knowledge-based potentials[25]). Compared to the Poisson-Boltzmann equation, this approach however lacks two advantages: implicit counter ions and different dielectric constants for solvent and solute.

In an extensive optimization study, Demchuck & Wade[1] determined that the best dielectric constant for solvent exposed residues is close to the one of water (80), while the protein interior should be assigned a value in the range of 10 to 20. Since 20 differs from 80 only by a factor of 4, we hypothesized that a single global dielectric constant could suffice for accurate predictions, provided that some additional structural information was incorporated to account for the simplification.

## The pKa can be approximated as a function of electrostatic potential, hydrogen bonds and accessible surface

Using simplified physical considerations and some modeler's experience, we defined three rules of thumb for pKa prediction. The first and partly the second rule have also been mentioned in a recent analysis of carboxyl pKa values[26]:

(1) If an ionizable group is surrounded by negatively charged residues, corresponding to a negative electrostatic potential, protonation becomes easier, the pKa increases. Similarly, if there are positively charged residues around, the pKa decreases. As a first approximation, the pKa shift is thus assumed to be proportional to the electrostatic potential.

(2) If an ionizable group accepts hydrogen bonds, the space to place a proton is reduced, protonation becomes harder, and the pKa decreases. If after protonation, the group can donate a bond, protonation is favorable, the pKa increases.

(3) If a group accepts hydrogen bonds and is buried, the pKa is decreased even further, because the side-chain cannot facilitate protonation by moving to a different conformation where it does not receive hydrogen bonds. If a buried group can donate a hydrogen bond after protonation, the pKa increases, because there is no space for water molecules that could ease the energetic cost of two hydrogen bond acceptors facing each other.

These three assumptions were fused into an empirical equation relating the pKa of a residue with the electrostatic potential, the number of hydrogen bonds and the accessible surface area:

$$pKa = Model\ pKa + Sign(HBSum) * C * SurfaceLoss + \sum_{Ionizable\ atoms} \left[ -A * EwaldE_i + B * HB_i \right] \quad (1)$$

In this equation, *Model pKa* is the standard pKa value of a certain residue type, $EwaldE_i$ is the reciprocal space portion of the Ewald energy of a charge +1 at the location of the $i^{th}$ ionizable atom in the residue (in kcal/mol), $HB_i$ is the difference between (potentially) donated and accepted hydrogen bonds at the $i^{th}$ atom, *HBSum* is the sum over all $HB_i$, and *SurfaceLoss* is the loss of accessible surface area of the side-chain with respect to a fully exposed state. *A, B* and *C* are empirical proportionality constants. The four unknown parameters *Model pKa*, *A, B* and *C* are globally optimized for each amino acid type so that the RMSD between predicted and observed pKa values is minimal. More details about the equation can be found in the Materials & Methods section.

The RMSD was chosen as optimization target because it is ideally suited for analyzing pKa prediction accuracy:

one is not so much interested in the small shifts of isolated surface residues, but in the large shifts that significantly influence the protonation states and dominate the RMSD when mispredicted. As the main goal of this work is to develop a method for pKa prediction in protein crystals used in force field parameterization, all residues are equally important. No matter if a wrong protonation state is assigned to an active site residue or to a surface residue involved in crystal contacts – the influence on the optimized force field is equally bad. Hence the RMSD is calculated for all residues with experimentally determined pKa values, and our goal is to obtain a low overall RMSD.

When fitting the parameters to reproduce experimental pKa values, it is important to note that these pKa values were measured in solution. The Ewald energy in equation 1 must therefore also be calculated in a solution environment. This is achieved by placing an isolated protein in a very large cell, so that the periodic boundaries implied by the Ewald summation have no significant influence. Validation of prediction accuracies is thus also only possible in solution, since there is currently no experimental method to directly measure pKas in protein crystals. The resulting parameters can however be used directly for pKa prediction in protein crystals, just like the same molecular dynamics force field parameters can be used to simulate proteins in solution and crystals.

# Materials & Methods

## Datasets of experimental pKa values

A total of 227 experimentally measured pKa values were compiled for this study. They consisted of the Asp/Glu specific dataset collected by Forsyth et al.[26], from which we removed double occurrences of the same protein to prevent compromising the jack-knife test, and four cases where it was uncertain if the structure deposited in the PDB was close to the one used for the pKa measurements: CD2 because it undergoes domain swapping (see PDB IDs 1CDC and 1HNG), chymotrypsin inhibitor 2, because 19 important N-terminal residues were disordered in the structure, subunit C of F0F1-ATPase because its structure was determined in a chloroform-methanol mixture and HIV-1 protease/KNI-272 complex because there were no force field parameters available for the essential ligand. Then the histidine specific dataset from Edgcomb&Murphy[27] and our own previously described collection[15] were included, which added mainly pKa values for lysines and tyrosines. Too few pKa values of carboxyl-termini were available to be included in this analysis, which lead to slightly different RMSDs in Table 3 compared to our previous work[15].

## Hydrogen bond counting

The numbers of accepted and donated hydrogen bonds contributing to *HBSum* in equation 1 were determined after an optimization of the hydrogen-bond network with WHAT IF[20], which was previously shown to significantly improve pKa prediction accuracy[15]. A hydrogen bond was allowed to contribute to *HBSum* if the distance between the hydrogen and the acceptor was below 2.5 Å and donor and acceptor were separated by more than three covalent bonds. For carboxyl oxygens and histidine nitrogens that were not protonated in the optimized network, a (potentially) donated bond was counted if there was an acceptor separated by more than three bonds within 3.5 Å.

## Calculation of the electrostatic potential

*EwaldE* in equation 1 was calculated using the PME algorithm[24] implemented in YASARA (available from www.YASARA.org, including the pKa prediction module described here). To avoid singularities and short-range noise, all calculations were done with the reciprocal part of the Ewald sum only, with a grid spacing <1Å, 6$^{th}$ order B-splines and a tolerance of 1e-5 for the direct space sum (which was used to determine the convergence parameter for the reciprocal sum). The simulation cell was 30Å larger than the protein along each axis. Charges were assigned to all atoms based on the Amber 99 force field[28]. All ionizable groups were in their standard protonation states (i.e. D, E and H deprotonated, K and Y protonated), no iterations to sample different protonation patterns were done. For lysine and tyrosine, the potential was calculated at the protonated NZ and OH atoms, for histidine at the deprotonated NE2 atom, and for aspartate and glutamate at both deprotonated oxygens (for the latter two residues, the sum in equation 1 therefore runs over two atoms). As the histidine ring can flip, the NE2 atom was assigned based on the following rules: 1) If the histidine accepts a hydrogen bond, the acceptor is the NE2 atom. 2) If the histidine donates a bond, the other nitrogen is labeled NE2. 3) If neither the first nor the second is true, both nitrogens are temporarily protonated and the one with the higher electrostatic potential is assumed to be NE2. The resulting energies to be used in equation 1 typically fall in the range –100 to +100 kcal/mol, the common scaling factor 0.5 accounting for the bi-directionality of electrostatic interactions is omitted.

## Accessibility calculations

The loss of accessible surface area was calculated by subtracting the side-chain accessibilities calculated with WHAT IF's standard parameters[29] from the following values corresponding to a fully exposed state: Asp 34Å$^2$, Glu 40Å$^2$, Tyr 60Å$^2$, His 51Å$^2$, Lys 55Å$^2$.

## Performance details

The performance of the method was evaluated on a 3 GHz PentiumIV machine using a typical TIM barrel (PDB ID 5TIM chain A). Evaluation of the electrostatic potential with the PME algorithm[24] takes 0.25 seconds, followed by calculation of the accessible surface (0.10 seconds) and assignment of hydrogen bonds (0.004 seconds), summing up to 0.354 seconds in total. For comparison, one step of a 5TIM molecular dynamics simulation (29000 atoms including water, simulation parameters as described previously[21]) takes 0.63 seconds. Hence it costs only ~2% per-

formance to monitor all pKa values ever 25th simulation step.

The pKa predictions together with the user-specified pH can then be used to reassign the protonation states, or preferably (since the influence on the simulation is smaller) to assign new target values for the fractional protonation λ, which defines a mixture of force field parameters for the protonated and deprotonated states[30].

## Results and Discussion

To evaluate the accuracy of the Ewald summation approach, pKa predictions were made for 227 aspartate, glutamate, histidine, lysine and tyrosine residues in a set of 27 structures. For the remaining ionizable side-chain types, we did not have enough experimental data to fit the parameters in equation 1.

The RMSDs of the predicted pKa values from the experimental ones are listed in Table 1. All RMSDs, including the one for the null-model, have been obtained with a Jack-knife approach, i.e. the parameters were separately determined for each of the 27 structures using the remaining 26 structures.

One important question is whether or not a pKa prediction method performs better than the null-model, which trivially assumes a constant pKa value for all residues of a certain type, and can be surprisingly difficult to beat[15]. As can be seen from Table 1, our empirical equation gives better overall results (0.879) than the null-model (0.965). The best results are achieved for aspartate (0.71 compared to 0.95), while histidine turns out to be the most difficult residue (1.59 compared to 1.56).

It has been noted before that there is virtually no correlation between accessible surface area and pKa shift[26,27]. Indeed, the surface term of our equation makes the weakest contribution and can be left out without significantly compromising the accuracy. This is shown in the fourth column of Table 1 (overall RMSD 0.899).

To estimate the relative importance of the Ewald- and hydrogen bonding terms, we also tested a reduced model considering only the two parameters 'model pKa' and 'hydrogen bonding'. The resulting RMSD of 0.935 was roughly half-way between the null- model (0.965 in Table 1) and the three parameter model (0.899), indicating that Ewald energy and hydrogen bonding contribute about equally to the prediction.

| Residue Type | No. | Null Model | Ewald, 3 Parameters | Ewald, 4 Parameters |
|---|---|---|---|---|
| Asp | 83 | 0.948 | 0.739 | 0.710 |
| Glu | 81 | 0.717 | 0.673 | 0.678 |
| His | 35 | 1.563 | 1.636 | 1.592 |
| Tyr | 6 | 0.837 | 0.837 | 0.837 |
| Lys | 22 | 0.502 | 0.399 | 0.399 |
| All | 227 | 0.965 | 0.899 | 0.879 |

**Table 1: Prediction accuracy for 227 experimentally determined pKas.** The second column lists the number of predictions per residue type. The RMSDs obtained with the optimized null-model are shown in the third column. The fourth column lists the results obtained if only three parameters are used (parameter C in the equation 1 set to zero). Only six tyrosine residues were present, which was not enough to reliably fit more than one parameter (the model pKa). RMSDs for tyrosine are therefore identical in all cases. For the 22 lysine residues, only three parameters could be fit, giving the same results in the fourth and fifth column.

Table 2 lists three different parameter sets: The null-model alone, the parameters for equation 1 without the surface term (C set to zero), and the parameters for the complete equation 1. It can be seen that without the surface term, parameters have a clear physical meaning. E.g. accepting a hydrogen bond lowers the pKa of aspartate by 0.3 units, and the one of glutamate by 0.17 units. This matches the previous finding that pKa values of glutamates are less influenced by hydrogen bonds[26]. One simple explanation might be that the glutamate side-chain is more flexible, so rather than being protonated while accepting hydrogen bonds, it adapts a different conformation.

As soon as the surface term is included, we find param-

| Residue Type | Model-1 | | Model-3 | A | B | | Model-4 | A | B | C |
|---|---|---|---|---|---|---|---|---|---|---|
| Asp | 3.220 | | 3.280 | 0.00264 | 0.3032 | | 3.270 | 0.00254 | 0.4725 | -0.01663 |
| Glu | 4.090 | | 3.949 | 0.00209 | 0.1670 | | 3.904 | 0.00224 | 0.2883 | -0.01145 |
| His | 6.200 | | 5.942 | 0.01112 | 0.7447 | | 5.871 | 0.01002 | -1.1772 | 0.05323 |
| Tyr | 10.800 | | 10.800 | - | - | | 10.800 | - | - | - |
| Lys | 10.760 | | 10.938 | 0.00408 | 0.0924 | | 10.941 | 0.00424 | -0.0042 | 0.00479 |

**Table 2: Empirical parameters for pKa prediction.** The table shows three different sets with an increasing number of parameters. In the first case, only the model pKa is optimized, resulting in the null-model (pKa values in column 2). If additionally the electrostatic potential and the hydrogen bonds are considered, three parameters are required (columns 3 to 5). Inclusion of the surface term requires four parameters (columns 6 to 9). A, B and C are the parameters used in equation 1. The lack of data for tyrosin allowed to fit just one parameter, the model pKa.

| Residue Type | No. | Null Model | Ewald, 3 Parameters | Ewald, 4 Parameters |
|---|---|---|---|---|
| Asp | 45 | 0.847 | 0.592 | 0.573 |
| Glu | 41 | 0.802 | 0.777 | 0.788 |
| His | 8 | 1.591 | 1.318 | 1.206 |
| Tyr | 6 | 0.837 | 0.837 | 0.837 |
| Lys | 22 | 0.502 | 0.399 | 0.399 |
| All | 122 | 0.853 | 0.714 | 0.699 |

**Table 3: Comparison of pKa prediction accuracy for four different methods**: the null-model (column 3), the empirical equation described here, without (column 4) and with (column 5) the surface term, and finally the Poisson-Boltzmann equation based approach described previously[15]. Listed is the RMSD between predicted and experimentally measured pKa.



eter dependencies that make a physical interpretation hardly fruitful (e.g. the sign of one parameter changes unexpectedly, while another one compensates). This finding does not indicate that desolvation effects are unimportant, it just shows that either the surface term cannot truly capture their physical basis, or that the number of residues with significant desolvation effects is small. Nevertheless, the surface term passed the Jack-knife test, indicating that the increase of accuracy is not just due to the additional optimization parameter, but that there is indeed a small signal present.

Comparison of these results with other prediction methods is difficult, because they are very dataset dependent (e.g. the relatively high RMSD for aspartate is mainly caused by Asp-26 in Thioredoxin, with a predicted pKa of 3.5 and a measured pKa of 8.1[31]). We therefore calculated RMSDs for a subset of residues, matching the dataset used in our previous analysis based on the Poisson-Boltzmann equation[15], which compared favorably to other commonly used pKa calculation methods. Table 3 shows that the empirical approach works surprisingly well, giving lower RMSDs in all cases. The optimized null-model also performs better than expected.

While the focus in this work is on accurate pKa prediction for all residues, catalytic active site residues are often the most interesting ones. Based on the articles describing the 27 proteins used here, 20 catalytic aspartates, glutamates and histidines could be identified. Not surprisingly, prediction accuracy is worse (1.36 pKa units RMSD) but still compares well to the optimal null-model (1.51). The same holds for the comparison to the Poisson-Boltzmann method in the common subset (1.46 to 1.61).

## Conclusion

While the initial motivation for this work was the need to predict pKa values in protein crystals, we ended up with two interesting findings.

First, our empirical approach based on a global dielectric constant and hydrogen bond counting resulted in a lower RMSD than the PBE based method. Partly, this can be attributed to the parameter fitting procedure, which allows us to find optimum values for variables that are difficult to determine, both theoretically and experimentally. PBE calculations also provide room for improvement by parameter optimization. This has already been shown for the dielectric constants[1] and the hydrogen bonding network[15], while the next obvious candidates are the model pKa values. These are crucial parameters, and estimating them from compounds that only resemble amino acids carries the inherent danger of a systematic error. No physical meaning is lost if they are optimized instead. Another important reason for the lower RMSD is fewer mispredictions. Due to its quadratic nature, the RMSD is dominated by the large mispredictions. We found that the PBE method occasionally predicts large pKa shifts that are not observed experimentally. While it is sometimes suggested that pKa prediction should only look at residues that exhibit large shifts and not bother about the rest, this finding shows that all residues are important: a lot can be learned from analyzing why theory predicts a shift if none is found experimentally.

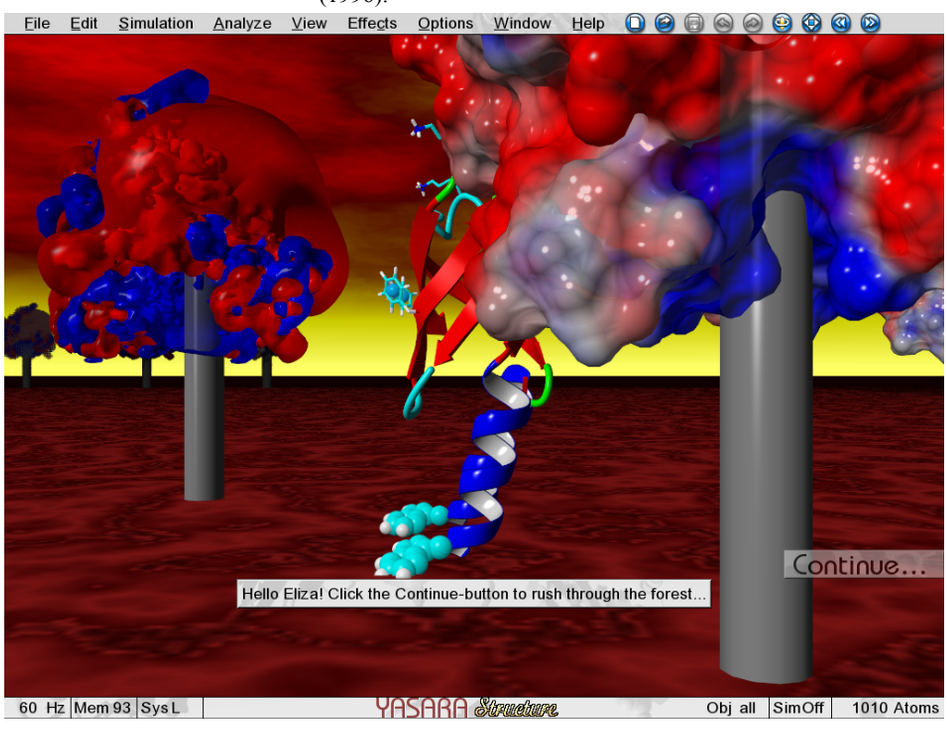
Second, the null-model is still hard to beat, as long as it is optimized and not assumed to be equal to pKa values of model compounds used in PBE calculations. E.g. the optimum null-model pKa for aspartate is 3.22 (Table 2). This differs by ~0.8 pKa units from the value used in PBE calculations[15]. When comparing prediction accuracies with the null-model, it is essential to use the optimized values, so that the null-model gets a fair chance, since results obtained with the classic null-model[15] are much worse (1.069 instead of 0.853 pKa units RMSD in Table 3).

As a conclusion, pKa prediction remains a difficult topic, hampered by uncertainties in the experimental data and the underlying protein structures. It is likely that even a perfect method would still be facing unexplainable outliers and a lot of noise, making it hard to objectively decide between competing theories. Even though our approach can in principle not reproduce titration curves of tightly coupled residues, its focus on high overall accuracy for real-life residue distributions and support for periodic boundaries and non-orthorhombic cells makes it well suited for the intended purpose: the rapid large scale prediction of pKa values and assignment of protonation states

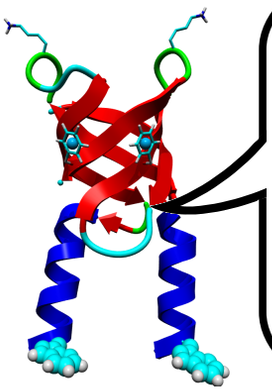
in force field parameterization, molecular dynamics simulations and homology model refinement.

*Acknowledgements:* this work was supported by the EU (5th Framework program, contract number QLG2-CT-2000-01313) and the users of the molecular modeling program YASARA.

1. Demchuk, E. & Wade, R. C. Improving the continuum dielectric approach to calculating pKas of ionizable groups in proteins. *J.Phys.Chem.* **100**, 17373-17387 (1996).
2. Warshel, A. Calculations of enzymatic reactions: calculations of pKa, proton transfer reactions, and general acid catalysis reactions in enzymes. *Biochemistry* **20**, 3177 (1981).
3. Tanford, C. & Roxby, R. Interpretation of protein titration curves. Application to lysozyme. *Biochemistry* **11**, 2193-2198 (1972).
4. Warwicker, J. & Watson, H. C. Calculation of the electric potential in the active site cleft due to alpha-helix dipoles. *J.Mol.Biol.* **157**, 671-679 (1982).
5. Yang, A. S., Gunner, M. R., Sampogna, R., Sharp, K. & Honig, B. On the calculation of pKas in proteins. *15*, 252-265 (1993).
6. Antosiewicz, J., McCammon, J. A. & Gilson, M. K. Prediction of pH-dependent properties of proteins. *J.Mol.Biol.* **238**, 415-436 (1994).
7. Raquet, X., Lounnas, V., Lamotte-Brasseur, J., Frere, J. M. & Wade, R. C. pKa calculations for class A beta-lactamases: methodological and mechanistic implications. *Biophys.J.* **73**, 2416-2426 (1997).
8. Morikis, D., Elcock, A. H., Jennings, P. A. & McCammon, J. A. Native-state conformational dynamics of GART: A regulatory pH-dependent coil-helix transition examined by electrostatic calculations. *Protein Sci.* **10**, 2363-2378 (2001).
9. Alonso, D. O., Dill, K. A. & Stigter, D. The three states of globular proteins: acid denaturation. *Biopolymers* **31**, 1631-1649 (1991).
10. Lambeir, A. M. et al. The ionization of a buried glutamic acid is thermodynamically linked to the stability of Leishmania mexicana triose phosphate isomerase. *Eur.J.Biochem.* **267**, 2516-2524 (2000).
11. Yang, A. S. & Honig, B. Structural origins of pH and ionic strength effects on protein stability. Acid denaturation of sperm whale apomyoglobin. *J.Mol.Biol.* **237**, 602-614 (1993).
12. Warwicker, J. Simplified methods for pKa and acid pH-dependent stability estimation in proteins: removing dielectric and counterion boundaries. *Protein Sci.* **8**, 418-425 (1999).
13. Sandberg, L. & Edholm, O. A fast and simple method to calculate protonation states in proteins. *Proteins* **36**, 474-483 (1999).
14. Mehler, E. L. & Guarnieri, F. A self-consistent, microenvironment modulated screened coulomb potential approximation to calculate pH-dependent electrostatic effects in proteins. *Biophys.J.* **77**, 3-22 (1999).
15. Nielsen, J. E. & Vriend, G. Optimizing the hydrogen-bond network in Poisson-Boltzmann equation-based pKa calculations. *Proteins* **43**, 403-412 (2001).
16. Nielsen, J. E. & McCammon, J. A. On the evaluation and optimization of protein X-ray structures for pKa calculations. *Protein Sci.* **12**, 313-326 (2003).
17. Georgescu, R. E., Alexov, E. G. & Gunner, M. R. Combining conformational flexibility and continuum electrostatics for calculating pK(a)s in proteins. *Biophys.J.* **83**, 1731-1748 (2002).
18. Krieger, E. & Vriend, G. Models@Home: distributed computing in bioinformatics using a screensaver based approach. *Bioinformatics* **18**, 315-318 (2002).
19. Krieger, E., Koraimann, G. & Vriend, G. Increasing the precision of comparative models with YASARA NOVA - a self-parameterizing force field. *Proteins* **47**, 393-402 (2002).
20. Hooft, R. W. W., Sander, C. & Vriend, G. Positioning hydrogen atoms by optimizing hydrogen-bond networks in protein structures. *Proteins* **26**, 363-376 (1996).
21. Krieger, E., Darden, T., Nabuurs, S. B., Finkelstein, A. & Vriend, G. Making optimal use of empirical energy functions: force field parameterization in crystal space. *Proteins* **57**, 678-683 (2004).
22. Ewald, P. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann.Phys.* **64**, 253-287 (1921).
23. Perram, J. W., Petersen, H. G. & de Leeuw, S. W. An algorithm for the simulation of condensed matter which grows as the 3/2 power of the number of particles. *Mol.Phys.* **65**, 875-889 (1988).
24. Essman, U. et al. A smooth particle mesh Ewald method. *J.Chem.Phys.* **103**, 8577-8593 (1995).
25. Vriend, G. & Sander, C. Quality control of protein models: Directional atomic contact analysis. *J.Appl.Cryst.* **26**, 47-60 (1993).
26. Forsyth, W. R., Antosiewicz, J. M. & Robertson, A. D. Empirical relationships between protein structure and carboxyl pKa values in proteins. *Proteins* **48**, 388-403 (2002).
27. Edgcomb, S. P. & Murphy, P. M. Variability in the pKa of histidine side-chains correlates with burial within proteins. *Proteins* **49**, 1-6 (2002).
28. Wang, J., Cieplak, P. & Kollman, P. A. How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules? *J.Comp.Chem.* **21**, 1049-1074 (2000).
29. Vriend, G. WHAT IF - A molecular modeling and drug design program. *J.Mol.Graph.* **8**, 52-56 (1990).
30. Börjesson, U. & Hünenberger, P. H. Explicit-solvent molecular dynamics simulation at constant pH: Methodology and application to small amines. *J.Chem.Phys.* **114**, 9706-9719 (2001).
31. Qin, J., Clore, G. M. & Gronenborn, A. M. Ionization equilibria for side-chain carboxyl groups in oxidized and and reduced human thioredoxin and in the complex with its target peptide from the transcription factor NF kappa B. *Biochemistry* **35**, 7-13 (1996).




**Space filler:** Screenshot of YASARA's help movie 4.3 "Poisson-Boltzmann Surfaces"



In the last chapter, we started building complete crystallographic unit cells as target structures for force field parameter optimization. We looked at ionizable amino acids and predicted the presence of hydrogen atoms that are too small to be visible in the X-ray density, but still essential to get the forces right. This chapter continues the task by predicting hydrogen bonding networks, which can then be used to infer the most likely protonation state of any ionizable group (not just standard amino acids) as well as the identity of atoms that cannot be reliably identified in the electron density map (e.g. -NH2 and =O have a similar density, so the orientation of Asn and Gln side-chains is ambiguous).

# Assignment of protonation states in proteins and ligands: Combining $pK_a$ prediction with hydrogen bonding network optimization

Elmar Krieger[1], Roland L. Dunbrack Jr.[2], Rob W. W. Hooft, Barbara Krieger[3]

*Methods Mol Biol.* **819**, 405-421 (2012)

*1) Centre for Molecular and Biomolecular Informatics (CMBI) and 3) Netherlands Bioinformatics Centre (NBIC), Route 260, NCMLS, Raboud University Nijmegen Medical Centre, PO Box 9101, 6500 HB Nijmegen, The Netherlands*
*2)Institute for Cancer Research, Fox Chase Cancer Center, 333 Cottman Avenue, Philadelphia PA 1911, USA*
*3)YASARA Biosciences GmbH, Wagramer Strasse 25/3/45, 1220 Vienna, Austria*



## Abstract

Among the many applications of molecular modeling, drug design is probably the one with the highest demands on the accuracy of the underlying structures. During lead optimization, the position of every atom in the binding site should ideally be known with high precision to identify those chemical modifications that are most likely to increase drug affinity. Unfortunately, X-ray crystallography at common resolution yields an electron density map that is too coarse, since the chemical elements and their protonation states cannot be fully resolved.

This chapter describes the steps required to fill in the missing knowledge, by devising an algorithm that can detect and resolve the ambiguities. First, the $pK_a$ values of acidic and basic groups are predicted. Second, their potential protonation states are determined, including all permutations (considering for example protons that can jump between the oxygens of a phosphate group). Third, those groups of atoms are identified that can adopt alternative but indistinguishable conformations with essentially the same electron density. Fourth, potential hydrogen bond donors and acceptors are located. Finally, all these data are combined in a single 'configuration energy function', whose global minimum is found with the SCWRL algorithm, which employs dead-end elimination and graph theory. As a result one obtains a complete model of the protein and its bound ligand, with ambiguous groups rotated to the best orientation and with protonation states assigned considering the current pH and the H-bonding network. An implementation of the algorithm has been available since 2008 as part of the YASARA modeling & simulation program.

## Introduction

Virtually all molecular modeling methods that employ all-atom force fields benefit heavily from 'having the details right'. If a molecular dynamics simulation is run with incorrectly oriented or protonated side-chains, the protein stability can be reduced significantly. Likewise, docking a ligand to a receptor may fail miserably if a wrong tautomer is chosen for a key active site residue[1]. It is therefore a true pity that these important atomic details can normally not be resolved experimentally, since only a tiny fraction of the X-ray diffraction experiments reach the resolution required to locate individual hydrogen atoms or distinguish the heavier elements (which becomes important if groups of atoms can adopt multiple conformations that all fit the X-ray density equally well). One is thus forced to infer the missing details from mainly two sources of information: first, the $pK_a$ values, that in principle allow to determine the probabilities of the various protonation states of a ionizable group at the pH of interest (from the Henderson-Hasselbalch equation, which is in practice complicated by coupling effects between nearby groups[2]). And second, the environment: most importantly the hydrogen bonding possibilities[3], but also potential clashes (which for example favor the smaller oxygen over the larger NH2 group of an ambiguous glutamine side-chain amide group[4]).

While $pK_a$ values can be measured experimentally, only about 500 have been reported for proteins to date[5], so for most purposes they need to be predicted instead. Many different, initially physics-based, $pK_a$ prediction methods have been developed, ranging from simplified models based on Debye-Hueckel theory[6] or electrostatic screening functions[7] to 'high-end methods' that solve the Poisson-Boltzmann equation (which allows to consider the influ-

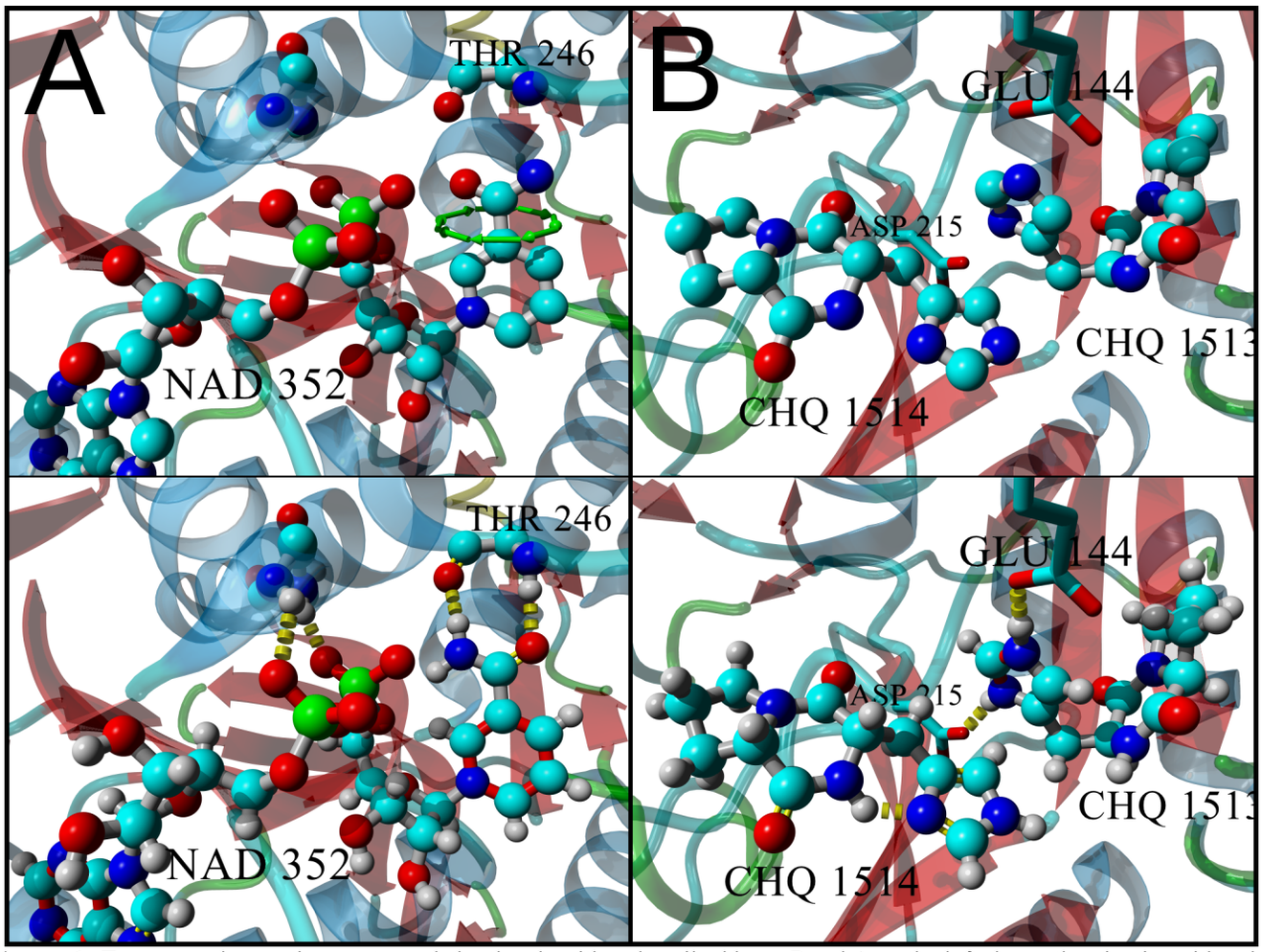



**Figure 1**: Two exemplary assignments made by the algorithm described here. Panel A on the left shows the nicotinamide-adenine-dinucleotide (NAD) cofactor in PDB file 1A5Z. The amide group has been placed incorrectly in the electron density and needs to be rotated by 180° (green arrows, top figure) to form hydrogen bonds with the backbone of residue Thr 246 (bottom). Panel B on the right shows two identical inhibitors (CHQ) sharing the binding site in PDB file 1W1T. While the imidazole ring in the left ligand is kept in the neutral state to accept an internal hydrogen bond, the same imidazole ring in the other ligand is predicted to experience a pKa shift, get protonated, and donate two hydrogen bonds to the nearby residues Glu 144 and Asp 215.



ence of dielectric solute/solvent boundaries and ionic strength on the local electrostatic potential and thus the $pK_a$[8-10]). Surprisingly, most $pK_a$ prediction methods perform about equally well (due to inherent prediction difficulties, *see* **Note 1**), which makes the development of simple and fast empirical methods feasible, that cut some corners[11-15]. The one summarized here belongs to the latter category and has been evaluated in detail before[12].

The H-bonding possibilities in the neighborhood can be readily analyzed to determine the best placement for a hydroxyl hydrogen, or to decide if the side-chain amide group of a glutamine should be rotated by 180°. Unfortunately such ambiguous cases are rarely isolated, and the choice made for one group immediately influences the possible choices for its neighbors, often leading to extensive H-bonding networks that stretch over protein and ligand, and are too large to be solved by brute force evaluation, especially if waters are included. The methods developed to untangle the knot differ in various aspects. Some focus on asparagine/glutamine side-chains[16], others on entire proteins including water[3] and even certain common groups in ligands, using information obtained from the PDB HetGroup dictionary[4] or direct ligand analysis[17]. The methods applied to disentangle the network range from simulated annealing[3] to dynamic programming[17].

The approach described here adds mainly three new features: first, $pK_a$ prediction is included to consider the influence of the pH on the hydrogen bonding network. Second, non-standard amino acids and ligands are fully accounted for with the help of a chemical knowledge library in SMILES format[18]. And third, the use of the SCWRL algorithm[19] allows to find the globally optimal solution almost instantly (the major part of the time is spent on the setup). Since the goal is to predict structural details that cannot be resolved with X-ray spectroscopy, evaluation of the prediction accuracy is a major challenge[20]. While developing and tuning this method, we therefore did not only look at the small but growing number of structures solved by neutron diffraction (which can better resolve hydrogen positions), but also took a pragmatic approach: we compared our prediction results for proteins with those of the three programs NQ-Flipper[16], Reduce[4] and WHAT IF[3], and then manually checked cases where the programs disagreed. These were either truly ambiguous or 'interesting', i.e. offered new insights that allowed us to improve the method. Two assignment examples involving ligands are shown in Figure 1.

# Methods

## 3D structure preparation

Virtually all applications in computational biology that perform energy calculations require 3D structures to be in a clean state, so that force field parameters can be assigned. Since this is a very common procedure also in the other chapters, it is only outlined briefly. All the steps below are for example performed by the 'Clean' command of the YASARA program, which can be accessed via a web server at www.YASARA.org/minimizationserver:

(1) Detect missing bonds and add them, assign bond orders (which are not stored in PDB files).
(2) Rebuild protein side-chains with missing atoms.
(3) Delete terminal protein residues with largely incomplete backbone, that often occur in X-ray structures when the chain enters a disordered region.
(4) Delete atoms that are present more than once at alternate locations, keeping those with the highest occupancy.
(5) Delete residues and chains that overlap significantly with others and are most likely the result of incorrect PDB format usage, like PDB file 1GTV or BTN/BTQ in 2F01.
(6) Add terminal oxygens at the protein C-termini, and capping groups at internal chain breaks (missing X-ray density).
(7) Add missing cysteine bridges between close Cys SG atoms, provided that their positions allow bridge formation[19].
(8) Add missing hydrogens (using the default states "Asp, Glu , His deprotonated" and "Tyr, Lys protonated") to provide a starting point for the following analysis.
(9) Assign force field parameters, at least the charges will be needed in the next steps. In the context of the AMBER force fields[21], ligands can be handled easily using GAFF[22] and AM1-BCC charges[23]. Some of the available tools are Antechamber[24], the AutoSMILES server (www.YASARA.org/autosmiles) or PDB2PQR[25].

## Fast empirical $pK_a$ prediction

The configuration energy function devised here includes the $pK_a$s of ionizable groups, so that the energetic cost of adding or removing protons at the pH of interest can be considered. As mentioned above, the recipe provided is rather simple, knowing that very complex approaches are not guaranteed to perform better (*see* **Note 1**). It can in principle be replaced with any other available method, just make sure that this other method has been validated with a data set of statistically significant size, and has been compared to an optimized jack-knifed null-model (*see* **Note 2**):

(1) Determine the initial default $pK_a$s for protein residues, which are simply the average experimentally measured values. In a rather large set of 541 measured $pK_a$s reported recently[5], the averages are 3.3 for the C-terminus, 3.5 for Asp, 4.2 for Glu, 6.6 for His, 6.8 for Cys, 7.7 for the N-terminus, 10.3 for Tyr and 10.5 for Lys. Alternatively, $pK_a$ prediction often employs so called 'intrinsic $pK_a$s' (*see* **Note 3**).
(2) Consider the electrostatic influence of the environment on the $pK_a$: if there are lots of positively charged residues in the neighborhood, they will repel other protons and make it harder for a ionizable group to get protonated, thus lowering its $pK_a$. Likewise, a negative electrostatic potential will raise the $pK_a$. A very convenient way to estimate the electrostatic potential (ESP) is provided by the Particle Mesh Ewald method[26], which has been developed to efficiently handle long-range electrostatic interactions without a cutoff: first it is part of essentially all molecular simulation programs, and second, it expresses the ESP as the sum of a short-range and a long-range term[26]. The latter replaces point charges with extended Gaussian charge densities and thus yields a smoothed representation of the ESP, which makes it well suited for our purpose: short-range noise is avoided, and there are no singularities, allowing to calculate the ESP directly at the coordinates of the atoms (where it is normally infinite). This way, the estimated $pK_a$ of a given residue is obtained as

$$pK_a = default\ pK_a + \sum_{i=1}^{ionizableAtoms} -A_i * EwaldEnergy_i \quad (1)$$

(3) In the formula above, *default* $pK_a$ is the average $pK_a$ of the residue type from step 1 above, the sum runs over the i ionizable atoms in the group (one in Lys, two in Asp and Glu), '*A*' is the empirical proportionality constant[12] (0.00264 for Asp, 0.00209 for Glu, and 0.00408 for Lys) and $EwaldEnergy_i$ is the smooth long-range portion of the Ewald energy of a charge +1 at the location of the $i^{th}$ ionizable atom in kcal/mol (the energy is used instead of the ESP only for convenience). No parameters are provided for other residues, since there were either not enough measurements available when the method was developed[12] (termini, Tyr, Cys) or there was no improvement (His).
(4) The $pK_a$ prediction could be improved further by considering two additional well-known factors that shift the $pK_a$. First, the desolvation (Born) effect: it costs energy to bury a charge inside the protein where the dielectric constant is lower, which means that the environment cannot shield the charge as well as water. In theory, the Born effect should thus favor the neutral state (raise the $pK_a$ of Asp/Glu and lower the one of Lys/His), but in practice it is found that desolvation mainly increases the magnitude of the $pK_a$ shift[27], which makes it hard to use for empirical prediction schemes[12]. The second factor is a much more helpful indicator: hydrogen bonds. If a group accepts a hydrogen bond, there is less space to bind a proton, and the $pK_a$ will be lowered. Likewise, if a group can immediately use a bound proton to donate a hydrogen bond, the $pK_a$ will be raised. This knowledge could be incorporated into equation (1) above[12], but this is not needed here, since hydrogen bonds are explicitly taken care of in our configuration energy function, which is better

fed with the $pK_a$ before consideration of hydrogen bonds.

(5) Determine the default $pK_a$s for ionizable groups in ligands. While protein side-chains $pK_a$s depend mostly on the environment in the 3D structure, those in ligands are additionally influenced by the local electronic structure, which depends on other functional groups. These effects can of course be considered[28], but they are beyond the scope of this chapter. Instead, the default $pK_a$s are simply obtained by matching the ligand with a library of SMILES strings[18], which encode the potential protonation states and associated $pK_a$s for all common functional groups. Three typical examples are shown in Figure 2, the complete library can be downloaded from www.YASARA.org/grouplibrary.txt, a regularly updated version is distributed with the YASARA program.

```
# Carboxyl group
C?(=O)O?  <4.0
C?(~O)~O  >4.0

# Phosphate group
# Estimated average of
# methyl phosphate (1.54/6.31),
# ethyl phosphate (1.6/6.6),
# sugar phosphates (1.0/6.1)
P(=O)(O?)(O?)OC  <1.38
P(~O)(~O)(O?)OC  <6.33
P(^O)(^O)(^O)OC  >6.33

# Imidazole ring
c?(~n?c?=1)~n?c?=1 <6.95
c?(=nc?=1)n?c?=1   <14.2
c?(~nc?=1)~nc?=1   >14.2
```

**Figure 2:** Description of protonation states and associated $pK_a$s using SMILES strings[18] for three exemplary groups (lowercase characters are aromatic atoms, numbers are used as ring closures). To better express chemical equivalence (and find proton placement permutations), the standard SMILES syntax has been expanded with fractional bond orders: the characters '^' and '~' represent bonds of orders 1.33 and 1.5, respectively. The question mark '?' is a placeholder for any external group, possibly a single hydrogen. E.g. the middle example shows a phosphate group, which carries two protons ('?') below pH 1.38. From pH 1.38 to pH 6.33, one oxygen is protonated, the other two are not, forming equivalent bonds of order 1.5 (therefore carrying a formal charge of -0.5 each). Above pH 6.33, the bond orders of three oxygens are 1.33 (formal charges -0.66, total charge -2). Figure 3 shows the corresponding structures.

## Definition of the configurational energy function

The structure to analyze (e.g. a protein-ligand complex) contains variable groups that can adopt different protonation states (e.g. a carboxyl group), different ambiguous orientations (that yield about the same X-ray density and can thus not be distinguished, e.g. the terminal amide group of an asparagine side-chain), or both at the same time (e.g. the imidazole ring in histidine). To indicate their relation to side-chain rotamers, these different configurations are called 'confimers' here, Figure 3 shows some examples. The total energy $E$ of the system is

$$E = \sum_{i=1}^{Groups} \left( SelfEnergy(C_i) + \sum_{j=1}^{i-1} InteractionEnergy(C_i, C_j) \right) \quad (2)$$

which simply loops over the variable groups and sums up the self-energy of the current confimer $C_i$ and the interaction energies with the current confimers $C_j$ of nearby variable groups. The goal is to choose the confimers such that the energy becomes minimal. A very fast algorithm to find this global minimum has originally been developed for side-chain rotamer prediction[19], and it comes with just one drawback: it requires that all energies are positive (unfavorable, like steric clashes), while our configurational energy function also needs to consider the negative (favorable) hydrogen bonding energies. This fundamental problem can fortunately be resolved by keeping in mind that a well formed hydrogen bond contributes almost nothing to the stability ($\Delta G$) of a protein or a protein-ligand complex, because an equally good hydrogen bond can be formed with surrounding water molecules in the unfolded or unbound state[29]. What really counts is the (positive) energetic cost of missing or sub-optimal hydrogen bonds. So our goal is not to maximize the number of hydrogen bonds (which may lead to incorrect results), but instead to minimize the number of buried unsatisfied hydrogen bond donors or acceptors. Since these all contribute a positive energy penalty, negative energies can be avoided. The following steps are required to calculate the self- and interaction energies in equation 2:

(1) Create a neighbor search grid to quickly find atoms close in space.

(2) Calculate molecular surface areas of the heavy atoms. There are different ways to achieve that[30]. Here, a triangle mesh of the molecular surface is created, then each mesh vertex gets assigned one third of the areas of all triangles it is part of. Finally, the vertex areas are assigned to the closest atoms, then hydrogen areas are transferred to the bound heavy atom.

(3) Assign aromaticity: atoms are simply flagged as aromatic if they are in a planar ring and no atom in the ring forms bonds outside the ring plane.

(4) Preliminarily identify potential hydrogen bond donors and acceptors (*see* **Note 4**).

(5) In each SMILES string[18] in the SMILES library (Figure 2) identify those atoms that are chemically equivalent (i.e. yield the same SMILES strings when used as the first atom in the string), and transfer this knowledge of equivalence to the corresponding atoms in the other SMILES strings of the group. For example in the each of the three SMILES strings that describe a phosphate group (Figure 2), three of the oxygens will be tagged as equivalent, because they truly are in the last of the three strings. At this point, the use of fractional bond orders facilitates the detection of equivalence.

(6) Match all the atoms with the SMILES library. Every group of atoms that matches a SMILES string becomes a 'variable group' with a certain number of confimers, which is initially simply the number of different proto-

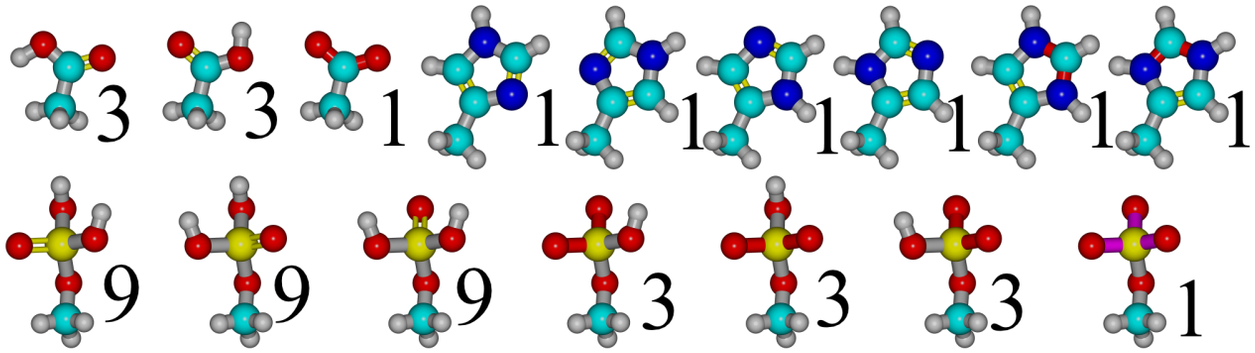



**Figure 3:** Ball&stick models of the three exemplary variable groups from Figure 2 with their confimers. Darkened bonds (red in the electronic version) have bond order 1.5 (except for the bottom right phosphate group, where the bond order is 1.33). The **top left carboxyl group** has maximally 3+3+1=7 confimers: There are three different protonation states with a hydrogen on the first, second or no oxygen. Since the hydrogen can rotate freely, up to three confimers are used to cover various hydrogen positions. Two where the hydrogen forms a hydrogen bond with the two closest potential acceptors (if present), and one where the hydrogen faces away from potential donors and clashing atoms. The **top right imidazole group** has six confimers: Three different protonation states, and since the X-ray electron density does not reliably permit to distinguish carbon from nitrogen, each state is present twice, with the ring rotated by 180 degrees. Finally, the **phosphate group at the bottom** has up to 37 confimers: At low pH <1.38, two hydrogens are present, which can be distributed over the three oxygens in three ways (calculated from the binomial coefficient '3 choose 2'), and since each hydrogen can take up to three positions (see above), there are up to 3*9 confimers. At pH <6.33, only one oxygen is protonated, yielding up to 3*3 confimers. Finally, at pH >6.33 there is only one confimer without protons.



nation states (Figure 2). If an atom matches more than one SMILES string, it is part of the largest variable group.

(7) For each confimer, and for each set of atoms tagged as equivalent in the confimer (step 5 above) sum up the number of hydrogens bound. If it is >0, split the confimer into a set of confimers that cover all potential ways of distributing the hydrogens over the equivalent atoms (the number of confimers can simply be calculated from the binomial coefficient, see Figure 3).

(8) For each confimer, determine the number of freely rotatable hydrogens Nh (e.g. hydroxyl groups) and split it into at most $3^{Nh}$ confimers, that let each of the hydrogens point into up to three different directions: towards the two closest hydrogen bond acceptors (if present), and away from nearby hydrogen bond donors and clashing atoms (*see* **Note 5**).

(9) For each hydrogen bonding atom in a confimer (*see* **Note 4**) determine *DonSites* (the number of bound hydrogens), *AccSites* (the number of H-bond acceptance sites) and *HBSites* (the sum of both).

(10) For each covalent bond, check if it can serve as a 180° rotation axis for a planar group of atoms, where both rotation states yield essentially the same X-ray density (i.e. where each atom falls 'on top' (distance<0.75 Å) of a partner atom on the other side after a 180° rotation). If any pair of partner atoms has different hydrogen bonding preferences, add the two rotation states as new confimers. In proteins, this will for example add two confimers for the amide groups of asparagine and glutamine, and the imidazole ring of histidine (Figure 3). Rotation angles other than 180° (e.g. 120°) are currently not considered.

(11) For each donor hydrogen *i* and acceptor *j* in each confimer, calculate the penalty for being unsatisfied (to help keep track of the sign, the word 'penalty' is used for positive energies):

$$UnsatDonPenalty_i = \max\left(0, -IdealHBEnergy/2 + \sum_{k=1}^{fixedAcceptors} HBEnergy_{i,k}/2 + WaterHBEnergy_i/2\right) \quad (3)$$

(12) where *IdealHBEnergy* is the energy of an ideal H-bond (-25 kJ/mol) and *HBEnergy* is the energy of an actual H-bond (*see* **Note 6**), in this case formed with a nearby fixed (i.e. not part of any variable group) acceptor. So we take the cost of leaving the donor unsatisfied (-*IdealHBEnergy*/2, the division by 2 makes clear that we distribute H-bonding energies equally over donor and acceptor), and add the energies of (usually 0 or 1) hydrogen bonds formed with nearby acceptors and water molecules (*WaterHBEnergy*, calculated from the donor's molecular surface area, *see* **Note 7**). The penalty for unsatisfied acceptors is calculated in a similar way, except that there are often more than one acceptor sites (*AccSites* aka 'lone pairs', step 9), and that acceptors cannot only be satisfied by H-bond donors, but also by cations (*AccIonEnergy*, *see* **Note 8**):

$$UnsatAccPenalty_j = \max\left(0, AccSites_j * -IdealHBEnergy/2 + \sum_{k=1}^{fixedDonors} HBEnergy_{j,k}/2 + \sum_{l=1}^{Cations} AccIonEnergy_{j,l}/2 + WaterHBEnergy_j/2\right) \quad (4)$$

(13) Calculate the self-energy of each confimer (see formula (2) above):

$$
\begin{aligned}
SelfEnergy = \; & pK_a Deviation^2 * 2.5 \\
& + \sum_{j=1}^{Acceptors} \Big( UnsatAccPenalty_{j,reduced} \\
& \qquad + \sum_{k=1}^{fixedSoleAcceptors} Acc\,2\,Penalty_{j,k} \Big) \\
& + \sum_{i=1}^{DonHyds} \Big( UnsatDonPenalty_{i,reduced} \\
& \qquad + \sum_{l=1}^{fixedDonHyds} Hyd\,2\,Penalty_{i,l} \\
& \qquad + \sum_{m=1}^{Cations} HydIonPenalty_{i,m} \\
& \qquad + \sum_{n=1}^{Neighbors} HydClashPenalty_{i,n} \Big) \\
& + UnsatFixedDonAccPenalty
\end{aligned}
\tag{5}
$$

(14) where $pK_aDeviation$ is either 0 (if the confimer's protonation state is the most probable one at the current pH), or $pK_a$-pH ($pK_a$ is taken from Figure 2, either from the current confimer (if $pK_a$<pH) or from the previous one (if the previous $pK_a$>pH)). *Acc2Penalty* is the cost of two sole acceptors (that are not also donors) facing each other (40/*Distance* kJ/mol). *Hyd2Penalty* is the cost of two donor hydrogens getting close (*see* **Note 9**), *HydIonPenalty* is the cost of a donor hydrogen facing a cation (*see* **Note 10**) and *HydClashPenalty* is the cost of a donor hydrogen bumping into any other nearby atom (*see* **Note 11**). Before *UnsatDonPenalty* (3) and *UnsatAccPenalty* (4) are plugged into equation (5), they need to be further reduced (if possible down to 0) by adding potential H-bonding energies from nearby variable groups. Since the confimer of the nearby variable group is at this point undetermined, this can be considered an optimal potential interaction with the neighboring 'confimer cloud'. The potential hydrogen bonds added here must be remembered till step 13 (*PotHBEnergySum*), where the interaction energies of the confimers will be calculated. Finally, *UnsatFixedDonAccPenalty* is the penalty for leaving nearby buried fixed donors, acceptors and ions (that do not belong to any variable group) unsatisfied. It is -0.5 * the sum over the energies of the best H-bonds formed between these nearby atoms and other confimers (but not the current confimer) of the current variable group.

(15) For each confimer pair *i, j* of two interacting variable groups (which have been identified in the previous step) calculate the interaction energy (formula (2)):

$$
\begin{aligned}
InteractionEnergy = \; & \sum_{k=1}^{DonHyds\ i} \sum_{l=1}^{DonHyds\ j} Hyd\,2\,Penalty_{k,l} \\
& + \sum_{m=1}^{SoleAccs\ i} \sum_{n=1}^{SoleAccs\ j} Acc\,2\,Penalty_{m,n} \\
& + \max\Big(0, \sum_{o=1}^{H-Bonds} HBEnergy_o - PotHBEnergySum_i \\
& \qquad - PotHBEnergySum_j \Big)
\end{aligned}
\tag{6}
$$

(16) The interaction energy thus consists of three obvious parts (the penalties for two hydrogens or two sole acceptors facing each other, and the summed up H-bonding energies) and one less obvious term: The summed up potential H-bonding energies (*PotHBEnergySum*, a negative value) which have been added in the previous step 12 to lower the *UnsatDonPenalty* and *UnsatAccPenalty* terms (representing the ideal potential interaction of a certain confimer with the complete 'confimer cloud' at the neighboring variable group) is subtracted here again to avoid double-counting. As a result, the negative H-bonding energy is usually replaced with a positive value (because more potential H-bonds can be formed with a confimer cloud than actual H-bonds with a single confimer). In case *UnsatDonPenalty* and *UnsatAccPenalty* reached 0 already in step 11 or 12, before all potential H-bonds were added, *PotHBEnergySum* might not be low enough now to compensate, in this case the max() function sets the term to zero. This just means that all H-bonding sites of the confimer have been fully satisfied from the beginning (either by water or the neighboring confimer cloud), and can be ignored for this interacting confimer pair. The one and only purpose of this complex compensation scheme is to keep the self- and interaction energies positive, which allows to find the optimum solution quickly, as described below.

## Finding the global minimum of the configurational energy function

The name 'confimer' has been chosen due to its similarity with 'rotamer', which already provides a hint that finding the best confimer for each variable group is exactly the same as finding the best rotamer for each amino acid side-chain in a protein. Consequently, the well developed methodology of protein side-chain prediction can be used without change. Were we employ the SCWRL algorithm, which is extremely fast and essentially guaranteed to find the global energy minimum [19]:

(1) Build an undirected graph, where each variable group is a node (with two or more confimers), and those nodes that interact (have an interaction energy>0 according to equation 6) are connected with an edge.

(2) For each node, discard those confimers whose self-energy is higher than the maximum energy (=self+interaction energy) of another confimer (dead-end elimination).

(3) Break the graph into biconnected components (subgraphs that cannot be split in two by removing a single node). Solve the biconnected components individually for each confimer of the articulation node (i.e. the node that connects a biconnected component to the next one), adding the resulting energies to the self-energies of the confimers. This effectively 'collapses' a subgraph onto its articulation node, thereby reducing the search space until only a single node is left, whose lowest confimer energy is simply the global energy minimum.

(4) Start from the final node, walk along the graph in the reversed direction and determine for each node which confimer contributed to the global energy minimum. Transfer this configuration to the actual 3D structure (adding or deleting hydrogens and turning ambiguous groups around where needed).

An important point that has so far not been mentioned are water molecules. They also participate in the hydrogen bonding network, and can in principle be included in the energy function[3]. But in practice it's quite hard to find a case where a structurally important water uniquely determines the hydrogen bonding network of the protein. Most of the time – thanks to their ability to change from donor to acceptor with just a rotation – waters simply adapt to the solute. One can therefore obtain a useful H-bonding assignment for waters by considering the solute as fixed, finding the water that forms the largest number of potential hydrogen bonds with fixed atoms, choosing the resulting best orientation, fixing this water too and iterating until all waters are assigned. An even simpler alternative is to perform an energy minimization of the waters with any force field, while keeping the water oxygens and the solute fixed.



## Notes

(1) The majority of protein $pK_a$ values have been measured by NMR spectroscopy for proteins in solution. But often, the actual NMR solution structure is not available (or of limited quality[31] and presumably also accuracy), so that one is forced to use a crystal structure to predict solution $pK_a$ values[32]. That's why $pK_a$s may be strongly influenced by features missing in the structure used for the prediction, which obviously adds a considerable amount of random noise. This might explain why protein $pK_a$ prediction accuracy is usually around 0.8 $pK_a$ units (RMSD between predicted and measured $pK_a$), independent of the method used.

(2) When evaluating the accuracy of a $pK_a$ prediction method, the first question is: does the method perform any better than the null-model (the trivial 'prediction'), which just assigns the same $pK_a$ to all ionizable groups of a certain type (e.g. a $pK_a$ of 4.09 to all glutamate side-chains). It is crucial that the null-model $pK_a$s are optimized, i.e. chosen such that the RMSD between null-model $pK_a$ and experimentally measured $pK_a$ is minimal. To avoid bias, this can be done with a jackknife approach, which excludes the $pK_a$ to predict from the null-model optimization[12]. If the null-model was not optimized but instead based on some arbitrary default values (e.g. the experimentally measured $pK_a$ of an isolated glutamate residue), then performing better than this arbitrary null-model would not be a valid proof of usefulness.

(3) Intrinsic $pK_a$ values are those measured for Ala-Ala-X-Ala-Ala pentapeptides, which thus represent the $pK_a$ of residue 'X' in a protein with minimum influence of surrounding residues. In theory, these should be the ideal starting points for $pK_a$ prediction, but for the empirical method described here, we found that the optimal starting points were closer to the average measured $pK_a$s[12].

(4) The elements N, O, S and P are donors if they have a hydrogen bound, metal ions are always 'donors'. The number of potentially accepted hydrogen bonds is determined as follows: Elements O and S accept one hydrogen bond if they are aromatic, and max(0,4-valence) bonds otherwise (the valence is the sum of the bond orders). Phosphorous with <=3 bonds accepts one hydrogen bond. Nitrogens that are planar (sp2) or form >3 bonds don't accept any hydrogen bond, and one otherwise.

(5) The last of the three positions considered for a freely rotating hydrogen is facing away from other hydrogens and clashing atoms. It is estimated quickly by summing up $a^*\boldsymbol{r}/|\boldsymbol{r}|^3$, where $\boldsymbol{r}$ is the vector from nearby metal ions, hydrogen bond donors and their hydrogens (a=1.65) or carbon atoms (a=1.0) closer than 5 Å to the donor atom. The empirical exponent 3 is chosen because the interaction is not purely electrostatic (exponent 2) but also includes Van der Waals repulsion (exponent 13). The rotating hydrogen is then placed in the plane spanned by the summed up direction vector and the hydrogen rotation axis.

(6) A central goal of the configuration optimizer is to reach a high performance. Energies are therefore generally not calculated from all atoms involved (as known from MD force fields), but from the minimum set of atoms required. Consequently, they are mostly 'effective empirical energies' which have been balanced to yield the result considered correct (see Introduction). The energy of a hydrogen bond is defined as a function of the hydrogen-acceptor distance *HADis* and two angle-dependent scaling factors:

$$HBEnergy\ [kJ/mol] = \min\left(0, -25 * \frac{2.6 - \max(HADis, 2.1)}{0.5} * DHAScale * HAXScale\right)$$

where *DHAScale* is 0 for Donor-Hydrogen-Acceptor angles <100, 0..1 for angles 100..165, and 1 for angles >165. *HAXScale* is 0 for Hydrogen-Acceptor-X angles <85, 0..1 for angles 85..95, and 1 for angles >95. 'X' is the atom bound to the acceptor. If the acceptor forms more than one covalent bond, the one with the minimum H-A-X angle (and thus the worst energy) is taken (this accounts for bumps between 'X' and the donor, which lower the quality of the hydrogen bond).

(7) The hydrogen bonding energy with surrounding water molecules is defined as

$$WaterHBEnergy = (-25+2.5) * \frac{MolSurfArea}{6} * \frac{UsedSites}{HBSites} \quad kJ/mol$$

where 2.5 is the 'entropic cost of a H-bond with water' (which ensures that internal H-bonds are preferred), *MolSurfArea* is the molecular surface area of the donor or acceptor including all bound hydrogens in Å$^2$, '6' is the area typically needed per hydrogen bond with water, *UsedSites* is 1 for donors and *AccSites* for acceptors, which is explained like *HBSites* in the main text.

(8) The coordination energy between an acceptor and a cation is set equal to a hydrogen bond when they touch and then decays like an electrostatic interaction (the AccRadii for N,O,P,S are 1.34, 1.14, 2.0, 2.0 Å):

$$AccIonEnergy = \frac{-25}{\max(1, AccIonDis - AccRadius - IonRadius + 1)} \quad kJ/mol$$

(9) The penalty for two polar hydrogens facing each other consists of long-range electrostatic repulsion and short-range VdW repulsion (using a softer exponent 4 instead of the usual 12):

$$Hyd2Penalty = 40/Distance + 40*\max(0, 2.7-Distance)^4 \quad kJ/mol$$

(10) The repulsion energy between a donor hydrogen and a cation is defined accordingly as

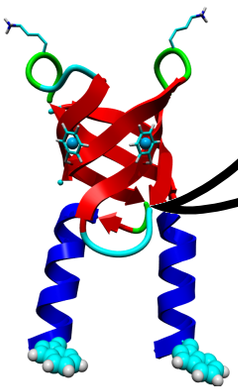
$$HydIonPenalty = \frac{53}{\max(0, HydIonDis - 0.32 - IonRadius + 1)} kJ/mol$$

(11) The penalty for a hydrogen bumping into another atom (with *VdWRadius*) that is separated by more than three covalent bonds is

$$HydClashPenalty = 40*\max(0, 1.2 + VdWRadius - Distance)^4 \quad kJ/mol$$

*Acknowledgment:* We would like to thank the users of the molecular modeling and simulation program YASARA for financing this work.


1. Nabuurs, S. B., Wagener, M. & de Vlieg, J. A flexible approach to induced fit docking. *J.Med.Chem.* **50**, 6507-6518 (2007).
2. Ishikita, H., Stehlik, D., Golbeck, J. H. & Knapp, E. W. Electrostatic influence of PsaC protein binding to the PsaA/PsaB heterodimer in photosystem I. *Biophys. J.* **90**, 1081-1089 (2006).
3. Hooft, R. W. W., Sander, C. & Vriend, G. Positioning hydrogen atoms by optimizing hydrogen-bond networks in protein structures. *Proteins* **26**, 363-376 (1996).
4. Word, J. M., Lovell, S. C., Richardson, J. S. & Richardson, D. C. Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. *J.Mol.Biol.* **285**, 1735-1747 (1999).
5. Grimsley, G. R., Scholtz, J. M. & Pace, C. N. A summary of the measured pK values of the ionizable groups in folded proteins. *Protein Sci.* **18**, 247-251.
6. Warwicker, J. Simplified methods for pKa and acid pH-dependent stability estimation in proteins: removing dielectric and counterion boundaries. *Protein Sci.* **8**, 418-425 (1999).
7. Sandberg, L. & Edholm, O. A fast and simple method to calculate protonation states in proteins. *Proteins* **36**, 474-483 (1999).
8. Warwicker, J. & Watson, H. C. Calculation of the electric potential in the active site cleft due to alpha-helix dipoles. *J.Mol.Biol.* **157**, 671-679 (1982).
9. Yang, A. S., Gunner, M. R., Sampogna, R., Sharp, K. & Honig, B. On the calculation of pKas in proteins. *15*, 252-265 (1993).
10. Antosiewicz, J., McCammon, J. A. & Gilson, M. K. Prediction of pH-dependent properties of proteins. *J.Mol.Biol.* **238**, 415-436 (1994).
11. Czodrowski, P., Dramburg, I., Sotriffer, C. A. & Klebe, G. Development, validation, and application of adapted PEOE charges to estimate pKa values of functional groups in protein-ligand complexes. *Proteins* **65**, 424-437 (2006).
12. Krieger, E., Nielsen, J. E., Spronk, C. A. E. M. & Vriend, G. Fast empirical pKa prediction by Ewald summation. *J Mol Graph Model* **25**, 481-486 (2006).
13. Bas, D. C., Rogers, D. M. & Jensen, J. H. Very fast prediction and rationalization of pKa values for protein-ligand complexes. *Proteins* **73**, 765-783 (2008).
14. Lee, A. C., Yu, J. Y. & Crippen, G. M. pKa prediction of monoprotic small molecules the SMARTS way. *J.Chem.Inf.Model.* **48**, 2043-2053 (2008).
15. Cruciani, G., Milletti, F., Storchi, L., Sforna, G. & Goracci, L. In silico pKa prediction and ADME profiling. *Chem.Biodivers.* **6**, 1812-1821 (2009).
16. Weichenberger, C. X. & Sippl, M. J. NQ-Flipper: validation and correction of asparagine/glutamine amide rotamers in protein crystal structures. *Bioinformatics* **22**, 1397-1398 (2006).
17. Lippert, T. & Rarey, M. Fast automated placement of polar hydrogen atoms in protein-ligand complexes. *J.Cheminform.* **1**, 13 (2009).
18. Weininger, D. SMILES, a chemical language and information system. *J.Chem.Inf.Comput.Sci* **28**, 31-36 (1993).
19. Canutescu, A. A., Shelenkov, A. A. & Dunbrack, R. L. J. A graph-theory algorithm for rapid protein side-chain prediction. *Protein Sci.* **12**, 2001-2014 (2003).
20. Forrest, L. R. & Honig, B. An assessment of the accuracy of methods for predicting hydrogen positions in protein structures. *Proteins*, 296-309 (2005).
21. Duan, Y. et al. A Point-Charge Force Field for Molecular Mechanics Simulations of Proteins. *J.Comp.Chem.* **24**, 1999-2012 (2003).
22. Wang, J., Wolf, R. M., Caldwell, J. W., Kollman, P. A. & Case, D. A. Development and testing of a general AMBER force field. *J.Comp.Chem.* **25**, 1157-1174 (2004).
23. Jakalian, A., Jack, D. B. & Bayly, C. I. Fast, efficient generation of high-quality atomic charges. AM1-BCC model: II. Parameterization and validation. *J.Comput.Chem.* **23**, 1623-1641 (2002).
24. Wang, J., Wang, W., Kollman, P. A. & Case, D. A. Automatic atom type and bond type perception in molecular mechanical calculations. *J.Mol.Graph.Model.* **25**, 247-260 (2006).
25. Dolinsky, T. J. et al. PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations. *Nucleic Acids Res.* **35**, W522-W525 (2007).
26. Essman, U. et al. A smooth particle mesh Ewald method. *J.Chem.Phys.* **103**, 8577-8593 (1995).
27. Edgcomb, S. P. & Murphy, P. M. Variability in the pKa of histidine side-chains correlates with burial within proteins. *Proteins* **49**, 1-6 (2002).
28. Milletti, F. et al. Extending pKa prediction accuracy: high-throughput pKa measurements to understand pKa modulation of new chemical series. *Eur.J.Med.Chem.* **45**, 4270-4279 (2010).
29. Sippl, M. J. Helmholtz free energy of peptide hydrogen bonds in proteins. *J.Mol.Biol.* **260**, 644-648 (1996).
30. Connolly, M. L. Analytical molecular surface calculation. *J.Appl.Cryst.* **16**, 548-558 (1983).
31. Hooft, R. W. W., Vriend, G., Sander, C. & Abola, E. E. Errors in protein structures. *Nature* **381**, 272-272 (1996).
32. Nielsen, J. E. & McCammon, J. A. On the evaluation and optimization of protein X-ray structures for pKa calculations. *Protein Sci.* **12**, 313-326 (2003).



When building a model by homology, a lot of supplementary information about the available templates is required. Which parts of the template structures are well resolved? Where are potential errors that will be propagated to the model? For structure-based alignment correction, one needs to know where the secondary structure elements are located, which residues are buried, which are well conserved and which are part of structurally divergent regions. All these questions are answered by the PDBFinder II database described in this chapter.

# PDBFinder II - a database for protein structure analysis and prediction

Elmar Krieger, Rob W.W. Hooft, Sander B. Nabuurs and Gert Vriend

Published as part of an article about the databases housed at the CMBI:

**A series of PDB related databases for everyday needs**

Joosten RP, te Beek TA, Krieger E, Hekkelman ML, Hooft RW, Schneider R, Sander C, Vriend G

*Nucleic Acids Res.* **39**, D411-D419 (2011)



## Abstract

The PDBFinder II database provides uniform access to data typically required by applications in the fields of protein structure analysis and prediction. These include the sequences of experimentally determined structures, chain breaks, assigned secondary structure (DSSP), residue variability and entropy, hot-spots for insertions and deletions (HSSP), accessibilities, crystal contacts, B-factors, quality indicators, as well as general information parsed from PDB files: experimental methods, resolution, R-factor, authors, compounds, chains, hetgroups and many more. The quality indicators consist of normality Z-scores describing bonds, angles, torsions, planarity, chirality, packing, and inside/outside distribution. Unusual backbone conformations, unsatisfied hydrogen bond donors/acceptors, and flips of Asn, Gln and His side-chains are also reported.

The database is updated weekly and is available as a flat, human readable text file via anonymous FTP from ftp.cmbi.ru.nl/pub/molbio/data/pdbfinder2/. A Python plugin for the molecular modeling program YASARA allows to load PDB structures and visualize the data by coloring residues accordingly (www.YASARA.org/plugins). A separate Python module to directly access the PDBFinder II is also available.

## Introduction

While the Protein Data Bank provides the central resource for protein structures, a lot of additional information can be found in accessory databases spread all over the world. At the CMBI, we maintain the DSSP[1] (assigned secondary structure, residue accessibilities, hydrogen bonds), HSSP[2] (alignments of the PDB sequence against Swissprot and TrEMBL), PDBFinder[3] (all important information from PDB files, including the sequences) and PDBReport[4] databases (a detailed structural analysis with a focus on potential problems). Researchers in structural bioinformatics often need quick, automated access to all these data. So far, this either required separate downloads and parsers for each file format, or was simply impossible, e.g. the PDBReports are only available in human-, but not computer-readable form (www.cmbi.ru.nl/pdbreport).

The PDBFinder II database offers quick access to all these and additional data in a format that is easily readable for humans as well as computer programs. Applications dealing with structure prediction can quickly determine which residues have experimental coordinates, they can correct initial sequence-based alignments by considering secondary structure elements, conserved and buried residues, as well as regions with a high probability of insertions and deletions. The per-residue quality indicators allow to identify the less reliable parts of a structure and build hybrid models consisting of experimentally well determined fragments in multiple templates. Researchers developing new modeling methods can use the PDBFinder II to generate reliable test-sets, that exclude residues involved in crystal contacts or problematic structures altogether.

## Results

### The PDBFinder II file format

The PDBFinder II entry for crambin (PDB ID 1CRN) is shown in Figure 1. In addition to information parsed from the PDB header, which is taken from the PDBFinder data-

```
ID           : 1CRN
Header       : PLANT SEED PROTEIN
 Date        : 1981-04-30
Compound     : Crambin
Source       : Abyssinian cabbage (crambe abyssinica) seed
Author       : W.A.Hendrickson
Author       : M.M.Teeter
Exp-Method   : X
 Resolution  : 1.50
 R-Factor    : 0.114
Ref-Prog     : PROLSQ
HSSP-N-Align : 25
T-Frac-Helix : 0.48
T-Frac-Beta  : 0.09
T-Nres-Prot  : 46
Chain        : _
 Sec-Struc   : 46
  Helix      : 22
   i,i+3     : 3
  Beta       : 4
   Anti-Hb   : 4
 Amino-Acids : 46
  CYSS       : 6
 Sequence    : TTCCPSIVARSNFNVCRLPGTPEAICATYTGCIIIPGATCPGDYAN
 DSSP        : CEECSSHHHHHHHHHHTTTCCHHHHHHHHSCEECSSSCCCGGGCC
 Nalign      : 27777777777777777777777777777777777777777777| 32.9348
 Nindel      : 00000000000000000000101000000000000200000000|  0.0870
 Entropy     : 01003435212411400340434440056330213345401532 3|  0.2728
 Cons-Weight : 97984353786298399739443449821559576552397357 44|  0.6220
 Access      : 42003476225425624776263550276241342957834762 63|  0.4417
 Cryst-Cont  : ____++++__+___+_+++++++_+__++___+__+__+___+++_|  0.4783
 Quality     :  0.6450
  Present    : 99999999999999999999999999999999999999999999|  1.0000
  B-Factors  : 99999999999999999999999999999999999999999999|  0.9975
  Bonds      : 98998979999899899999986988979799999892979999999|  0.9246
  Angles     : 26999969999080898997988998986799966967379969 29|  0.7963
  Torsions   : ?64644647678687546352666375655764644423755543?|  0.5475
  Phi/psi    : ?45545655666665434455646453453654644433754335?|  0.4712
  Planarity  : 99999999999999999999999999999999999999999999|  1.0000
  Chirality  : 08999999999999999899999999999999999999998979|  0.9601
  Backbone   : ??99999999999999991979999999996999986999999??|  0.9476
  Peptide-Pl : ??99999999999999?9?999999999?8999??999999??|  0.9944
  Rotamer    : ??55995995476796969 9?9699899799?98499997998 99?|  0.7833
  Chi-1/chi-2: 44445554545746445554565345454554655554455 4455|  0.4756
  Bumps      : 99999090999999999049909999990099999999999999999|  0.7159
  Packing 1  : 64893976666776540132386566530146633143146312 13|  0.5163
  Packing 2  : 94666534566746554445474347443368431425366535 32|  0.4089
  In/out     : 99999999999999999999999999999999999999999999999|  0.9998
  H-Bonds    : 99999797999999999999999999999999999999999979|  0.9837
  Flips      : 99999999999999999999999999999999999999999999|  1.0000
//
```

**Figure 1**: PDBFinder II entry of 1CRN

base[3], the following fields have been added at the chain-level:

(1) Sequence: the sequence of all residues with experimentally determined coordinates. Contrary to the original PDBFinder, the sequence contains chain break markers '-'. This allows to easily align it with the complete sequence deposited in Swissprot or given in the SEQRES field of the PDB file.

(2) DSSP: the secondary structure assigned by DSSP[1], with 'C' indicating 'coil' to facilitate parsing.

(3) Cryst-cont: residues involved in crystal contacts are flagged with a '+' sign.

(4) In the remaining fields, numbers or Z-scores are mapped to the range [0..9]. The formulas used are described in the header of the PDBFinder II file.

(5) Nalign: the number of aligned sequences at this position in the HSSP-file.

(6) Nindel: the sum of insertions and deletions.

(7) Entropy: and Cons-weight: the sequence entropy and conservation weights as defined in [2].

(8) Access: residue accessibilities, '0' is completely buried and '9' is maximally exposed.

(9) Quality: overall estimator for the chain quality, obtained by averaging the 'Phi/Psi', 'Backbone' and 'Packing 1' fields described below (which are among the most reliable quality indicators). High resolution X-ray structures reach values around 0.75, while NMR structures determined from only a few experimental restraints can be found around 0.3.

(10) For the following quality-related fields, the mapping is chosen such that '9' corresponds to 'perfect' and '0' to 'requires attention'. Exceptions are the fields 'Torsions', 'Phi/Psi', 'Chi-1/2', 'Packing 1' and 'Packing 2', where '5' indicates about the average of high resolu-

tion X-ray structures, '9' corresponds to 'suspiciously good' and '0' to 'treat with caution', according to the WHAT_CHECK output[4].

(11) Present: 9 minus the number of missing atoms per residue.

(12) B-Factors: average crystallographic B-factor per residue.

(13) Bonds and Angles: absolute Z-score of the largest bond or angle deviation per residue according to the Engh&Huber parameters[5].

(14) Torsions: Average Z-score of the torsion angles per residue. This one and the following Z-scores including 'in/out' are calculated from the distributions found in the internal WHAT IF database of high resolution X-ray structures[6].

(15) Phi/Psi: Ramachandran Z-score per residue. No value can be determined for the N- and C-terminal residues, which is indicated by a question mark '?'.

(16) Planarity: Z-score of the side-chain planarity. Residues without a planar side-chain always score '9'.

(17) Chirality: Average absolute Z-score of all 'improper dihedrals' per residue, defined by one central and three bound heavy atoms, excluding planar groups. Gly always scores '9'.

(18) Backbone: Number of similar backbone conformations found in the database, determined by superimposing stretches of five residues. No score can be obtained for the N- and C-terminal two residues. If less than 10 hits are found, there are not sufficient data to perform the following two checks for peptide plane flips and rotamers (indicated by question marks).

(19) Peptide-Pl: RMS distance of the backbone oxygen from the oxygen in similar backbone conformations found in the database. Low scores indicate that the peptide-plane may have been flipped.

(20) Rotamer: Probability that the side-chain rotamer (chi-1 only) is correct. Gly, Ala and Pro always score '9'.

(21) Chi-1/Chi-2: Z-score for the side-chain chi-1/chi-2 combination.

(22) Packing 1: Three-dimensional packing quality Z-score[7].

(23) Packing 2: Second packing quality Z-score.

(24) In/Out: Absolute Z-score for the residue accessibility.

(25) Bumps: Sum of bumps (i.e. difference between the van der Waals- and the actual distance) per residue.

(26) H-Bonds: 9 minus the number of unsatisfied hydrogen bonds. Additional penalties: 1 is subtracted for buried unsatisfied backbone nitrogens, 4 for unsatisfied side-chain hydrogen bonds.

(27) Flips: Asparagine, glutamine and histidine side-chains that need to be flipped in the optimum hydrogen bonding network[8] are scored with '0', all other residues with '9'.

(28) Unless indicated otherwise, numbers at the right border (separated with a pipe symbol '|') are the average over the chain, multiplied with 0.9. This average is calculated before the individual residue values are saturated to fit into the interval [0..9] and can therefore lie outside the corresponding interval [0..1].

## Database interfaces

A Python plugin for the molecular modeling program YASARA (www.YASARA.org/plugins) allows to load PDB structures, automatically retrieve the corresponding PDBFinder II entry via HTTP and map the information onto the structure. An example is shown in figure 2, where trypsin has been colored according to the HSSP conservation weights. A separate Python module to access the PDBFinder II directly is available from www.YASARA.org/biotools.

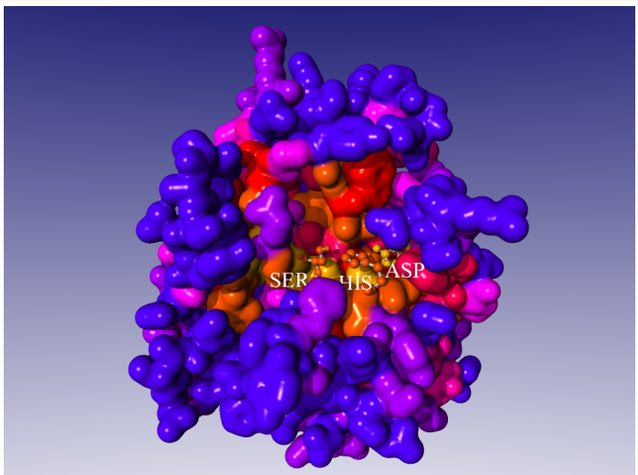
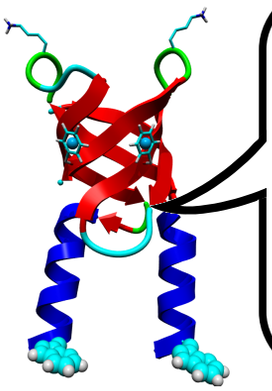



**Figure 2:** Crystal structure of trypsin from Atlantic salmon[9], colored by HSSP conservation weights. Blue corresponds to 'not conserved' and yellow to 'completely conserved'. The Ser-His-Asp catalytic triad is indicated.


1. Kabsch, W. & Sander, C. Directory of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features. *Biopolymers* **22**, 2577-2637 (1983).
2. Dodge, C., Schneider, R. & Sander, C. The HSSP database of protein structure-sequence alignments and family profiles. *Nucleic Acids Res.* **26**, 313-315 (1998).
3. Hooft, R. W. W., Sander, C., Scharf, M. & Vriend, G. The PDBFINDER database: a summary of PDB, DSSP and HSSP information with added value. *Comput.Appl.Biosci.* **12**, 525-529 (1996).
4. Hooft, R. W. W., Vriend, G., Sander, C. & Abola, E. E. Errors in protein structures. *Nature* **381**, 272-272 (1996).
5. Engh, R. A. & Huber, R. Accurate bond and angle parameters for X-ray protein structure refinement. *Acta Cryst.A* **47**, 392-400 (1991).
6. Hooft, R. W. W., Sander, C. & Vriend, G. Verification of protein structures: side-chain planarity. *J.Appl.Cryst.* **29**, 714-716 (1996).
7. Vriend, G. & Sander, C. Quality control of protein models: Directional atomic contact analysis. *J.Appl.Cryst.* **26**, 47-60 (1993).
8. Hooft, R. W. W., Sander, C. & Vriend, G. Positioning hydrogen atoms by optimizing hydrogen-bond networks in protein structures. *Proteins* **26**, 363-376 (1996).
9. Schroder, H. K., Willassen, N. P. & Smalas, A. O. Structure of a non-psychrophilic trypsin from a cold-adapted fish species. *Acta Crystallogr.D Biol.Crystallogr.* **54**, 780-798 (1998)



Molecular modeling often involves a performance/accuracy trade-off. The higher the level/accuracy/flexibility of the computations, the longer it takes to complete them. E.g. in homology modeling it often helps to build multiple models from different templates and alignments. Model refinement using molecular dynamics simulations also takes a long time. Consequently a normal desktop PC is not enough to finish the task in time. One possible solution is to buy an expensive supercomputer. Another one is to link those PCs that are already present and use them when their owners sleep@home. This approach is described in the following chapter.

# Models@Home: distributed computing in bioinformatics using a screensaver based approach

Elmar Krieger and Gert Vriend

*Bioinformatics* **18,** 315-318 (2002)


## Abstract

**Motivation:** Due to the steadily growing computational demands in bioinformatics and related scientific disciplines, one is forced to make optimal use of the available resources. A straightforward solution is to build a network of idle computers and let each of them work on a small piece of a scientific challenge, as done by Seti@Home, the world's largest distributed computing project.

**Results:** We developed a generally applicable distributed computing solution that uses a screensaver system similar to Seti@Home. The software exploits the coarse-grained nature of typical bioinformatics projects. Three major considerations for the design were: 1) Often, many different programs are needed, while the time is lacking to parallelize them. Models@Home can run any program in parallel without modifications to the source code. 2) In contrast to the Seti project, bioinformatics applications normally are more sensitive to lost jobs. Models@Home therefore includes stringent control over job scheduling. 3) To allow use in heterogeneous environments, Linux and Windows based workstations can be combined with dedicated PCs to build a homogeneous cluster. We present three practical applications of Models@Home, running the modeling programs WHAT IF and YASARA on 30 PCs: Force field parameterization, molecular dynamics docking, and database maintenance.

**Availability:** Models@Home is freely available including source code and detailed instructions from www.YASARA.org/models.


## Introduction

No matter how smart the algorithm, it is always too slow to do the job - overnight on a desktop PC. And when PCs have finally become fast enough - the algorithm has become obsolete, replaced by a new approach, with fewer approximations. That is a well known experience in bioinformatics, almost comparable to Murphy's law. The usual approach is to split the problem into little jobs that can be executed independently. The execution time is then reduced by 1/n, with n being the number of computers working in parallel. As noted by Amdahl[1], this ideal speedup cannot be reached in practice, leading to the more realistic formulation

$$S_n = \frac{1}{(1-PF)+PF/A_n} \quad (1)$$

in which $S_n$ it the the total speedup when going from one to n processors, PF is the "parallelizable fraction" of the program (i.e. the fraction of the total execution time that can be reduced by working in parallel), and $A_n$ is the speedup of the algorithm's PF on n processors.

Algorithms in bioinformatics tend to be applied to a large number of different targets, e.g. all ORFs in a genome, all sequences in CASP, or all potential drug candidates in a library. If each computer is assigned one target, these jobs are completely independent and parallelize perfectly, as no communication overhead is required: PF is close to one, $A_n$ and $S_n$ are close to n. The coarse-grained nature of typical bioinformatics projects is probably one of the reasons why most follow-ups to Seti@Home fall into this area, e.g. FightAids@Home or Folding@Home. An overview of other approaches to distributed computing, clusters and computational grids is given in table 1.

## Methods

Models@Home has been designed for use in typical departmental situations, in which a large number of Linux or Windows workstations is idle for about 16 hours a day. The program consists of several functional units, as shown in figure 1. Detailed installation instructions are available from www.YASARA.org/models.

| Program name | www Address | S | O | W | U | Description |
|---|---|---|---|---|---|---|
| Beowulf | www.beowulf.org | + | + | - | + | Software for dedicated Linux clusters |
| Berkeley NOW | now.cs.berkeley.edu | + | + | - | + | Inhouse network of workstations |
| Condor | www.cs.wisc.edu/condor | + | + | + | + | Distributed computing library |
| Cosm | www.mithral.com | + | + | + | + | Distributed computing library |
| Distributed.net | www.distributed.net | - | - | + | + | Cracking encryption keys |
| Entropia | www.entropia.com | - | $ | + | - | Distributed computing for Windows |
| FightAids@Home | www.fightaidsathome.com | - | - | + | - | Drug design using Entropia |
| Folding@Home | www.stanford.edu/group/pandegroup/Cosm | - | - | + | - | Protein folding simulations using Cosm |
| Globus Project | www.globus.org | + | + | - | + | Computational grid for Unix |
| Legion | legion.virginia.edu | - | + | - | + | Worldwide computer for Unix |
| Models@Home | www.yasara.org/models | + | + | + | + | Screensaver cluster for any program |
| Mosix | www.mosix.cs.huji.ac.il | + | + | - | + | Software for dedicated Linux clusters |
| Seti@Home | setiathome.berkeley.edu | - | - | + | + | Search for extraterrestrial intelligence |
| United Devices | www.ud.com | - | $ | + | - | Distributed computing for Windows |

**Table 1:** List of different approaches to clusters and distributed computing. S = source code available, O = run your own programs ($ requires payment), W = available for Windows, U = available for Unix/Linux.



## Supervisors

Users of the cluster run programs called "supervisors". These are typically applications that keep track of the work that has to be done, but do not do it themselves. Instead they cut it into pieces and submit the individual jobs to the Models@Home job scheduler (via an interface of C functions or a Python class). The supervisors thus form the part that has to be adapted specifically for a certain application. This is most easily done by adding a "submit to Models@Home"-command to the inner loop of an existing script. All other aspects of Models@Home are totally general.

## Working clients

As the name suggests, they do the actual work. Initially, only the Idle Detection (ID) module is active on these computers. The ID module is highly operating system specific and has been derived from existing open-source screensavers. The Linux version is based on Jamie Zawinski's XScreensaver (www.jwz.org/xscreensaver), with additional changes to Linux configuration files, the Windows equivalent on Bill Buckel's work. The modified sources are available from www.YASARA.org/models.

As soon as the computer is idle (i.e. no mouse movements or key-strokes occur for 15 minutes), the ID module launches a graphical screensaver (ScrSaver) and the execution module (Exec). Both have been implemented in an operating-system independent way based on the SDL library, including SDL_net for the TCP/IP interface (www.libsdl.org). Users running batch jobs can configure the screensaver to become active only if they are not logged in.

The Exec module contacts the server. This contact message also contains the time stamps of files that should be kept up to date (these names are stored with other information, like the server's IP address, in the local configuration file "cluster.cnf").

If there are jobs waiting in the queue, the Exec module receives a job description (the name of the application to run, command line parameters, scripts), file updates if needed, and the data files required to complete the job (e.g. specific PDB files). These are stored in the working directory /job. Exec runs the requested application (App 1 or App 2 in figure 1) and waits until the job has finished. Result files are sent back to the server. Large applications consisting of hundreds of files should be permanently installed on the clients, small and compact programs can be transmitted as part of the job description.

If a user on a working client terminates the screensaver, Exec kills the running application and notifies the server that the job could not be completed. No attempt is made to put the application on hold and continue at a later time, as this would negatively affect the cluster's performance (it is not known if and when the client will be available again). Jobs taking a long time must therefore return checkpoint files in reasonable time intervals. Usually this does not require a modification of the source code, as most time-intensive programs already have these mechanisms built in (e.g. every molecular dynamics program can save a snapshot of the current simulation state).

Exec can also be run as a stand-alone module (without the screensaver) on the nodes of a dedicated cluster. It then allows a very efficient use of cluster resources: While many batch queuing systems generate the maximum overhead when the cluster is busy (by asking clients sequentially if they want to accept a job), Models@Home works the other way round: Clients ask the server for a job, only reaching the maximum number of requests (and thus network load) when the cluster is not used at all.

## Server

The server is the link between supervisors and working clients. It manages the job queue "cluster.job" and distributes jobs according to their priority. It also handles the transfer of job data files from a job-specific directory on the supervisor (via NFS) to the working directory on the working client (via TCP/IP). Job results travel back in the reversed direction.

The latest file updates are stored in subdirectory /updates and transmitted to working clients to replace outdated versions. (In principle any file on the working client can belong to this update group, including the Models@Home software and the actual applications).

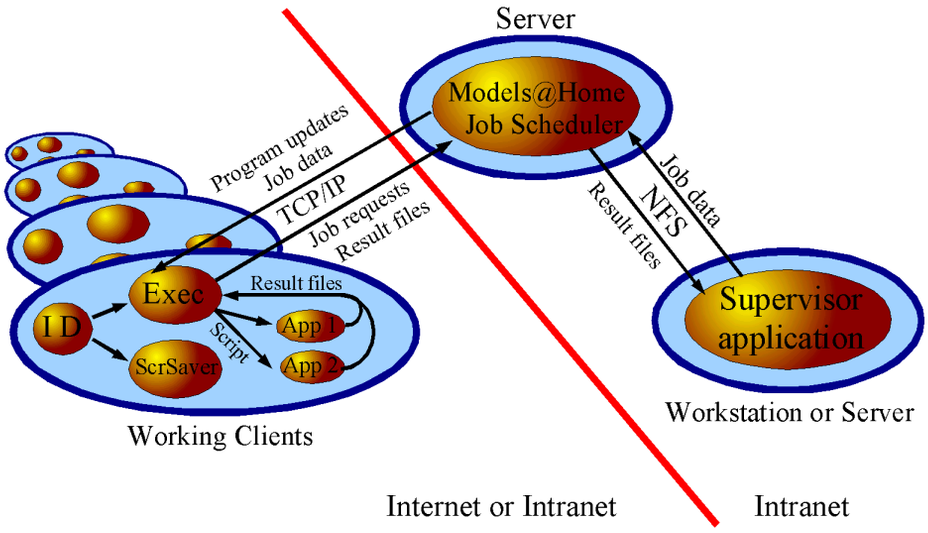



**Figure 1:** Models@Home data flow. Refer to the methods section for details. Abbreviations: ID = Idle Detection module, Exec = Program execution module, ScrSaver = graphical screensaver module, App 1+2 = application 1+2, TCP/IP = Transmission Control Protocol / Internet Protocol, NFS = Network File System.

The server also collects "still alive messages" from working clients. If a client does not send such a message for a given time period, it is assumed to have "disappeared without notice" (e.g. a power failure) and the job is retransmitted to another client.

## Implementation

Models@Home is based on a straightforward client-server architecture using the TCP/IP protocol as shown in figure 1. Every client has the Models@Home software and all applications that are too large for repeated transmission installed locally (which is most easily done by remote administration). The main component of Models@Home is a screensaver that becomes active when the client is not used for a certain time period. Beside displaying some graphical animations, the program contacts the central server and requests a new job. Jobs can be added via a Python or C interface. The screensaver detects when a program has finished, returns the specified result files and requests a new job. As all this is done by the Models@Home software, it is possible to run any program in parallel without modifications to the source code, as long as user input is not needed.

The above procedure applies to ideal conditions only. A number of additional features had to be implemented to cope with problems encountered in practice:

(1) **Hard resets and power failures:** Computers occasionally stop working, making it impossible to notify the server about the job interruption. Clients therefore send a "still alive message" in fixed time intervals. Especially short intervals require a permanent inter- or intranet connection and exclude any dial-up clients.

(2) **Program development and bugs:** If the program is under development, it is of course not reasonable to continuously reinstall it on every client. The Models@Home communication protocol therefore includes the time-stamps of selected files including executables. These are automatically updated if newer versions are available on the server.

(3) **Security issues:** In the majority of cases, Models@Home will be used in the intranet only (normally protected by a firewall). Either because this provides already enough computer power, or because the programs are not freely distributable, or because of the large amount of work required to support users outside the department. If the "world wild web" is targeted, the update feature mentioned above must be deactivated, and it is up to the developer to ensure that the executed programs cannot do any damage (e.g. a simple SAVE command in a program script could already overwrite system files in a Windows environment).

## Discussion

The Models@Home environment has been installed and tested on 30 mixed Linux / Windows PCs at the CMBI. The following paragraphs describe some of the applications and concentrate on the aspects related to parallel execution, the very details will be described elsewhere.

### Molecular dynamics docking

A protocol was developed where docking is performed with YASARA during a molecular dynamics simulation (see Di Nola et al.[2] for an early description of a comparable method), which inherently considers both ligand and protein flexibility (Krieger et al., submitted 2001). The ligand is shot towards the protein, allowing side chain reorientations during complex formation, followed by a short MD simulation, an energy minimization (simulated annealing) and the evaluation of the final energy.

To sample conformational space reasonably well, thousands of molecular dynamics simulations with different initial orientations of ligand and protein are required. As these are completely independent, the supervisor could spawn all docking jobs at once, making molecular dynamics docking an ideally "coarse-grained" application for dis-

tributed computing.

## Force Field parameterization

Selecting force field parameters that optimally fit a given force field equation is a lengthy procedure, usually requiring extensive validation studies. In addition it is very difficult to obtain an internally consistent parameter set. We therefore let a force field "parameterize itself" while energy minimizing protein structures[3]. This was done with Monte Carlo moves in parameter space, that were accepted if the resulting force field did less damage to high resolution X-ray structures and at the same time improved models built by WHAT IF[4].

Each parameter optimization cycle required an energy minimization of 50 protein structures. The supervisor therefore spawned 50 jobs. Each job included the PDB file of the structure, as well as a YASARA script to do the energy minimization and to finally calculate the RMSD from the initial structure. Only this RMSD value was sent back as a result. The 50 RMSDs were averaged and used as a progress indicator.

For this application of Models@Home, the rate limiting step was the time it took to minimize the largest protein in the set. All RMSDs had to be known before the quality of the current force field could be estimated and new jobs could be spawned. More computers would thus not have improved the performance. This is an example of a "medium-grained" application.

## Database maintenance

The CMBI hosts a large number of databases, many of which are not mirrored but generated in-house. Especially those related to protein structure are often time consuming to maintain, like the PDBREPORT database, that lists anomalies in protein structures[5] (www.cmbi.ru.nl/pdbreport). It contains one entry for every PDB file, and is mainly used by modelers to find optimally suited templates.

To keep this database up to date, the supervisor compares it with the PDB once a week, deletes obsolete PDB reports and spawns jobs to create new ones. All these jobs are independent and parallelize perfectly.

## Conclusion

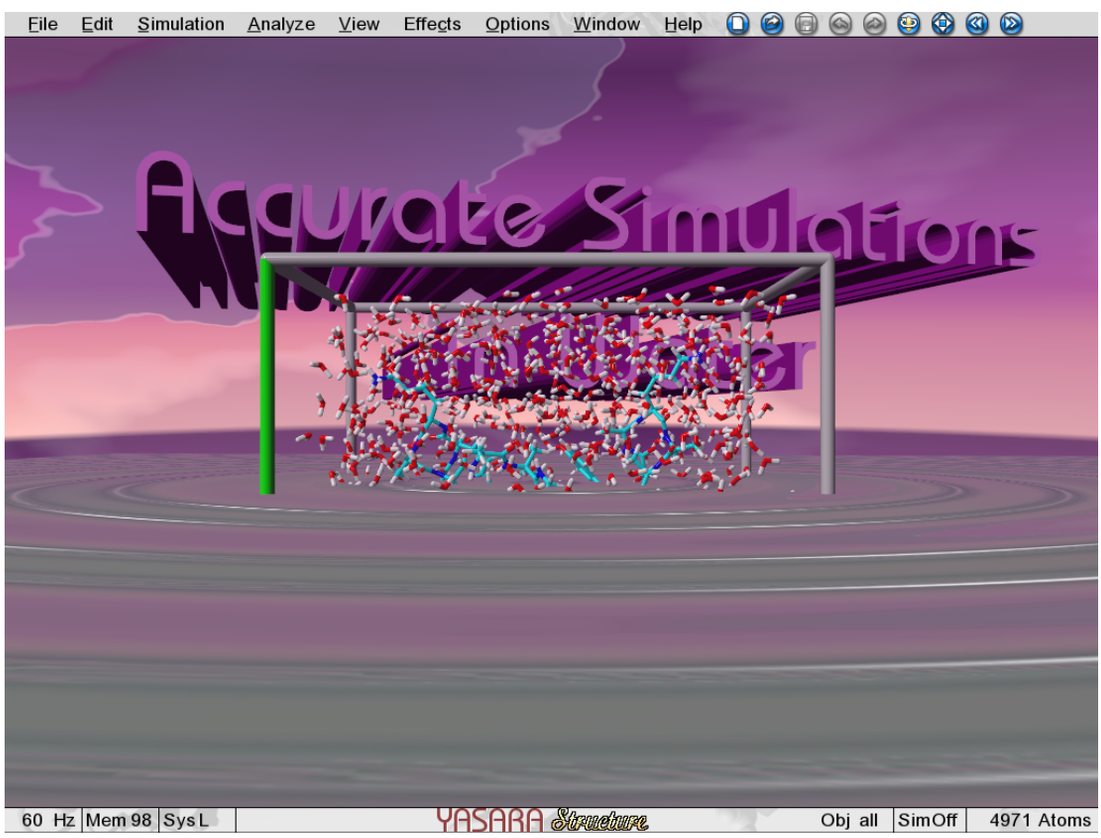
Models@Home provides a flexible environment for parallel execution of different applications without the need to modify any of these programs. It is therefore well suited for bioinformatics, where both the turn-over of different software packages and the requirement for computer power are huge.

Models@Home can be reconfigured for use with different programs without making changes to the source code. It is freely available and can be downloaded from www.YASARA.org/models.

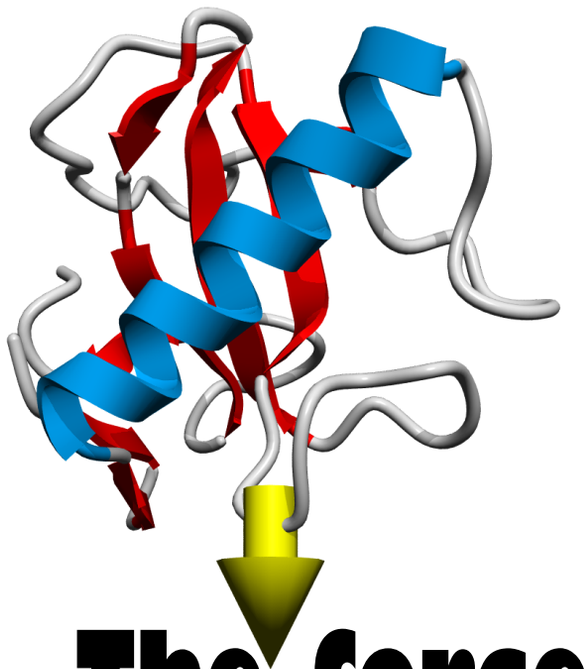
*Acknowledgements:* We would like to thank all researchers at the CMBI for participating in the Models@Home screensaver project.

1. Amdahl, G. (1967) The validity of the single processor approach to achieving large scale computing capabilities. AFIPS Conf. Proc. 30, 483-485
2. Di Nola, A., Roccatano, D. and Berendsen, H.J. (1994) Molecular dynamics simulation of the docking of substrates to proteins. Proteins 19, 174-1823.
3. Krieger, E., Koraimann, G. & Vriend, G. (2002) Increasing the precision of comparative models with YASARA NOVA - a self-parameterizing force field. Proteins 47, 393-402.
4. Vriend, G. (1990) WHAT IF: A molecular modeling and drug design program. J. Mol. Graph. 8, 52-56
5. Hooft, R.W.W, Vriend, G., Sander, C. and Abola, E.E. (1996) Errors in protein structures. Nature 381,272

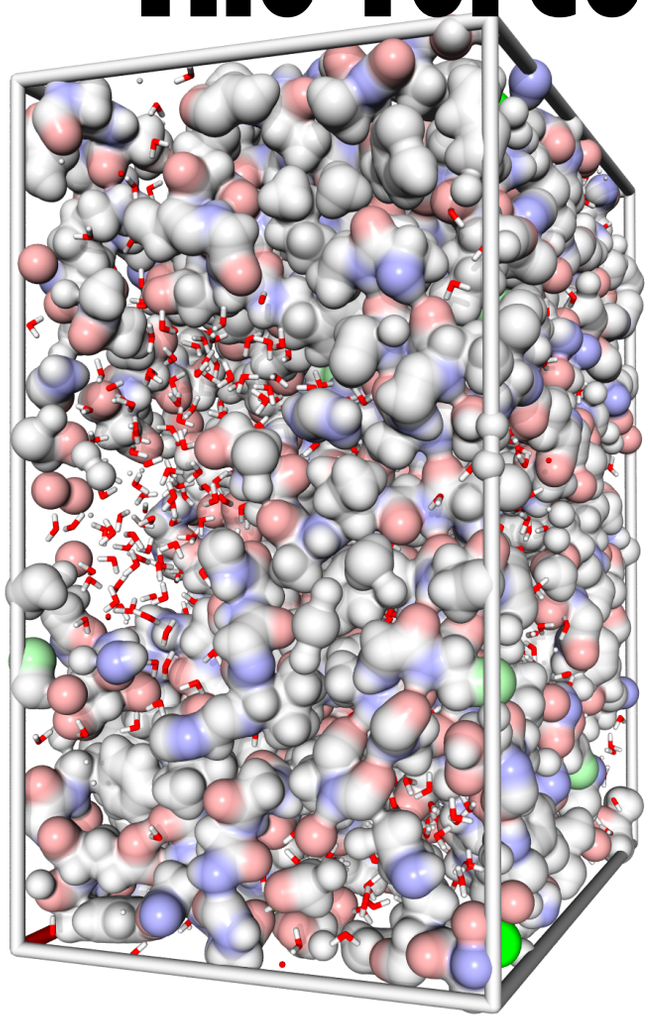




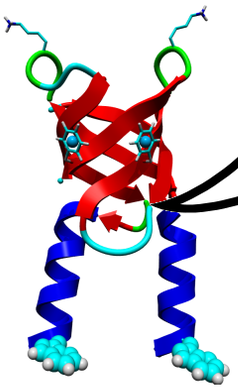

**Space filler:** Screenshot of YASARA's help movie 3.3 "Accurate simulations in water"



# The force fields



3



Proteins are huge molecules, consisting of thousands of atoms. That's why it is not feasible to apply the tools of quantum chemistry on large simulation time scales. Instead we need simplified empirical energy functions, which are designed to capture the most important features but leave out the fuzzy details. One such energy function is described here: the NOVA force field. With NOVA, it is possible to energy minimize homology models and actually improve them, which can be harder than often believed. Note that in 2012, the NOVA force field equation in Figure 2 was changed to the one of the YASARA force field (second next chapter) to support multiple CPU threads and AVX2. The resulting new NOVA force field is not only much faster, but also more accurate than shown here, see the 'ForceField' command in the user manual for details.

# Increasing the precision of comparative models with YASARA NOVA - a self-parameterizing force field

Elmar Krieger, Günther Koraimann and Gert Vriend

*Proteins* **47**, 393-402 (2002)




## Abstract

One of the conclusions drawn at the CASP4 meeting in Asilomar was that applying various force fields during refinement of template-based models tends to move predictions in the wrong direction - away from the experimentally determined coordinates. We have derived an all-atom force field aimed at protein and nucleotide optimization in vacuo - NOVA - which has been specifically designed to avoid this problem. NOVA resembles common molecular dynamics force fields, but has been automatically parameterized with two major goals: 1) Not to make high resolution X-ray structures worse and 2) to improve homology models built by WHAT IF. Force field parameters were not required to be physically correct, instead they were optimized with random Monte Carlo moves in force field parameter space, each one evaluated by simulated annealing runs of a 50 protein optimization set. Errors inherent to the approximate force field equation could thus be canceled by errors in force field parameters. When compared to the optimization set, the force field did equally well on an independent validation set and is shown to move *in silico* models closer to reality. It can be applied to modeling applications as well as X-ray and NMR structure refinement. A new method to assign force field parameters based on molecular trees is also presented. A NOVA server is freely accessible at www.YASARA.org/servers


## Introduction

The search for Nature's folding function has been a tempting scientific adventure ever since Linus Pauling predicted the α helix back in 1951[1]. As an accurate quantum chemical calculation of the true energy function is still hardly feasible for macromolecules, one works with approximations, like the AMBER[2], CHARMm[3] or GROMOS[4] molecular dynamics force fields.

When developing a new force field, the first step is to set up a general equation that matches the various forces present in the studied system. Then one defines rules to derive force field parameters from quantum chemical calculations or experimental measurements on (usually) small molecules. Here we took a different approach and just defined three goals:

(1) For every global (or lowest accessible local) minimum of the true conformational free energy, a minimum of NOVA should lie close by.

(2) The regions around the minima of NOVA need to be as smooth as possible and thereby facilitate energy minimization algorithms.

(3) NOVA should be a function of solute atom coordinates only, the solvent must thus be implicitly included. (See Roux & Simonson[5] for a recent review of implicit solvent models).

The force field was allowed to "parameterize itself" while trying to optimally fulfill these goals. This was achieved by randomly changing force field parameters and evaluating the "fitness" of the resulting force field with a protocol step by step matching the three goals defined above:

(1) Energy minimization of high resolution X-ray structures: The smaller the RMSD from the initial structure, the closer are the NOVA minima to reality.

(2) Energy minimization of homology models built for high resolution X-ray structures: The smaller the RMSD to the experimental structure, the better is the energy landscape suited for getting there. (Other methods for smoothing and reducing the height of energy barriers include umbrella sampling and soft-core potentials[6-8]).

(3) All energy minimizations are done as *in vacuo*.

To make such a global search in force field parameter

space computationally feasible, the number of optimized parameters had to be kept small. Precisely known parameters (i.e. equilibrium bond lengths and angles) were not optimized. Well known parameters (i.e. bond stretching and angle bending force constants) came from the AMBER force field and were rescaled together using two scaling factors: one for the bonds, and one for the angles. All other parameters (e.g. Van der Waals interactions and off-center point charges) were optimized independently. To further reduce computational requirements, the energy minimization algorithm searched for NOVA's closest minimum. It can therefore only be guaranteed that NOVA has minima close to real protein structures, but not that these minima are also global ones.

Algorithms that search for global minima with big steps in conformational space (e.g. *ab initio* fold prediction[9]) can be applied safely if the search is restricted to a specific region with additional data (like NMR NOESY restraints). NOVA is useful for applications that require a search for a local minimum near by, like refinement of experimental low resolution structures, models built by homology or docked complexes. The force field is shown to significantly reduce the Cα RMSD between experimental structures and theoretical models during an energy minimization.

## Methods

The NOVA force field (protein + nucleotide optimization in vacuo) has been implemented as part of the newly developed interactive real-time molecular dynamics program YASARA ("Yet Another Scientific Artificial Reality Application", www.YASARA.org). It looks like common molecular dynamics force fields, with the total energy being expressed as a sum of individual contributions: Bonds, angles, planarity, Van der Waals, and electrostatic terms. Most negative point charges are placed outside the nuclei (off-center charges), Van der Waals interactions are modeled by Born-Mayer Exp6 instead of the familiar Lennard Jones 12-6 potentials. Planarity is treated by least-squares plane fitting instead of improper torsions.

### Molecular trees define the chemical environment

One of the aims during the development of NOVA was the possibility to extend the force field to ligands without the need for manual intervention. This was achieved by using molecular trees (Figure 1) instead of predefined atom types to assign the force field parameters. Normally a topology file lists all atoms by name (e.g. N, H, CA, 1HA, 2HA, C, and O for Glycine) and assigns at least an atom type and often also a point charge. Bond lengths, angles etc. are specified for all combinations of atom types.

In contrast, YASARA builds a molecular tree to define the chemical environment of every atom, and then chooses force field parameters based on the closest reference tree found in the NOVA definition file (electronic supplement). Starting from every atom in the molecule (the root) the program follows the various branches (the chemical bonds) of the molecule - up to a certain recursion depth (usually 3).

An example is the molecular tree built from the Cβ atom of phenylalanine (Figure 1). Bond types (single, double, triple and resonance) are an integral part of the tree. These types are taken from a connectivity table that contains atom names and bonds for every residue or ligand. These tables can be generated automatically by analyzing heavy atom coordinates, predicting their hybridization state and adding hydrogens where needed. This is for example done by the Dundee PRODRG Server[10] (http://davapc1.bioch.dundee. ac.uk/programs/prodrg/prodrg.html). Each molecular tree can also be written as a single string, e.g. for the Cβ of Phe:

CH2(-C(-NH-C)-C(=O)-N)-C(÷CH÷C) ÷CH÷C

Comparable approaches have been suggested before (e.g. by Levitt[11] or Weininger[12]). Here we extend this approach to assign a complete set of force field parameters.

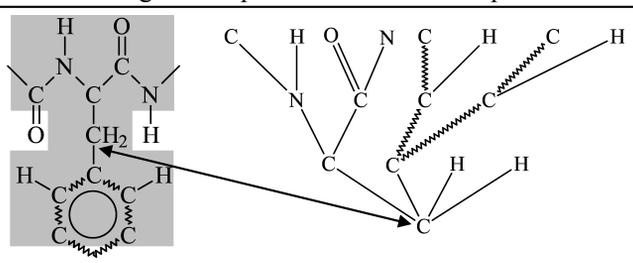


**Figure 1**. A phenylalanine residue (left). The Cβ carbon atom marked with an arrow serves as the root for creating the molecular tree shown on the right. All atoms in the gray area are within recursion depth 3 of the root and thus part of the tree. Beside the topology, the tree stores data about the bond types as well: single, double and resonance bonds are shown (the weaker resonance effects of the peptide bond are not considered). From top to bottom, the parts of the tree are called "depth 3","depth 2","depth 1" and "root". A comparable tree is generated for every atom in the simulated system, force field parameters are then assigned based on the most similar tree in the force field definition file.

### Building reference trees

The NOVA definition file (available from www.YASARA.org/nova) does not explicitly specify equilibrium bond lengths and angles. These are extracted from the 25 highest resolution X-ray structures in the PDB, with less than 30% sequence identity, obtained from the PDBSELECT algorithm[13] (a list with PDBID codes of all proteins used in this work is also available from the above address). Missing hydrogen atoms were added with the WHAT IF hydrogen bonding network optimizer[14], and then relaxed to the closest energy minimum with the AMBER force field[1] (parm94), while all heavy atoms were kept fixed. For each atom in these proteins a molecular tree was built. The distances between the root and depth 1 atoms (the bond lengths in Figure 1) and between any two depth 1 atoms (the bond "angles") are stored within the tree. If two atoms share the same local covalent structure, their molecular trees are identical. Bond lengths (and also an-

$$E_a = \sum_{j<a} 0.5 * k_j * (R_{j0} - R_j)^2 + 0.5 * l_p * R_p^2 + \sum_{i<a} (A_i * e^{-B_i * R_i} - C_i / R_i^6) + \sum_{k<a} \sum_{m} \sum_{n} \frac{q_m * q_n}{4\pi\varepsilon_0 * R_{mn}}$$

$$\mathbf{F_a} = \sum_{j} k_j * (\mathbf{R_{j0}} - \mathbf{R_j}) - l_p * \mathbf{R_p} + \sum_{i} (\mathbf{R_i} / R_i) * (A_i * B_i * e^{-B_i * R_i} - C_i / R_i^7) + \sum_{k} \sum_{m} \sum_{n} \frac{\mathbf{R_{mn}} * q_m * q_n}{4\pi\varepsilon_0 * R_{mn}^3}$$

**Figure 2.** The YASARA NOVA force field. Atom distances are named R, equilibrium values $R_0$. Vectors are shown in bold print. The energy contribution of atom a ($E_a$) is the sum over all chemical bonds (to atom j, with bond stretching force constant $k_j$, j<a), plus a planarity term ($0.5*l_p*R_p^2$), plus the sum over all non-bonded Van der Waals interactions (with atom i, using EXP6-potential parameters $A_i$, $B_i$ and $C_i$, i<a), plus the electrostatic Coulomb-interactions between all m point charges on atom a and n point charges on atom k (k<a). More details are given in the methods section and the electronic supplement.

gles) were therefore averaged over identical trees. This way the program obtained a set of partly residue-dependent bond lengths and angles, that can capture features which are missed when using residue-independent parameters.

## Bonds

Chemical bond stretching is described by a harmonic potential. Bond lengths ($R_{j0}$ in Figure 2) are taken from the reference trees. The initial bond stretching force constants came from the AMBER Parm94 set. During force field parameter optimization, one common scaling factor (parameter 1 in Table I) was assigned to the force constants. The final optimized values ($k_j$ in Figure 2) are listed in the NOVA definition file. To associate a force constant to a type of bond, slightly modified molecular trees are used (as in the case of VdW parameters): Instead of choosing any of the two bonded atoms, the bond itself becomes the root of the tree. There are thus always two atoms with depth 1 and the bond type is the same for both of them.

## Angles

Bond angles are treated like true bonds between 1-3 bonded atoms (Urey-Bradley method). Equilibrium distances ($R_{j0}$ in Figure 2) are taken from the reference trees, the initial angle bending force constants were converted to the required distance dependent form from the AMBER Parm94 set. Again one common scaling factor (parameter 2 in Table I) was assigned to all the angle bending force constants. The final optimized values ($k_j$ in Figure 2) can be found in the NOVA definition file.

We chose the Urey-Bradley approach for two reasons:

(1) Numerical stability: In many typical NOVA applications, very high temperatures (5000 K) temporarily cause bond angles to approach 180°. This creates problems with angle dependent formulations which contain a singularity at 180° and assign very large forces close to this angle, that can trap part of the molecule in an unrealistic local minimum (mainly if it belongs to a planar group).

(2) At least qualitatively, the Urey-Bradley method implicitly contains a bond/angle cross-term which is normally missing. (A change in bond lengths influences the bond angles and vice versa.)

## Dihedrals

Accurate potentials are more difficult to obtain for torsions. We therefore decided to optimize all parameters. To achieve this goal without making the set of optimization parameters too large, we reduced the torsion forces to the repulsion between 1-4 bonded atoms. 1-4 interactions are thus treated exactly like non-bonded interactions. However, because Lennard-Jones 12-6 potentials do not model the torsion energy properly, we chose the more flexible Born-Mayer Exp6 potentials[15], combined with "distant geometry links".

A difficulty with this approach is the hydrogen bonding problem: The electrostatic attraction between the point charges on polar protons and H-bond acceptors is normally not large enough to compensate for the Van der Waals repulsion. This requires a large reduction of the Van der Waals radii of polar protons, which in turn leads to unrealistic torsion energy profiles that must be corrected with additional terms. By using negative off-center point charges, we increased the electrostatic attraction between proton and acceptor sufficiently to reproduce the experimental H-bond lengths of about 1.9 Å, while still keeping realistic VdW parameters.

## Distant Geometry Links

Many planar groups contain charged atoms in close proximity. E.g. the terminal hydrogen atoms HH12+HH22 and HH21+HE in Arginine are separated by four bonds, but lie very close to each other. The distance is about 2.3 Å, leading to repulsive Van der Waals forces. The atoms are all positively charged, adding further repulsive forces with a non-negligible influence on the molecular geometry. But these forces are already implicit in the equilibrium bond lengths and angles taken from the high resolution structures. Distant geometry links (DGLs) define such critical atom pairs and link them with a pseudo-bond to exclude them from the calculation of non-bonded interactions.

## Planarity

All force field terms considered so far were a function of the distance between two atoms. Planarity of atom groups is one of the features that cannot be based on atomic distances only, as out-of-plane bending is accompanied by only small changes in distances. Here we present a method that differs from the normally used "improper dihedrals". It is fast and has some advantageous features when used with the NOVA force field. Our approach calculates the optimal plane through all members of a planar group. Knowing the normal vector of the plane, a force towards the plane is applied to every atom. For a given atom $A_i$, this force is simply the distance from the plane ($R_p$ in Figure 2) times the "plane stretching force constant" $l_p$ (pa-

rameters 3+4 in Table I): $F_p = -R_p * l_p$. A compensating force $-F_p/n$ is applied to all the n atoms bound to $A_i$. While not entirely conserving energy, this approach offers the advantage that the least-squares plane fitting has to be done only once per planar group every time step, and that the resulting normal vector can be used to apply non-spherical Van der Waals potentials in DNA base pair stacking.

## Van der Waals Interactions

The NOVA force field uses Born-Mayer potentials[15] to describe interatomic forces. This function consists of an attractive $R^{-6}$ term and a short-range exponential repulsion term:

$$E = A * e^{-B*R} - C / R^6 \qquad (1)$$

Our reasons for choosing this function were:

(1) The Born-Mayer Exp6 function contains three adjustable parameters and thus allows us to shift the root independent of the minimum, whereas the more common Lennard Jones 12-6 potential always has its root at 0.89 times the distance of its minimum.

(2) As there is no need to minimize numerical noise as in long-time MD simulations, a simple look-up table can be used to avoid the very expensive evaluation of the exponential term. This approach is particularly handy because the Exp6 function requires special care at short distances: When R approaches zero, the repulsive term reaches A (see equation 1) whereas the attractive part tends towards -infinity. The potential thus becomes attractive again at close separations, somehow resembling "nuclear fusion". We therefore replaced the interval from 0 to the first root of the third derivative with an $R^{-12}$ damping function.

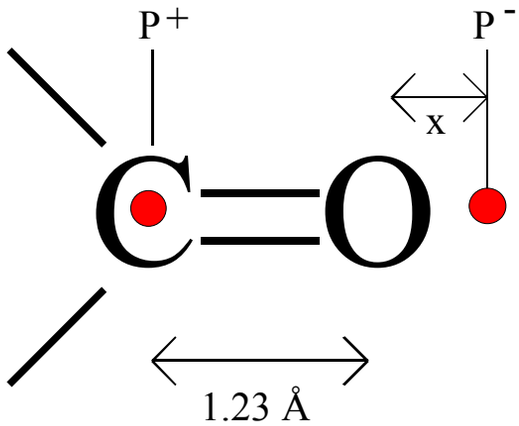



**Figure 3**. Placing off-center point charges as a linear combination of atom coordinates. If the distance x of the partial charge $P^-$ from the O nucleus was 0.8 Å, the weighting factors i, j for coordinates O, C to obtain $P^-$ would be: $P^- = O + (0.8 / 1.23) * (O - C)$, $P^- = (0.8 / 1.23 + 1) * O - (0.8 / 1.23) * C$ and thus i=1.65, j=-0.65. In this case, the position of the point charge depends on the length of the C=O bond.

Initial parameters were taken from a table published by Mirsky[16], manually adjusted by up to 5% so that secondary structure elements did not fall apart and a stable starting guess was available, then parameters were optimized for all atom pairs, no combining rules were used. To keep the number of parameters below a reasonable limit, all interactions involving sulfur were not optimized but taken from AMBER Parm94 and transformed to Exp6 format.

The Exp6 parameters make the largest contribution to the optimization set (parameters 14 to 43, Table I).

## Electrostatics

The electrostatic forces are added by assigning point charges to the atoms. To maximize the level of consistency with the remaining force field, all electrostatic parameters were optimized. This required a description of the essential features with a minimal set of 9 parameters (Table I). These included the charges and also their positions, if placed outside the nucleus. Using off-center point charges was required mainly for hydrogen bonds. As numerical long-time stability is not an issue, charges can be assumed mass-less. Forces are thus calculated between the charges, but they act on the associated nuclei. After each change in atom positions, the coordinates of the point charges are recalculated based on the following rules:

(1) Central charges: these are simply placed at the coordinates of the nucleus.

(2) Charges with positions that can be obtained as a linear combination of atom coordinates: A typical example is the lone pair of the carbonyl group (Figure 3).

(3) Charges with positions that require the calculation of a vector product: The lone pair coordinates of sp3 hybridized atoms (like the oxygen in hydroxyl groups or water, Figure 4) cannot be determined with a simple linear combination of atom coordinates. The point charges are placed in a plane N defined by vectors a and c (Figure 4), where c is a x b. a and b are themselves linear combinations of atom coordinates.

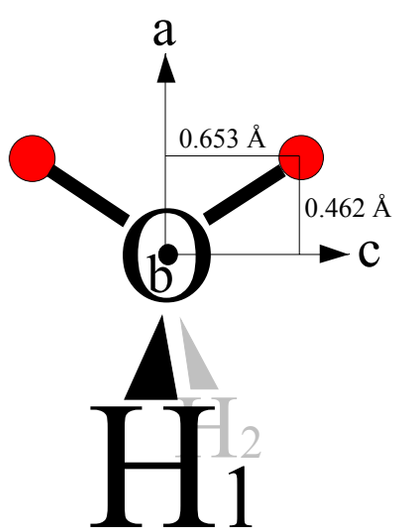



**Figure 4**. Placing off -center point charges at sp3 hybridized atoms. This example shows the application of the method to the ST2 water model. The required parameters are the number of atoms and weighting factors needed to calculate the vectors a and b, the number of charges placed in the plane defined by a and c (c = a x b) and finally for each of these charges the plane coordinates and size.

Initial guess values for the various charges were obtained from the experimentally measured dipole moments of small molecules, the ST2 parameters[17] were used for hydroxyl groups. Ionic charges were set to 0.3e. Then the parameters were optimized (Table I).

| Parameter | Parameter description |
|---|---|
| 1 | Common scaling factor for all AMBER bond stretching force constants |
| 2 | Common scaling factor for all AMBER angle bending force constants |
| 3 | Planarity force constant of peptide plane |
| 4 | Planarity force constant of all planar sidechains (D,E,F,H,N,Q,R,W,Y) |
| 5 | Charge c at H (+c) and N (-c) in peptide bond |
| 6 | Charge c at C (+c) and O (-c) in C=O groups |
| 7 | Distance from O-nucleus of the negative lone "pair" in C=O groups |
| 8 | Charge c at H (+c) and N (-c) in aromatic rings (H,W) and NE/HE of R. |
| 9 | Charge c at H (+c) and N(-2c or -3c) at $NH_2$ and $NH_3$ groups (N,Q,K,R,N-term.) |
| 10 | Ionic charge (D,E,K,H-protonated,R,N-terminus,C-terminus) |
| 11 | Charge c at C (+2c) and O (-c) in deprotonated carboxyl groups (E,D,C-term.) |
| 12 | Distance from O-nucleus of the negative lone "pair" in carboxyl groups |
| 13 | Charge c at H (+c), O (-c per lone pair) and C (+c) at hydroxyl groups (S, T, Y, D-protonated, E-protonated, C-terminus protonated) |
| 14-16 | Born-Mayer parameters of H - H interaction |
| 17-19 | Born-Mayer parameters of H - C interaction |
| 20-22 | Born-Mayer parameters of H - N interaction |
| 23-25 | Born-Mayer parameters of H - O interaction |
| 26-28 | Born-Mayer parameters of C - C interaction |
| 29-31 | Born-Mayer parameters of C - N interaction |
| 32-34 | Born-Mayer parameters of C - O interaction |
| 35-37 | Born-Mayer parameters of N - N interaction |
| 38-40 | Born-Mayer parameters of N - O interaction |
| 41-43 | Born-Mayer parameters of O - O interaction |

**Table I**. The optimization parameters of the NOVA force field. Equilibrium bond lengths and angles are taken from high resolution X-ray structures without optimization. Bond stretching and angle bending force constants come from the AMBER force field and are rescaled (parameters 1 and 2 above).

## Optimization and validation sets

The aim was to create an optimized force field that does not make highest resolution X-ray structures worse while at the same time improves approximate models during an energy minimization procedure. The generation of the required optimization and validation sets of proteins (each one consisting of 25 high resolution structures and 25 models of high resolution structures) was done automatically based on a list of highest quality PDB chains with less than 30% sequence identity, no chain breaks, resolutions better than 1.9 Å and R-factors below 0.19, generated by the PDB-SELECT program[13].

For each of these chains, the script searched for a modeling template in the corresponding FSSP file[18]. If there existed a template that allowed modeling the chain without insertions or deletions, and had less than 93% sequence identity, the model was built with WHAT IF[19] and added to the group of models (M), otherwise the chain became part of the structures group (S). 93% was the lowest value that allowed to reach the required number of 50 structure-model pairs. The first 25 odd entries of M and S were taken as optimization set, the even entries as validation set. The 25 entries in group S of the optimization set were also used to generate the reference trees.

## The practical limit of fold prediction

Comparisons of experimental crystal structures indicate limits on the accuracy that can be reached in structure determination experiments. They also reveal practical limits of fold prediction.

Using the PDBFINDER database[20], we identified chains with identical sequences, that have been solved by different authors and refinement programs at resolutions better than 1.9 Å. For these structure pairs, Cα, backbone and heavy-atom RMSDs were calculated, defining the experimental uncertainty of coordinates obtained at high resolution (top and bottom outliers were removed). We used these values (0.48 Å for Cα, 0.95 Å for all heavy-atoms) to define how much the force field may modify a structure during an energy minimization before we know that it got worse.

## Force field optimization methods

The NOVA force field described in this study has been optimized by Monte Carlo moves in parameter space. After every step, the quality of the force field was evaluated by running "simulated annealing" molecular dynamics simulations for the 50 structures in the optimization set with YASARA and the following protocol: The non-bonded force cutoff was set to 10.5 Å. 100 steps of steepest descent minimization with a maximum step size of 0.05 Å removed any sources of conformational stress that might lead to a collapse of the following simulation. Velocity vectors were initialized to average values found at 298 K4 followed by 3800 integration steps of the equations of motion with the leapfrog algorithm, using a time step of 2 fs for electrostatic plus Van der Waals interactions and 1 fs for all harmonic forces including planarity.

The random initial velocities were required to avoid optimization towards force field vectors with zero length, that are guaranteed not to introduce errors. The distortion at the beginning made sure that a realistic parameter set with the ability to "pull the structure back to where it belongs" was obtained. Every 20 fs, all velocity vectors were scaled by 0.9, the protein was thus slowly frozen. After 3.8ps, the time steps were reduced by 50% to 1 fs and 0.5 fs, respectively, for another 200 cycles. Finally the $C_{\neg\lambda}$, backbone and heavy-atom RMSDs were calculated (with respect to the starting structure (group S) or the modeling target (group M)). The heavy atom RMSDs of all 50 proteins were averaged and used as a progress indicator. A move in

parameter space was accepted with a probability of

$$p = \exp ( - (RMSD_{now} - RMSD_{best}) / 0.00045)$$

The value of 0.00045 for kT was empirically chosen so that progress was steady but local moinima could still be escaped. A total of 43 force field parameters were subjected to this minimization procedure (Table I). Each Monte Carlo move was done by picking one parameter randomly and then either increasing or decreasing it 1 to 10 times the minimal step size. The minimal step size was predefined for every parameter, and equivalent to the final precision required (0.002e for charges, 2% for scaling factors, planarity force constants and VdW contact energies, 0.01 Å for VdW radii and VdW potential roots).

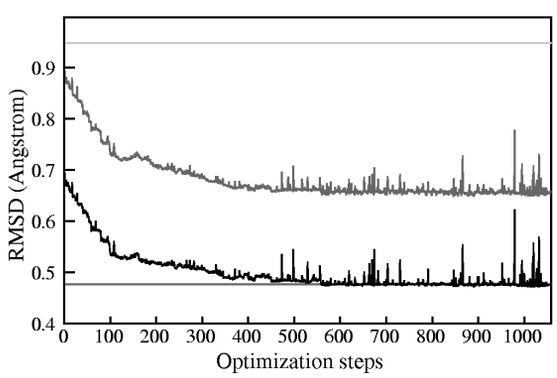



**Figure 5**. Force field optimization progress shown for the 25 high resolution structures in the optimization set. The average heavy atom (upper curve, gray) and Cα RMSDs (lower curve, black) after ~4 ps simulated annealing runs are drawn as a function of the optimization step. The two horizontal lines mark the border of experimental uncertainty (i.e. 0.95 Å for the heavy atom- (top) and 0.48 Å for the Cα-RMSDs (bottom) observed if the same structure is solved at high resolution by different authors and refinement programs). Anything above these lines surely got worse during the minimization. To make the optimization procedure computationally feasible, every simulated annealing run lasted only 4 picoseconds (and did not always reach the energy minimum). Exhaustive simulated annealing runs that proceed till the energy minium is reached are shown in Figure 6.

## Force field evaluation methods

After the optimization converged, the force field was evaluated with an extensive minimization: 250 steps of steepest descent and long 40 ps simulated annealing runs without initial velocities. Time steps were 1 and 0.5 fs for the first 4 ps and 2 and 1 fs for the remaining 36 ps. Force field energies were calculated without a cutoff distance every 200 fs. If the energy did not drop during five of these measurements (1000 integration steps), the energy minimum was reached and the procedure stopped (corresponding to horizontal lines in Figure 8). This avoided the problem that simulated annealing does not stop at the true energy minimum if a cutoff distance is used (it proceeds further to the minimum of the truncated energy function, leading to an increase in true energy).

## Computational requirements

The force field optimization required about 20000 hours of CPU time. The calculations were done on 26 PCs at the CMBI, using Models@Home, a freely available screen-saver that turns a network of normal, non-dedicated PCs into a distributed computing cluster (www.YASARA.org/models). Two months of computer time could be saved by implementing the NOVA force field in Assembly language.

# Results

## Force field optimization

Our goal was to parameterize a force field for energy minimization of proteins, that 1) does not "mess up" high resolution X-ray structures and 2) moves predicted models closer to reality. Most parameters were not chosen based on measured or calculated physico-chemical properties, but instead freely optimized with Monte Carlo moves in force field parameter space. Each move was evaluated with energy minimizations (by simulated annealing) of a 50 protein optimization set: 25 high resolution X-ray structures and 25 homology models built by WHAT IF[19].

The parameter optimization progress is shown in Figure 5 for the 25 high resolution structures. Three distinct regions are visible: During the first 100 optimization steps, the force field improved rapidly. The damage done to the Cα coordinates during the 4 ps minimization decreased by 0.15 Å. At step 100, the "low hanging fruits" were gone, the first local minimum was reached. From here on, a Monte Carlo algorithm that can escape local minima was obligatory. Nevertheless, there was slow but steady progress till step 450. From step 450 on, improvement was minimal but still measurable. Around step 600, the optimizer finally passed the experimental uncertainty barrier of 0.48 Å (Figure 5) for Cα atoms: Below 0.48 Å RMSD, it is impossible to decide whether the structure got worse during the minimization or not. The heavy atom RMSD barrier lies however much higher, at about 0.95 Å. This can be attributed to the influence of crystal contacts on surface rotamers, which can not be exactly determined experimentally and thus increase the RMSD. After 1000 steps (and thus 50000 simulated annealing runs) the procedure converged.

## Force field evaluation

The above results apply to the optimization set only, and were obtained for short 4 ps simulating annealing runs. (By making the minimization time short enough, one can stay arbitrarily close to the initial structures.) The truly important question is: How far are the NOVA minima away from reality? This can only be answered with an extensive simulated annealing run that proceeds till the energy converges. The result is shown in Figure 6 for an independent validation set of another 25 structures.

Before parameter optimization, the force field undoubtedly made the 25 high resolution structures worse.

The energy minima lay 0.15 Å (heavy atom RMSD) and 0.39 Å (Cα RMSD) above their experimental boundaries (Figure 6). During optimization, the minima moved closer to reality by 0.30 Å (heavy atoms) and 0.27 Å (Cα). The Cα RMSD thus came close to the boundary, while the heavy atom RMSD crossed it and converged 0.15 Å below. Figure 7 shows the results for all 100 proteins involved.

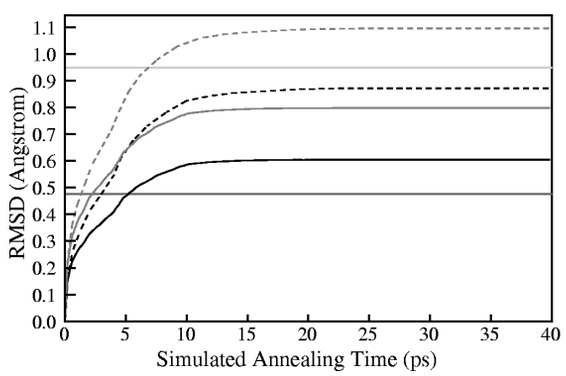



**Figure 6**. Extensive minimization of the structure validation set (25 proteins). The two horizontal lines mark the experimental boundaries described in Figure 5. The average heavy atom (gray curves) and Cα RMSDs (black curves) are shown as a function of simulated annealing time. Dashed curves were obtained with initial force field parameters, solid curves with final optimized parameters.

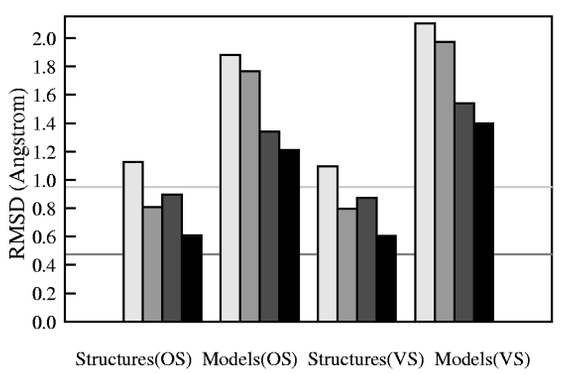



**Figure 7**. Force field parameter optimization results for optimization sets (OS) and validation sets (VS) of 25 structures and 25 models. For each of the four groups, the following values are shown: Average heavy atom RMSD before and after parameter optimization (bright gray and gray bars) and Cα RMSD before and after optimization (dark gray and black bars). RMSDs are measured after a 40 ps simulated annealing run. The two horizontal lines mark the experimental boundaries described in Figure 5. Time dependent results for the structure validation set (group 3) are shown in Figure 8.

## Model improvements

Not to mess up high resolution structures can be regarded as a basic requirement. But to be of practical use, the force field must also be able to move models towards reality. We concentrate on the evaluation of Cα RMSDs to indicate that a true improvement in backbone geometry and not just rotamer prediction accuracy was obtained. It is important to note that WHAT IF was *only* used to mutate the side chains, the backbones of the templates were simply copied to the models.

Figure 8 shows the energy minimization results for the 25 models in the validation set. There are two different regions to deal with: If the model is already very close to the true structure (Cα RMSD < 0.9 Å) it gets slightly worse during the minimization, otherwise it is significantly improved (up to 0.25 Å as in the case of the 1CYO model (top curve in Figure 8)). This result was to be expected: the closer one gets to the true structure, the more accuracy in the force field is required to improve the model.

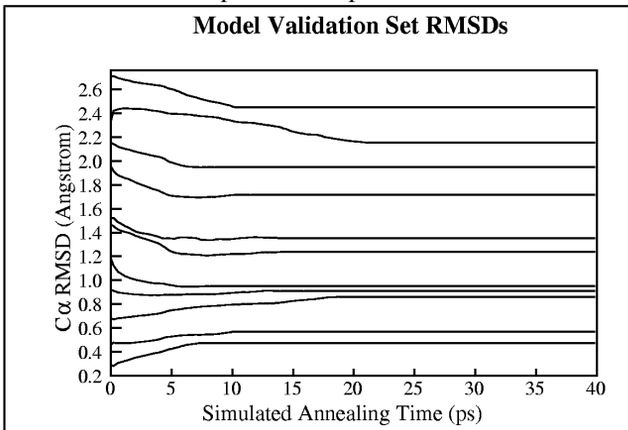



**Figure 8**. Extensive minimization of the model validation set (25 proteins) with the optimized force field parameters. 11 non-overlapping of 25 trajectories are shown.

The solution is obvious: Not to energy minimize if the model is closer than 0.9 Å to its high resolution X-ray structure. Because this RMSD is of course not known at the time of modeling, the decision must be based on different grounds. We derived the following empirical rule from the optimization set:

Only minimize a model if template resolution (Å) divided by sequence identity (%) is larger than 0.04.

We applied this rule to the validation set and obtained the results shown in Figure 9. Initially, minimizing the models clearly made them worse (Cα RMSD increased from 1.36 to 1.54 Å). During the parameter optimization, the model RMSD dropped from 1.54 Å to 1.40 Å, 0.04 Å above the initial RMSD without minimization. By energy minimizing only the models that match the selection rule, we found a true improvement: The backbone moved on average 0.111 Å (Cα) closer to reality (bottom curve in Figure 9).

To investigate the performance of NOVA relative to other force fields, we ran exactly the same energy minimization protocol also with the AMBER force field. While NOVA with the initial parameters did clearly worse, it is apparent from Figure 9 that the optimization procedure allowed to turn around and go into the right direction.

## Force field energy

Ideally, one would like the RMSD to decrease further than the 0.111 Å obtained above, all the way down to the experimental limit of 0.48 Å. It is obvious from Figure 8, that the proteins "get stuck" too early. Two reasons are possible: 1) The NOVA energy function is not accurate enough, or 2) the simulated annealing procedure is not adequate to find the way down.

Low temperature simulated annealing allows backbone shifts and reorientations of flexible surface rotamers, but certainly not a complete flip of a buried tryptophan. If such a rotamer is initially not predicted correctly, the minimization can easily get stuck in a wrong local minimum. To verify this hypothesis and clarify the possible involvement of point 1), we compared the NOVA energies of the models with those of the true structures. The latter should of course always be lower. Energies were calculated after 40 ps simulated annealing simulations, models and structures were subjected to the same procedure. The results are shown in Figure 10.

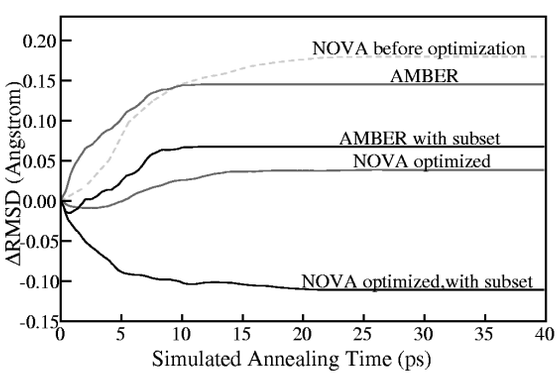



**Figure 9**. Average changes of Cα RMSDs during an extensive minimization of the model validation set. Results for the AMBER and NOVA force fields are shown. The minimization protocol was identical in both cases, only the central force field equation was changed. Gray lines correspond to the complete set (25 models), black lines indicate the subset of those models, where template resolution divided by percentage sequence identity was >0.04 (14 models). As this subset has a different average RMSD, only changes in RMSD are displayed. The performance of NOVA before optimization is also shown (dashed line).

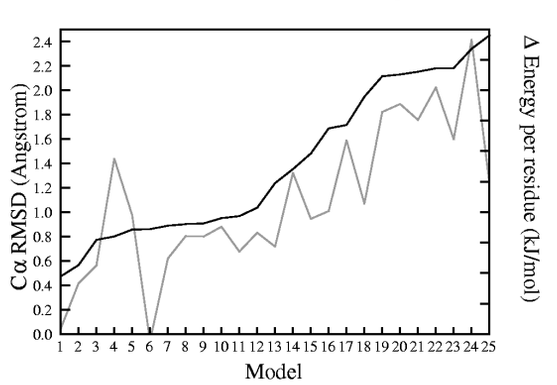



**Figure 10**. Comparison of model and structure energies. The Cα RMSDs between the 25 homology models in the validation set and their true structures are shown in black (models have been sorted by RMSD). The gray curve indicates the differences between structure and model energies, while the black curve serves as the root: as long as the true structure is lower in energy than the homology model, the gray curve stays below the black one. Every tick on the right axis corresponds to 1kJ/mol NOVA energy difference per residue.

In three of 25 cases, the real structure has a higher energy than the model. Structures 4 and 5 are not a big surprise, as Figure 8 has already shown that the force field loses its discriminative power below 0.9 Å Cα RMSD. If it were possible to "home in" from 2.4 to 0.9 Å, this would already be a huge step forward. Model 24 however looks disappointing at first sight: It has a lower energy than the X-ray structure and 2.3 Å Cα RMSD. Closer inspection reveals surprising characteristics: Hirustasin, a serine protease inhibitor, does not have an exactly determined native fold. It is a highly flexible protein[21], with a very loose residue packing (the WHAT IF packing quality Z-score[22] is -5) and almost no secondary structure (of 51 amino acids in 1BX7, 43 are neither strand nor helix, according to DSSP[23]). The only reason why this protein does not fall apart are five disulfide bonds. By assigning very similar energies to both conformations (the structure 1BX7 and the modeling template bdellastasin 1C9P-B) NOVA predicted this high level of flexibility.

## Discussion

We developed an all-atom force field that moves models on average 0.111 Å (Cα RMSD) closer to their true structures, in cases for which template resolution divided by percentage sequence identity is larger than 0.04. This result was achieved with an optimization in force field parameter space, that allowed to obtain a set of parameters that optimally fit the approximate force field equation, disregarding the physical correctness of individual values. This principle has been applied on a small scale since the very beginning of molecular dynamics simulations: The attractive $R^{-6}$ term in the Lennard Jones 12-6 potential is typically a factor 2 larger than suggested by experimental or theoretical data - which is meant to cancel the error caused by neglecting any higher order $R^{-8}$ and $R^{-10}$ terms[24]. With the Monte Carlo search method described here, it was possible to extend this idea to the entire force field.

The improvement in force field accuracy was equally large in the optimization and validation sets (Figure 7). This result is most likely due to the large optimization set with 50 proteins and no restrictions on the number of residues (the largest protein being 1C0P-A with 363 amino acids). The force field parameters thus did not "memorize" any features specific to the optimization set.

The composition of optimization and validation sets has been influenced by two arbitrary choices: first we split them into 25 structures and 25 homology models, and second we decided to use only models without insertions or deletions. The latter choice was made to assure that a signal of progress due to backbone shifts of secondary structure elements (which are the hardest to predict) was not masked by an improvement in loop modeling. To make sure that these choices did not reduce the range of application (e.g. to models without insertions or deletions), we continued the search procedure with models alone and structures alone (25 proteins each) for another 500 steps. Our remarkable finding was that the all-atom RMSDs decreased only by an insignificant amount of 0.003 Å (model

optimization set) and 0.008 Å (structure optimization set). This means that keeping a structure in its minimum and improving a model requires the same force field parameters. It also implies that the force field parameters *do not depend on the structural characteristics of the models* (the number of insertions, deletions etc.), and the algorithms used to build the models. The parameters simply provide an accurate description of protein structure.

It also became clear that the ability not to make a high resolution structure worse is a crucial feature: One of the main force field applications is the calculation and comparison of energies (Figure 10). For an all-atom force field, this only makes sense after an extensive, unrestrained energy minimization (otherwise bumps and bond lengths or angles that are slightly off, add a huge, random factor to the force field energy that makes structure comparisons impossible). If a structure is significantly distorted during the minimization, the whole procedure becomes questionable.

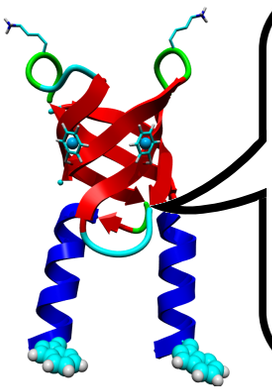
Comparing the initial and optimized values, we observed the smallest changes in experimentally and theoretically well determined parameters (bond stretching force constants changed by only 1%), while large shifts occurred in the less precisely known parameters (e.g. Van der Waals interactions).

We found that the current model improvement of 0.111 Å is limited by the simulated annealing search rather than the force field accuracy. Future work will therefore include the development of a more flexible minimization algorithm. Getting 0.111 Å closer to reality is a valuable achievement, as the best CASP predictions (http://PreictionCenter.llnl.gov) in homology modeling are often just a few hundredths of an Angstrom ahead of the competitors.

*Acknowledgements:* We would like to thank Arthur Lesk and Alexei Finkelstein for carefully reading the manuscript and providing valuable suggestions. We thank all researchers at the CMBI for participating in the Models@Home screensaver project (www.YASARA.org/models), which provided the computational resources for this work.

1. Pauling, L. & Corey, R. B. The structure of proteins: two hydrogen-bonded helical conformations of the polypeptide chain. *Proc.Natl.Acad.Sci.USA* **37**, 205-211 (1951).
2. Cornell, W. D. et al. A second generation force field for the simulation of proteins, nucleic acids and organic molecules. *J.Am.Chem.Soc.* **117**, 5179-5197 (1995).
3. MacKerell, J., A.D. et al. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *J.Phys.Chem.B* **102**, 3586-3616 (1998).
4. Van Gunsteren, W. F. et al. *Biomolecular Simulation: The GROMOS96 Manual and User Guide* (vdf Hochschulverlag, ETH Zürich, 1996).
5. Roux, B. & Simonson, T. Implicit solvent models. *Biophys.Chem.* **78**, 1-20 (1999).
6. Northrup, S. H., Pear, M. R., Lee, C. Y., McCammon, J. A. & Karplus, M. Dynamical theory of activated processes in globular proteins. *Proc.Natl.Acad.Sci.USA* **79**, 4035-4039 (1982).
7. Czaplewski, C. et al. Molecular simulation study of cooperativity in hydrophobic association. *Protein Sci.* **9**, 1235-1245 (2000).
8. Tappura, K. Influence of rotational energy barriers to the conformational search of protein loops in molecular dynamics and ranking the conformations. *Proteins* **44**, 167-179 (2001).
9. Simons, K. T., Bonneau, R., Ruczinski, I. & Baker, D. Ab initio structure prediction of CASP III targets using ROSETTA. *Proteins, Suppl.* **3**, 171-176 (1999).
10. Van Aalten, D. M. F. et al. PRODRG, a program for generating molecular topologies and unique molecular descriptors from coordinates of small molecules. *J.Comput.Aid.Mol.Des.* **10**, 255-262 (1996).
11. Levitt, M. Energy refinement of hen egg-white lysozyme. *J.Mol.Biol.* **82**, 393-420 (1974).
12. Weininger, D. SMILES, a chemical language and information system. *J.Chem.Inf.Comput.Sci* **28**, 31-36 (1993).
13. Hooft, R. W. W., Sander, C. & Vriend, G. Verification of protein structures: side-chain planarity. *J.Appl.Cryst.* **29**, 714-716 (1996).
14. Hooft, R. W. W., Sander, C. & Vriend, G. Positioning hydrogen atoms by optimizing hydrogen-bond networks in protein structures. *Proteins* **26**, 363-376 (1996).
15. Born, M. & Mayer, J. E. *Z.Phys* **75**, 1-6 (1932).
16. Mirsky, K. *Computing in Crystallography* (Delft University Press, 1978).
17. Rahman, A. & Stillinger, F. H. Improved simulation of liquid water by molecular dynamics. *J.Chem.Phys.* **60**, 1545-1557 (1974).
18. Holm, L. & Sander, C. Protein structure comparison by alignment of distance matrices. *J.Mol.Biol.* **233**, 123-138 (1993).
19. Vriend, G. WHAT IF - A molecular modeling and drug design program. *J.Mol.Graph.* **8**, 52-56 (1990).
20. Hooft, R. W. W., Sander, C., Scharf, M. & Vriend, G. The PDBFINDER database: a summary of PDB, DSSP and HSSP information with added value. *Comput.Appl.Biosci.* **12**, 525-529 (1996).
21. Uson, I. et al. The 1.2 A crystal structure of hirustasin reveals the intrinsic flexibility of a family of highly disulphide-bridged inhibitors. *Structure Fold.Des.* **7**, 55-63 (1999).
22. Vriend, G. & Sander, C. Quality control of protein models: Directional atomic contact analysis. *J.Appl.Cryst.* **26**, 47-60 (1993).
23. Kabsch, W. & Sander, C. Directory of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features. *Biopolymers* **22**, 2577-2637 (1983).
24. Stone, A. J. *The Theory of Intermolecular Forces* (Clarendon Press, Oxford., 1996).


**Space filler:** Cover of the CASP8 special issue of Proteins, showing the knowledge-based dihedral potentials of an Arginine in the YASARA force field (second next chapter).



In the main chapter 2, two approaches were described that help to build complete unit cells of protein crystals, containing all atoms that were most likely present when the X-ray diffraction experiment took place, even though they cannot really be located in the X-ray density map: pKa prediction and hydrogen-bonding network optimization. This chapter shows how to combine these high-resolution models of protein crystals with the parameter optimization procedure used for the NOVA force field to arrive at an explicit solvent force field suitable not just for energy minimization, but also for molecular dynamics simulations: YAMBER

# Making optimal use of empirical energy functions: force field parameterization in crystal space

Elmar Krieger, Tom Darden, Sander B. Nabuurs, Alexei Finkelstein and Gert Vriend

*Proteins* **57**, 678-683 (2004)

## Abstract

Today's energy functions are not able yet to distinguish reliably between correct and almost correct protein models. Improving these near-native models is currently a major bottle-neck in homology modeling or experimental structure determination at low resolution. Increasingly accurate energy functions are required to complete the 'last mile of the protein folding problem', for example during a molecular dynamics simulation.

We present a new approach to reach this goal. For 50 high resolution X-ray structures, the complete unit cell was reconstructed, including disordered water molecules, counter ions and hydrogen atoms. Simulations were then run at the pH at which the crystal was solved, while force field parameters were iteratively adjusted so that the damage done to the structures was minimal. Starting with initial parameters from the AMBER force field, the optimization procedure converged at a new force field called YAMBER (Yet Another Model Building and Energy Refinement force field), which is shown to do significantly less damage to X-ray structures, often move homology models in the right direction and occasionally make them look like experimental structures.

Application of YAMBER during the CASP5 structure prediction experiment yielded a model for target 176 that was ranked first among 150 submissions. Due to its compatibility with the well established AMBER format, YAMBER can be used by almost any molecular dynamics program. The parameters are freely available from www.YASARA.org/yamber .

## Introduction

Thanks to the exponential growth in processing power, the atomistic simulation of proteins has become feasible on personal computers, allowing scientists to routinely analyze internal motions[1,2] or the effects of point mutations on protein stability[3]. Because accurate quantum chemical calculations take orders of magnitude too long, such simulations are based on empirical energy functions, like the AMBER[4], CHARMm[5] or GROMOS[6] molecular dynamics force fields.

While many questions can be answered by today's force fields, the biannual CASP meetings regularly show that one major goal has not been reached yet: the successful refinement of homology models[7]. With structural genomics producing a rapidly growing number of templates for homology modeling[8-10], bridging the accuracy gap between homology models and high resolution X-ray structures becomes increasingly important. However, in close proximity to the native structure (0.5 - 2 Å Cα RMSD), energy functions lose their discriminative power due to the often very small structural and energetic differences involved[11]. As a conclusion, highest force field accuracy is required when refining homology models[12].

Fitting force field parameters is a very tedious task, usually involving quantum chemical calculations on small molecules[13,14]. And even perfectly accurate parameters cannot guarantee success, because the mathematical form of the energy function is an approximation by itself. Recently, we demonstrated a property we call 'force field equivalence': the force field parameters that performed best at improving homology models were virtually the same as those that did minimum damage to real structures during an energy minimization[15]. While it takes nanosecond simulations to check if a homology model improves during a molecular dynamics run, only picoseconds are required to minimize a real experimental structure. 'Force field equivalence' thus allows to gain a factor of 1000 in computing time when judging the suitability of a force field for model refinement. This in turn permits to increase force field accuracy with a rather uncompromising approach: a 'self-pa-

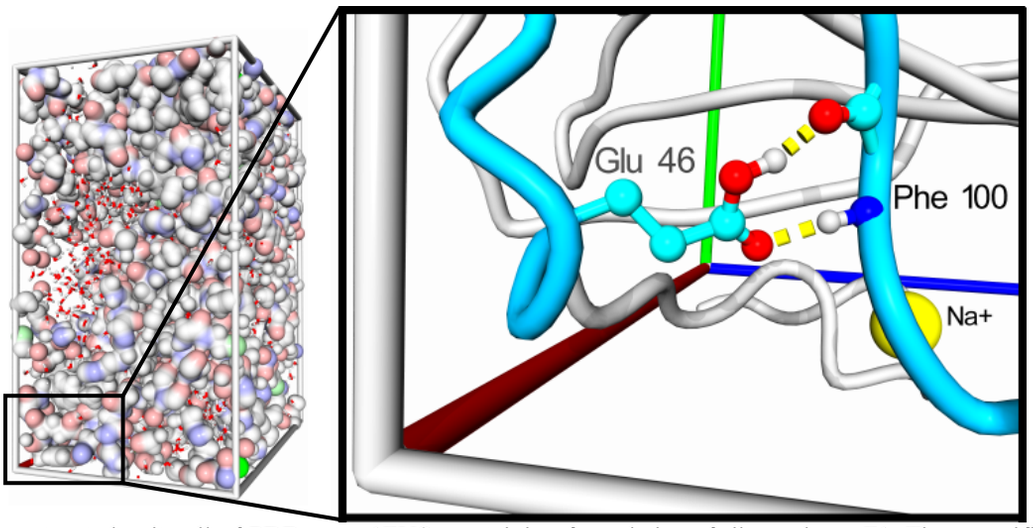



**Figure 1:** Reconstructed unit cell of PDB entry 1FUS, containing four chains of ribonuclease F1. The magnified area on the right shows residue Glu 46 contacting the backbone of Phe 100. The predicted pKa for Glu 46 is 4.1, while the protein was crystallized at pH 3.5, making a protonation very likely. A predicted sodium counter-ion is also shown. Image created with YASARA.

rameterizing force field', the parameters of which are iteratively optimized to minimize the damage done to a training set of high resolution X-ray structures during a simulation[15]. This method leads to a consistent set of parameters that optimally fit the given force field equation.

It is obvious that the protein structures in the training set should be as close to reality as possible, otherwise the force field might learn to reproduce features that simply do not exist. Here we describe a novel way of achieving that: the large-scale reconstruction of crystallographic unit cells, including disordered water molecules and counter ions, as well as hydrogen atoms. The latter turned out to be especially important, as can be seen from Figure 1. This example shows the unit cell of a ribonuclease F1 crystal (PDB[16] ID 1FUS[17]), with the side-chain carboxyl group of residue Glu 46 contacting the backbone of residue Phe 100. This close contact of one carboxyl- and one backbone oxygen is very unfavorable, unless there is a proton trapped in between. If the force field was optimized without this proton present, it would memorize a completely unrealistic interaction pattern. In addition to a hydrogen-bonding network optimizer[18], a new method for pKa prediction in protein crystals was required to assign the protonation states of ionizable groups.

Optimizing a force field in crystal space makes sure that all the forces giving rise to the experimentally observed structure are also present during the simulation and can be considered when fitting the parameters. As the physics acting in a crystal are the same as in solution, the resulting force field can be used in both environments.

# Methods

## Reconstruction of crystallographic unit cells

Using the PDBFINDER database[19], all X-ray structures without uncommon ligands and with unit cells smaller than 260000 Å$^3$ and were selected. From these, a non-redundant set (30% sequence identity cutoff) was extracted and sorted by a combined resolution/R-factor quality indicator[20]. The top 50 structures were chosen and divided into optimization (even numbers) and validation sets (uneven numbers). For every structure, WHAT IF[21] was run to add symmetry related chains based on the spacegroup information in the PDB file, and to optimize the hydrogen bonding network[18]. Then YASARA was used to predict the pKa values of the ionizable groups in the crystal, assign their protonation states based on the pH at which the crystal was solved (retrieved from literature if not specified in the PDB file) and fill the cell with water. Water molecules in the original PDB files were kept if they were closer than 5 Å to the protein. Using an iterative procedure, the AMBER99 electrostatic potential[13] was evaluated at all water molecules[22], and the one with the lowest or highest potential was turned into a sodium or chloride counter ion, respectively, until the cell was neutral. Then a startup simulation was run for 5 ps using the protocol described below, with all heavy protein atoms fixed, so that the solvent molecules could smoothly cover the protein surface. Finally a short steepest descent minimization of all atoms was done to remove severe bumps in the protein.

The 25 structures in the optimization set were 1et1, 1bqk, 1k1b, 1fus, 1ijv, 1ptf, 1g2b, 1bd8, 1h75, 1ifc, 1ajj, 1aho, 1kf3, 1jo8, 1d4t, 1hyp, 1bkr, 1a62, 1faz, 1aac, 1hcv, 2ovo, 1exr, 2erl and 1kth, the validation set consisted of 1psr, 2pth, 1ihr, 1g9o, 1fux, 1qtw, 1eqt, 1jek, 2a0b, 1gk7, 1c7k, 1d0d, 1g2r, 2ygs, 1fl0, 1hka, 1g2q, 1im5, 1gvp, 1hg7, 1i2t, 1f94, 2igd, 1cuo and 1gdu.

## Simulations of crystals and models

All simulations were run with YASARA (www.YASARA.org), using a multiple time step of 1 fs for intramolecular and 2 fs for intermolecular forces. A 7.9 Å cutoff was taken for Lennard-Jones forces and the direct space portion of the electrostatic forces, which were calculated using the Particle Mesh Ewald method[23] with a grid spacing <1 Å, 4$^{th}$ order B-splines and a tolerance of $10^{-4}$ for the direct space sum. Simulated annealing minimizations

started at 298K, velocities were scaled down with 0.9 every ten steps for a total time of five picoseconds. While it is tempting to let the unit cells relax during the energy minimizations and use the deviations as an additional force field quality indicator, we decided against it due to mainly two reasons: first, the required pressure calculations have been shown to be negatively influenced by the truncation of long-range Van der Waals interactions[23], potentially leading to optimization artifacts in our case. Second, there is no unambiguous way of combining the unit cell deviation with the atomic RMSD to arrive at a single optimization progress indicator.

Molecular dynamics simulations of crystals[24] were carried out at the temperature chosen during structure determination, the unit cells were again kept fixed (NVT ensemble). Simulations of homology models were set up in the same way as the crystal simulations (with respect to the placement of water molecules, counter ions and optimization of the hydrogen bonding network to determine e.g. Histidine protonation states), with cells 13 Å larger than the protein along each axis, and then run at 298K and constant pressure (NPT ensemble) to account for volume changes due to the much larger fluctuations of homology models in solution when compared to X-ray structures in a crystal environment. The temperature was adjusted using a Berendsen thermostat[25] based on the time-averaged temperature, i.e. to minimize the impact of temperature control, velocities were rescaled only about every ~100 simulation steps, whenever the average of the last ~100 measured temperatures converged. To allow a direct comparison, the 25 homology models were the same as in our previous study[15].

## Calculation of RMSDs and B-factors

To reduce the amount of noise in the data, various precautions were taken. In Figures 2 and 3, all atoms with alternate locations in the original PDB file were excluded from RMSD calculations. In Figure 4, the models were scanned for flexible N- and C-terminal tails (Cα atoms that have less than x other Cα atoms within 7 Å, where x=3 for the first and last residue, x=4 for the second and second last, and x=5 for all other residues). Those tails were excluded, as well as three models in the upper half of Figure 4 that could not be expected to contribute useful data: 1RB9 and 451C because they contained an iron-sulfur cluster and a hem-group for which no suitable force field parameters were available, and 1BX7, because we found previously that there was no 'correct' structure due to the protein's high flexibility[15].

B-factors were calculated from the last nanosecond of the simulation as described previously[26], using only backbone atoms and ignoring the side-chains to avoid artificially high results caused by rotamer flips. Of the 25 crystals in the validation set, the 20 structures that did not have backbone atoms with alternate locations were considered.

# Results

## Force field optimization

Initial force field parameters were borrowed from the AMBER99 force field, which has been shown earlier to be very accurate[25]. Because the total number of AMBER force field parameters is much larger than what can possibly be optimized, a subset of 37 parameters was chosen (Table I). The majority of these (20) describe Van der Waals radii and contact energies, which are usually among the most difficult to parameterize. Nine parameters capture shifts in the charge distribution, and the remaining eight relate to bonds, angles and torsions.

**YAMBER Optimization Parameters**

| Param. | Description |
|---|---|
| 1 | Common scaling factor for bond stretching force constants |
| 2 | Common scaling factor for angle bending force constants |
| 3 | Scaling factor for Lennard-Jones forces between 1-4 bonded atoms |
| 4 | Scaling factor for electrostatic forces between 1-4 bonded atoms |
| 5 | Common scaling factor for all torsion energies excl. peptide bond |
| 6 | Energy barrier of the peptide bond |
| 7 | Improper dihedral barrier for carbonyl and carboxyl groups |
| 8 | Improper dihedral barrier for all other planar groups |
| 9-24 | VdW radii of the following AMBER atom types: H, HC, H1, HP, HA, H4, H5, O, OH, C, CA, CT, N, S, C0, Zn |
| 25-28 | VdW contact energies of hydrogen, carbon, nitrogen and oxygen |
| 29 | Charge shift c for N (+c) and H (-c) in peptide bonds |
| 30 | Charge shift c for N (+c) and H (-c) in aromatic NH groups |
| 31 | Charge shift c for N (+c) and H (-c/2) in NH2 groups |
| 32 | Charge shift c for N (+c) and H (-c/3) in NH3 groups |
| 33 | Charge shift c for O (+c), H (-2c/3) and C (-c/3) in hydroxyl groups |
| 34 | Charge shift c for C (+c) and O (-c) in carbonyl groups |
| 35 | Charge shift c for C (+c) and O (-c/2) in carboxyl groups |
| 36 | Charge shift c for C (+c) and H (-c/3) in methyl groups |
| 37 | Charge shift c for C (+c) and H (-c) in aromatic rings |

**Table I**. The 37 optimization parameters of the YAMBER force field. Charge shift parameters are simply added to the AMBER charges, their distribution ensures that the overall net charge does not change.

The parameters were optimized using a Monte Carlo search algorithm[15]. The quality of each parameter set was evaluated by a simulated annealing minimization of 25 protein crystals. A lower RMSD from the initial structures meant higher quality. Due to the huge computational re-

quirements of this procedure, the Models@Home distributed computing system was used[26] (freely available from www.YASARA.org/models).

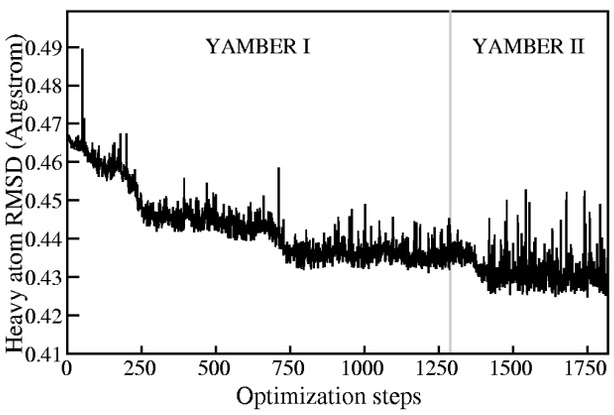



**Figure 2:** Average heavy atom RMSD of 25 protein crystals after 5 ps simulated annealing runs, as a function of force field optimization step.

The force field optimization progress is shown in Figure 2. From an initial value of 0.467 Å measured after minimization with the AMBER99 force field, the RMSD dropped quickly during the first 250 parameter optimization steps, until it hit an extended local minimum, from where it escaped around step 700. Then there was virtually no progress until step 1300, therefore we assumed convergence at 0.431 Å. This force field is called 'YAMBER 1', and except for different parameters, it is still the same as AMBER. We then decided to switch bond length and angle parameters to the Engh&Huber dataset[27], which required the introduction of 16 new atom types. This was mainly done to provide a slightly different starting point in search space and to avoid small systematic deviations from the WHAT_CHECK[28] standard values. Indeed, we found another noticeable improvement around step 1400, and finally stopped the procedure at step 1800, with an RMSD of 0.425 Å. This resulting force field is referred to as 'YAMBER 2'.

## Force field evaluation

During the force field optimization procedure, the improvement at every step is very small. Consequently, simulated annealing runs had to be used, because the randomness of a true molecular dynamics simulation at constant temperature completely masks the progress signal. To investigate how the reduction in simulated annealing RMSD affects the behavior of the two YAMBER force fields under 'real-life' conditions, we ran molecular dynamics simulations of another 25 protein crystals, sharing less than 30% sequence identity with those in the optimization set. The simulation temperatures were the same as during experimental structure determination. As can be seen from the top part of Figure 3, the Cα RMSDs from the starting structure after 1.25 nanoseconds are significantly lower for the YAMBER 1 (0.552±0.017 Å) and YAMBER 2 (0.559±0.017 Å) force fields, than for AMBER (0.657±0.018 Å), which provided the initial parameters.

The horizontal line at 0.48 Å marks the border of experimental uncertainty (i.e. the average RMSD observed if the same structure is solved at high resolution by different research groups and refinement programs[15]). As soon as the RMSD crosses this line, one can say that a structure got worse during the simulation. The optimization procedure therefore allowed to bridge ~60% of the gap between the maximum allowed RMSD and the initially measured RMSD.

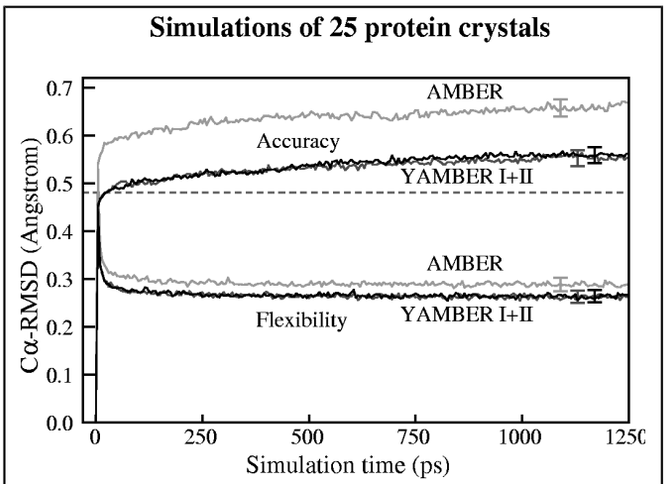



**Figure 3:** Molecular dynamics simulations of 25 protein crystals with three different force fields. Shown is the average Cα RMSD from the starting structure (upper part, indicating accuracy) and from the previous simulation snapshot 5 picoseconds before (lower part, indicating flexibility). YAMBER is shown in dark gray, YAMBER 2 in black. The horizontal line at 0.48 Å shows the limit of experimental uncertainty. Error bars were derived from the last 25% of the simulation by calculating the standard deviation for every protein and averaging over all proteins.

One could argue that the RMSD of single snapshots is not the ideal indicator of force field accuracy, because the X-ray diffraction pattern provides only a time-averaged view of the protein. Consequently, one should compare it with the time-averaged structure in the simulation. We therefore superimposed the snapshots covering the last 25% of the simulation on the X-ray structure and averaged the atom coordinates. The resulting RMSDs are 0.611 for AMBER, and 0.506/0.517 Å for YAMBER1/2, which is only a small improvement regarding the fact that the answer was implicitly given by providing the X-ray structure as a superposition target. The individual simulation snapshots are not equally spread around the true structure, but cluster at a different spot in conformational space, indicating that the source of the difference is indeed the force field accuracy.

Any parameter optimization procedure carries the inherent danger of producing artifacts. With the simulated annealing approach used here, the YAMBER force fields were trained to have stable energy minima as close to true structures as possible. An alternative view is that they were optimized not to move proteins away from the starting structure, so one might ask if they move proteins at all. This question was answered by measuring the RMSD between two consecutive simulation snapshots (saved in 5 ps

intervals), which is a good indicator of protein flexibility. The bottom part of Figure 3 shows that all three force fields lie close together at 0.288±0.014 (AMBER), 0.262±0.013 (YAMBER 1) and 0.263±0.013 Å (YAMBER 2). The overall flexibility of proteins simulated with YAMBER is thus slightly smaller, but less than two standard deviations away from AMBER, while the accuracy increased by six standard deviations. To investigate this finding in more detail, we calculated the B-factors from the simulations and compared them to the experimental values. As expected, the *in silico* B-factors are lower, because 1 ns simulations are not enough to sample conformational space exhaustively (http://amber.scripps.edu/tutorial/integrase/loop13.htm). The average B-factor is 17 for the experimental structures, 15 for AMBER and 11 for YAMBER1/2. While proteins simulated with YAMBER are thus indeed 'stiffer' on the 1 ns time-scale, they nevertheless show a better fit to the experimental data: the RMSD between calculated and experimental B-factors is 25 for AMBER, 21 for YAMBER 1 and 20 for YAMBER 2.

## Refinement of homology models

To evaluate the performance of YAMBER in model refinement, we ran simulations for a set of 25 randomly chosen homology models, some of which are very similar to the target. We previously concluded that it does not make sense to indiscriminately refine all homology models. Only the more distant models, where template (X-ray) resolution divided by percentage sequence identity with the target is larger than 0.04, are good candidates for successful refinement[15]. 11 of the 25 models match this criterion and can be simulated reliably. The difference between these two sets is shown in Figure 4. As expected, one still cannot blindly run simulations for any model built. For the complete set, the Cα-RMSD from the targets increases during the unrestrained simulation: averaged over the entire simulation time (1.5 ns), it is 0.51 (AMBER) and 0.38/0.37 Å (YAMBER1/2) higher than the initial RMSD (bottom half of Figure 4). While a temporary increase in RMSD may be required by individual models to overcome energy barriers, the fact that it happens immediately in the beginning and at a rather low starting RMSD of 1.35 Å indicates that the reason for the jump is the same as in Figure 3 - a lack of force field accuracy. The subset of distant models is shown in the upper half of Figure 4. Because it is smaller, the average RMSD is influenced more strongly by the random noise inherent to molecular dynamics simulations. Nevertheless, YAMBER 2 still performs best and actually can move models in the right direction, towards the target coordinates. After about 750 picoseconds, the YAMBER 2 curve crosses the initial line again. However, this does not mean that models generally get worse after a certain simulation time. Closer inspection shows that after 750 picoseconds, those models that can be improved reach a stable state, while the hopeless cases continue to go in the wrong direction and eventually pull the average across the line (e.g. after 750 picoseconds, five of the eleven models still have a lower RMSD (by 0.190 Å on average), and keep this improvement till the end of the simulation (0.193 Å)).

The bad models can fortunately be identified using structure validation tools. During CASP5, we used WHAT_CHECK[30] to pick out the pearls, and at least for target 176, our model was ranked first among 150 submissions (http://predictioncenter.llnl.gov/casp5/pubresult5/CASP_BROWSER/DATA.html/ 3d_T0176_all.html).

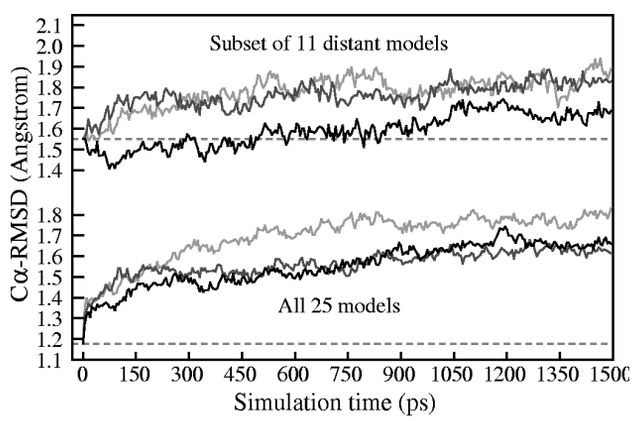



**Figure 4:** Molecular dynamics simulations of 25 homology models in solution, using three different force fields. AMBER is shown in light gray, YAMBER in dark grey and YAMBER 2 in black. The average Cα-RMSD from the target is shown on the vertical axis, the dotted lines indicates the initial models before any refinement. The upper half shows a subset of 14 distant models, where template resolution divided by percentage sequence identity to the target is >0.04.

# Discussion and Conclusion

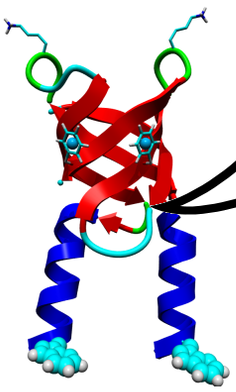
When looking at the final force field parameters, one obvious question is: which parameters changed and why? In a best case scenario, all parameter changes would seem random, indicating the absence of systematic errors in the initial force field as well as in the optimization procedure. And essentially this was the case: Van der Waals radii shifted up in six and down in eight cases, charges increased in five and decreased in three cases. Two systematic changes were noted however: the scaling factors for the Van der Waals and electrostatic forces between 1-4 bonded atoms shifted both down considerably (from 0.5 and 0.83 to 0.27 and 0.65, respectively), and the Van der Waals contact energies increased in all cases, almost doubling for hydrogen and carbon atoms. In the AMBER force field, hydrogen and carbon have a very small contact energy (0.015 and 0.09-0.11 kcal/mol) when compared to nitrogen (0.17) and oxygen (0.21). In YAMBER 2, carbon and nitrogen came out equal (0.19 kcal/mol). This increase in Van der Waals attraction may explain the slightly lower flexibility of proteins simulated with the YAMBER force fields (Figure 3). An additional reason may be the backbone hydrogens bonds, which got stronger during the parameter optimization (in contrast to other hydrogen bonds involving side-chain hydroxyl and amino groups that got weaker).

The fact that YAMBER 2 performed best not only in its trained area, the minimization of protein crystals, but also in the quite different application of homology model refinement, reaffirms our previous finding that there is only one optimum set of force field parameters. Therefore it seems likely that the YAMBER force fields will be more generally applicable.

In our previous work, the top improvement we found for any model was 0.25 Å[15]. We concluded that the problem was not the force field accuracy, but just the fact that the models 'got stuck' too early during the simulated annealing minimization. Here, we used molecular dynamics simulations to search conformational space, which are much less likely to get trapped in local minima, thereby raising our hopes for a significantly better result. An analysis of all 75 model trajectories yielded quite a surprise: the highest improvement was found for a protein G model, simulated with YAMBER 2. The Cα-RMSD dropped from an initial value of 1.74 Å, which is typical for a close homology model, all the way down to 0.7 Å, which corresponds to a medium-resolution X-ray structure. So at least in this case, we could observe the metamorphosis of a model to an experimental-like structure.

*Acknowledgements:* We would like to thank Herman Berendsen and Alan Mark for their helpful hints, and D.N. Ivankov, M.Yu. Lobanov, I. Litvinov, N.S. Bogatyreva, O.V. Galzitskaya, M.S. Kondratova, S.O. Garbuzinskii, M.A. Roytberg, who were team members at CASP-5. We also thank all colleagues at the CMBI for using the Models@Home screensaver, which provided the computational resources. This work was supported in part by the European community (5th Framework program NMRQUAL Contract Number QLG2-CT-2000-01313), by NWO grant 047.009.021, by the program "Physical and Chemical Biology" of the Russian Academy of Sciences and by an International Research Scholar's Award to A.V.F. from the Howard Hughes Medical Institute.

1. Karplus M, McCammon JA. Molecular dynamics simulations of biomolecules. NatStructBiol 2002;9:646-652.
2. de Groot BL, van Aalten DM, Scheek RM, Amadei A, Vriend G, Berendsen HJ. Prediction of protein conformational freedom from distance constraints. Proteins 1997;29:240-251.
3. el-Bastawissy E, Knaggs MH, Gilbert IH. Molecular dynamics simulations of wild-type and point mutation human prion protein at normal and elevated temperature. JMolGraphModel 2001;20:145-154.
4. Cornell WD, Cieplak P, Bayly CI, Gould IR, Merz KM, Jr., Ferguson DM, Spellmeyer DC, Fox T, Caldwell JW, Kollman PA. A second generation force field for the simulation of proteins, nucleic acids and organic molecules. JAmChemSoc 1995;117:5179-5197.
5. MacKerell J, A.D., Bashford D, Bellott M, Dunbrack RL, Evanseck JD, Field MJ, Fischer S, Gao J, Guo H, Ha S, Joseph-McCarthy D, Kuchnir L, Kuczera K, Lau FTK, Mattos C, Michnick S, Ngo T, Nguyen DT, Prodhom B, Reiher WE, Roux B, Schlenkrich M, Smith JC, Stote R, Straub J, Watanabe M, Wiorkiewicz-Kuczera J, Yin D, Karplus M. All-atom empirical potential for molecular modeling and dynamics studies of proteins. JPhysChemB 1998;102:3586-3616.
6. Van Gunsteren WF, Billeter SR, Eising AA, Hünenberger PH, Krüger P, Mark AE, Scott WRP, Tironi IG. Biomolecular Simulation: The GROMOS96 Manual and User Guide: vdf Hochschulverlag, ETH Zürich; 1996.
7. Tramontano A, Leplae R, Morea V. Analysis and assessment of comparative modeling predictions in CASP4. Proteins 2001;S5:22-38.
8. Sanchez R, Sali A. ModBase: a database of comparative protein structure models. Bioinformatics 1999;15:1060-1061.
9. Peitsch MC, Schwede T, Guex N. Automated protein modelling - the proteome in 3D. Pharmacogenomics 2000;1:257-266.
10. Teichmann SA, Murzin AG, Chothia C. Determination of protein function, evolution and interactions by structural genomics. CurrOpinStructBiol 2001;11:354-363.
11. Huang ES, Subbiah S, Tsai J, Levitt M. Using a hydrophobic contact potential to evaluate native and near-native folds generated by molecular dynamics simulations. JMolBiol 1996;257:716-725.
12. Lee MR, Tsai J, Baker D, Kollman PA. Molecular dynamics in the endgame of protein structure prediction. JMolBiol 2001;313:417-430.
13. Wang J, Cieplak P, Kollman PA. How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules? JCompChem 2000;21:1049-1074.
14. Ren P, Ponder JW. Consistent treatment of inter- and intramolecular polarization in molecular mechanics calculations. JCompChem 2002;23:1497-1506.
15. Krieger E, Koraimann G, Vriend G. Increasing the precision of comparative models with YASARA NOVA - a self-parameterizing force field. Proteins 2002;47:393-402.
16. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. Nucleic Acids Res 2000;28:235-242.
17. Vassylyev DG, Katayanagi K, Ishikawa K, Tsujimoto-Hirano M, Danno M, Pahler A, Matsumoto O, Matsushima M, Yoshida H, Morikawa K. Crystal structures of ribonuclease F1 of Fusarium moniliforme in its free form and in complex with 2'GMP. JMolBiol 1993;230:979-996.
18. Hooft RWW, Sander C, Vriend G. Positioning hydrogen atoms by optimizing hydrogen-bond networks in protein structures. Proteins 1996;26:363-376.
19. Hooft RWW, Sander C, Scharf M, Vriend G. The PDBFINDER database: a summary of PDB, DSSP and HSSP information with added value. ComputApplBiosci 1996;12:525-529.
20. Hooft RWW, Sander C, Vriend G. Verification of protein structures: side-chain planarity. JApplCryst 1996;29:714-716.
21. Vriend G. WHAT IF - A molecular modeling and drug design program. JMolGraph 1990;8:52-56.
22. Walser R, Hunenberger PH, Van Gunsteren WF. Comparison of different schemes to treat long-range electrostatic interactions in molecular dynamics simulations of a protein crystal. Proteins 2001;43:509-519.
23. Essman U, Perera L, Berkowitz ML, Darden T, Lee H, Pedersen LG. A smooth particle mesh Ewald method. JChemPhys 1995;103:8577-8593.
24. Van Gunsteren WF, Karplus M. Protein dynamics in solution and in a crystalline environment: a molecular dynamics study. Biochemistry 1982;21:2259-2274.
25. Berendsen HJC, Postma JPM, van Gunsteren WF, DiNola A, Haak JR. Molecular dynamics with coupling to an external bath. JChemPhys 1984;81:3684-3690.
26. Eastman P, Pellegrini M, Doniach S. Protein flexibility in solution and in crystals. JChemPhys 1999;110:10141-10152.
27. Liu H, Elstner M, Kaxiras E, Frauenheim T, Hermans J, Yang W. Quantum mechanics simulation of protein dynamics on long timescale. Proteins 2001;44:484-489.
28. Krieger E, Vriend G. Models@Home: distributed computing in bioinformatics using a screensaver based approach. Bioinformatics 2002;18:315-318.
29. Engh RA, Huber R. Accurate bond and angle parameters for X-ray protein structure refinement. Acta CrystA 1991;47:392-400.
30. Hooft RWW, Vriend G, Sander C, Abola EE. Errors in protein structures. Nature 1996;381:272-272.



Throughout the history of protein structure prediction, one approach always proved the most successful: extracting general knowledge about proteins from the countless structures deposited in the PDB. Even energy functions can be obtained this way – so called "knowledge-based potentials". This chapter describes how such knowledge-based dihedral angle potentials were added to the YAMBER force field (see previous chapter) to arrive at the YASARA force field. The original article from the CASP8 special issue of the journal *Proteins* contains descriptions of three other refinement methods that are not related to YASARA and have been omitted to save space.

# Improving physical realism, stereochemistry and side-chain accuracy in homology modeling: four approaches that performed well in CASP8

Elmar Krieger[1], Keehyoung Joo[2], Jinwoo Lee[3], Jooyoung Lee[2], Srivatsan Raman[4], James Thompson[4], Mike Tyka[4], David Baker[4], and Kevin Karplus[5]

*Proteins 77 Suppl* **9**, 114-122 (2009)

*1 Centre for Molecular and Biomolecular Informatics, Radboud University Nijmegen Medical Centre, the Netherlands*
*2 School of Computational Sciences, Korea Institute for Advanced Study, Seoul 130-722, Korea*
*3 Department of Mathematics, Kwangwoon University, Seoul 139-701, Korea*
*4 Department of Biochemistry, University of Washington, Seattle, U.S.A.*
*5 Biomolecular Engineering, University of California, Santa Cruz, U.S.A.*

## Abstract

A correct alignment is an essential requirement in homology modeling. Yet in order to bridge the structural gap between template and target, which may not only involve loop rearrangements, but also shifts of secondary structure elements and repacking of core residues, high-resolution refinement methods with full atomic details are needed. Here we describe four approaches that address this '*last mile of the protein folding problem*' and have performed well during CASP8, yielding physically realistic models:

YASARA, which runs molecular dynamics simulations of models in explicit solvent, using a new partly knowledge-based all atom force field derived from Amber, whose parameters have been optimized to minimize the damage done to protein crystal structures.

The LEE-SERVER, which makes extensive use of conformational space annealing to create alignments, to help Modeller build physically realistic models while satisfying input restraints from templates and CHARMM stereochemistry, and to remodel the side-chains.

ROSETTA, whose high resolution refinement protocol combines a physically realistic all atom force field with Monte Carlo minimization to allow the large conformational space to be sampled quickly.

And finally UNDERTAKER, which creates a pool of candidate models from various templates and then optimizes them with an adaptive genetic algorithm, using a primarily empirical cost function that does not include bond angle, bond length, or other physics-like terms.

## Introduction

The four groups of this paper were selected because their template-based predictions at CASP8 scored especially well on two aspects: First, they showed a good match to the target H-bonds and target side-chain positions and rotamers[1]. Second, they were physically realistic, with few all-atom clashes and good bond lengths, bond angles and Ramachandran plot. While the latter features are not among the most important ones in structure prediction, getting them right is both essential and difficult: essential because the fine-grained energy functions for high-resolution refinement depend on correct stereochemistry, and difficult because a simple energy minimization not only improves the look of the model, but also tends to move it away from the target[2].

The authors' prediction methods use distinctly different ways to obtain these results, and this article is an attempt to synthesize some commonality from these different approaches. There are two basic approaches to getting the details right: correct by construction and optimizing cost functions (also called energy functions).

The correct-by-construction approach makes sure that all parameters being considered are set correctly in initial models, and that conformation-change operators do not change these parameters. For example, in Rosetta's initial stages, all backbone bond angles and bond lengths are set to ideal values, and only torsion angles are modified. In undertaker, all backbone fragments and side-chains are copied from PDB files, and bond angles and bond lengths are not changed by conformation-change operators.

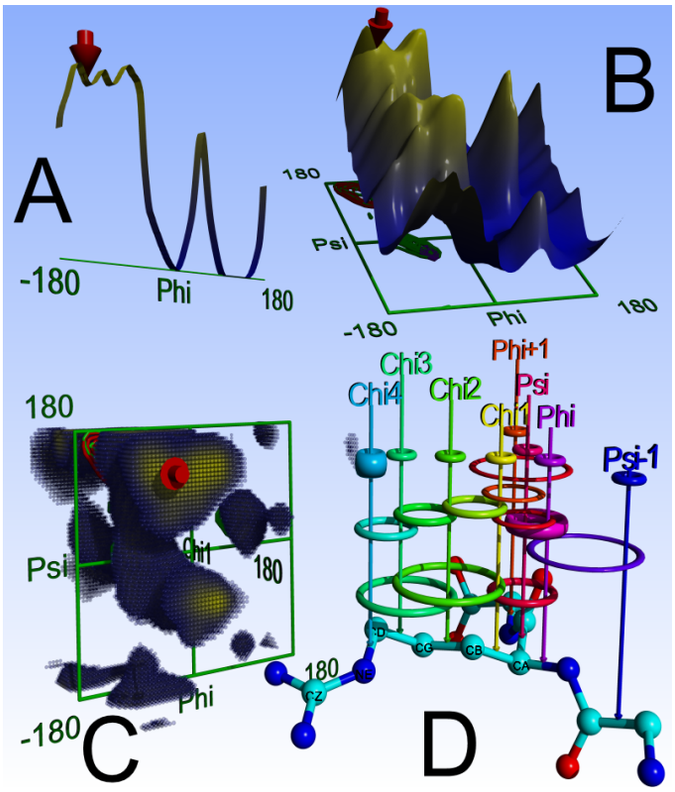



**Figure 1A:** The 1D knowledge-based potential of the backbone Phi dihedral of Arg. To aid visualization, the energy axis is flipped: favorable low energy regions are at the top and colored yellow, high energy regions are colored blue. The red arrow indicates the current conformation shown in 'D'. **B:** The 2D PhiPsi potential, most famous for its relationship with the Ramachandran plot. **C:** The 3D PhiPsiChi1 potential, the highest energy regions are transparent for clarity. **D:** Distribution of knowledge-based potentials, using the most complicated case of four Chi dihedrals as an example. A residue like Arg (or Lys) has seven 1D potentials (smallest rings at the top: Omega (not shown), Phi, Psi, Chi1-4), seven 2D potentials (rings in the middle: Psi-1Phi, PhiPsi, PhiChi1, PsiChi1, Chi12, Chi23, Chi34) and three 3D potentials (at the bottom: PhiPsiChi1, Chi123, Chi234). Note that the thickness of the rings symbolizes the weight: The 1D Chi4 potential contributes more because Chi4 is only covered by one 2D and one 3D potential. Likewise, the 2D PhiPsi potential (magenta) accounts for the fact that Phi and Psi are covered by just one 3D potential. The 2D PhiChi1 and PsiChi1 potentials count only half to ensure that Chi1 is not overweighted. Finally, the Psi-1Phi potential, which spans two subsequent residues, fills up the remaining gaps. Since this potential is special (the two dihedrals are not adjacent but separated by Omega), its weight was optimized independently (parameter 41 in Table 1). Graphics created with YASARA and PovRay.

The cost function approach requires attention to two issues: the accuracy of the function and the efficiency of the conformational search. Many terms are included in the cost function, and these terms often are in conflict, so setting the weights of the different components is important when optimizing the weighted sum. Rosetta and YASARA have put considerable effort in optimizing their energy functions so that low-energy protein models are much more likely to be correct predictions, and experimental crystal structures have very low energy. Rosetta has also put effort into improving their Monte-Carlo-based search strategy to sample conformation space sufficiently. The Lee method focuses on improved conformational search by conformational space annealing[3], using the standard Modeller[4] cost function, which includes competing constraints from "many" templates and CHARMM[5].

One problem with the cost-function approach is that some terms of the cost function (such as the Lennard-Jones potential for clash detection) are much stiffer than other terms (such as bond length and bond angle), so that poor bond lengths and bond angles are accepted to remove clashes, rather than doing more difficult combinatorial searches that remove the clashes without damaging the bond angles and bond lengths. One solution (used in YASARA and Rosetta) is to keep bond lengths and angles fixed until the worst clashes are gone. Another solution is to perform straightforward (but difficult) global optimization of the function considering all degrees of freedom as done by the LEE server. Undertaker ramps up the weight of the clash terms as the optimization progresses.

All four methods extract considerable information from the templates, using them to provide initial starting models and, for the LEE server and undertaker, constraints for the cost function. None of the four methods use the old "frozen-core" approach, in which portions of the backbone copied from the templates are not allowed to change.

The next sections will describe in more detail the approaches of each of the four methods.

## The self-parameterizing knowledge-based YASARA force field

Improving the physical correctness of protein models looks like the ideal task for a widely used physics-based method: all-atom molecular dynamics simulation with explicit solvent. Still, this approach has traditionally had a difficult stand at CASP: computers are too slow to simulate the folding of the CASP targets from a random initial conformation, and empirical force fields are usually too crude to really improve models built with other methods. The latter problem is related to the observation that high resolution X-ray structures 'jump away' during the first picoseconds of a simulation, accompanied by a deterioration of knowledge-based indicators like Ramachandran plot quality.

The NMR community, being haunted by the poor quality of structures obtained from molecular dynamics refinement (Ramachandran plot Z-scores of -7 were common[6]), soon came up with a solution: The force field was augmented with knowledge-based torsional potentials, that were extracted from high-resolution X-ray structures and ensured that the resulting models looked the same[7].

Interestingly, the models often looked even better than X-ray structures, raising the question whether these knowledge-based potentials really improved the accuracy, or just created artificially good-looking models.

The YASARA force field described here addresses these issues by combining the AMBER all-atom force field equation[8] with multi-dimensional knowledge-based tor-

sional potentials[1] (Figure 1) and with a consistent set of force field parameters to maximize the accuracy: this is achieved by making a random change to one or more parameters (e.g. a certain van der Waals radius, a charge, or the weight of a knowledge-based potential, see Table 1), energy-minimizing a training set of 25 high-resolution X-ray structures, measuring the damage done, and rejecting or accepting the new force field based on a Monte Carlo criterion[9]. To ensure that all forces responsible for the experimentally observed structure are considered, minimizations are done in crystal space, using complete unit cells[2]. As a result, one obtains a force field that has stable energy minima as close as possible to native structures. And as shown before, this is essentially equivalent to a force field that moves models closer to native structures during a simulation[9].

| *Par.* | *Description* |
|---|---|
| 38 | Scaling factor for those (Y)AMBER torsional potentials that are also covered by knowledge-based potentials (KBPs). (There are no KBPs involving terminal hydrogens and inside rings). |
| 39 | Height of the average 1D KBP energy barrier. |
| 40 | Height of the average 2D KBP energy barrier. |
| 41 | Ratio of 2D PhiPsi and $Psi^{-1}Phi$ KBPs |
| 42 | Height of the average 3D KBP energy barrier. |

**Table 1:** The optimized parameters 38 to 42 of the YASARA force field. The first 37 parameters involve bonds, angles, torsions, VdW radii and point charges, and have been described previously[2]. 'Height of the energy barrier' is a synonym for 'weight of the potential in the force field', i.e. parameter optimization was used to determine the optimal weights of 1D, 2D and 3D potentials.

The parameter optimization procedure is computationally intensive and took about half a year using the Models@Home distributed computing system[10]. After convergence, the contributions of 1D : 2D : 3D potentials (Table 1) were 2.6 : 0.33 : 3.82 kcal/mol. So the highest weight was assigned to the 3D potentials, which makes sense since they contain the most information. A simple explanation for the surprising result that 2D potentials came out last could be that they are (except for $Psi^{-1}Phi$) fully contained in the 3D potentials, whereas the 1D potentials have a higher resolution (256 instead of 64 bins, see footnote).

The cross-validated results are shown in Table 2. Obviously, the knowledge-based potentials helped a lot: First, the damage done to crystal structures during an energy minimization (RMSD column) is noticeably smaller with the YASARA force field than with YAMBER2 (which used the same parameter optimization approach, but without knowledge-based potentials[2]) or AMBER. While these RMSD differences look small, they translate to much larger differences during longer simulations[2]. And second, the old modeler's rule of thumb to '*never hurt a protein by energy minimization if it can be avoided*' is no longer an issue: the deterioration of structure validation Z-scores is gone, the minimized structures even look a bit better according to WHAT IF (third column in Table 2). This does not only hold for those checks that are related to the knowledge-based potentials described here (Ramachandran plot, backbone conformation quality), but also for the independent 3D packing quality check (which improves from -0.583 to -0.539).

| *Force field* | *RMSD* | *Quality Z-Score* |
|---|---|---|
| None | 0.000 | 0.348 |
| AMBER99 ([17]) | 0.440 | -0.581 |
| AMBER03 ([18]) | 0.437 | -0.364 |
| YAMBER2 ([2]) | 0.410 | -0.353 |
| YASARA | 0.379 | 0.616 |

**Table 2:** Simulated annealing minimization of protein crystals using different force fields and a protocol described previously[2], the average results for an independent validation set of another 25 proteins are shown. **RMSD** is the heavy-atom RMSD from the X-ray structure after the minimization converged, **Quality Z-Score** is the average of the three most sensitive WHAT IF checks[19]: Ramachandran plot (RAMCHK), backbone conformation (BBCCHK) and 3D packing quality (QUACHK). A Z-score is simply the number of standard deviations away from the average, a negative value is assigned to models that are worse than the average high resolution X-ray structure (e.g. have more Ramachandran plot outliers).

Regarding the practical application of the new YASARA force field during CASP8, two results are noteworthy: First, extensive parallel molecular dynamics simulations[2] aided by Concoord[11] won three of the 12 refinement targets (TR429, TR454 and TR469, based on GDT_TS scores for Model_1). And second, short energy minimizations with a solvent shell helped to improve the physical realism of homology models[1] built for the main CASP8 targets. The initial models and thus the starting points for the energy minimizations were obtained using the following protocol: PSI-BLAST[12] was run to identify the five closest templates in the PDB, then for each template up to five stochastic alignments were created[13] using SSALIGN scoring matrices[14]. For each of the maximally 25 template/alignment combinations, a 3D model was built using loop conformations extracted from the PDB[15] and the SCWRL side-chain placement algorithm[16]. After the minimization, the models were ranked by quality Z-score (Ta-

1 Knowledge-based torsion potentials have been extracted from ~11000 non-redundant PDB files (90% sequence identity cutoff, resolution better than 2.5 Å) using an approach described previously[6], with some modifications: to avoid secondary structure bias, only residues outside helices and sheets with an average B-factor<40 were considered, 256 bins were used for 1D potentials, 64*64 bins for 2D potentials and 64*64*64 bins for 3D potentials.

2 The refinement protocol consisted of 100 MD simulations in explicit solvent (each lasting about 10ps), run in parallel using Models@Home. Then the best model was picked considering YASARA force field energies and WHAT IF validation Z-scores, and subjected to another refinement cycle until the procedure converged. During the first refinement cycles, Concoord was tried as well to quickly sample conformational space just before each MD simulation. The latter then usually managed to restore the model quality scores after the difficult Concoord journey through distance space.

ble 2), and the top five were submitted.

The YASARA force field and the homology modeling protocol have been implemented as part of the molecular modeling program YASARA Structure, available from www.YASARA.org. A web-server can be found at www.YASARA.org/minimizationserver.

*Acknowledgments:* Many thanks to the users of YASARA (Yet Another Scientific Artificial Reality Application) for financing this work, and to Bert de Groot , Robbie Joosten and Gert Vriend for their support.

**Here the original article contains three more subsections: Protein 3D modeling by global optimization (Jooyoung Lee *et al.*), Rosetta all-atom refinement (David Baker *et al.*) and Undertaker keeps the good parts (Kevin Karplus).**

## Conclusion

In the early days of CASP, the best homology models were built using the 'frozen core' approach, keeping the backbone of aligned residues fixed. Optimists who allowed all atoms to move during the refinement usually had to pay the price of reduced accuracy. Unfortunately the 'frozen core' approach cannot produce models more accurate than the best available template, so it is good news that all of the four methods described here give the atoms the freedom they deserve.

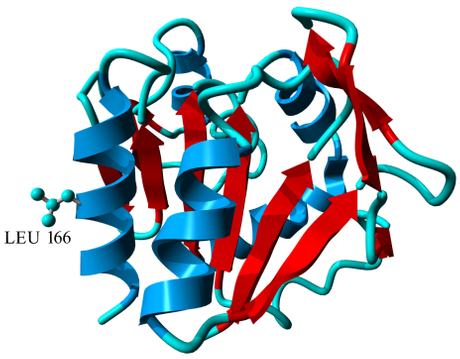
Cost functions have become accurate enough that they often move models in the right direction during an optimization, particularly when the initial homology model is close to the experimental structure.

One approach to success (Rosetta, undertaker, and YASARA) is multi-level optimization: first address the course-grained features, clipping clash costs and tweaking mainly torsion angles rather than bond lengths and bond angles, so that the latter will not absorb surrounding errors. Fine tuning involving all parameters is done only once the major gaps and clashes have been resolved. An alternative approach is to apply a single straightforward (but difficult) energy optimization to an all-atom cost function as in Modeller-CSA.

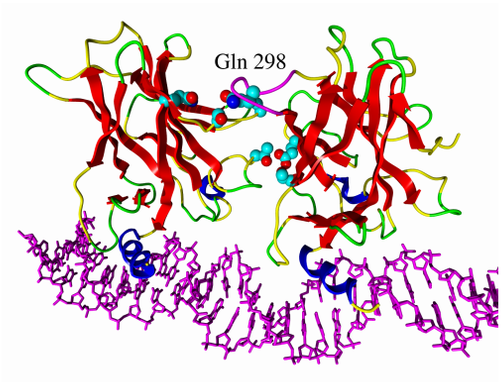
All four methods are tied to the template structures and will thus not yield very different solutions: Modeller-CSA and undertaker try to satisfy restraints extracted from the templates, while Rosetta and YASARA don't use restraints, but will also not move too far away, since the refinement simulation time is too short, or the energy barriers are too high. The differences in accuracy of the models produced by the four methods are probably due mainly to the differences in the initial alignments to templates used by the methods.

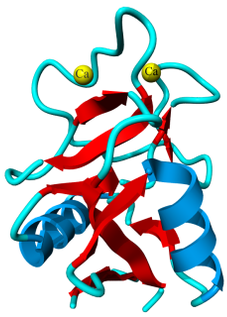
The above also implies that the alignment problem has not become obsolete yet. Incorporating structural information from single or multiple templates through accurate alignments is still a very important part of structure prediction. Three-dimensional modeling of apparently unaligned segments is still a challenging problem, though de novo modeling with proper loop closure is showing some promise.

1. Keedy, D. A., Williams, C. J., Headd, J. J., Arendall III, W. B., Chen, V. B., Kapral, G. J., Gillespie, R., Zemla, A., Richardson, D. C. & Richardson, J. S. The other 90% of the protein: Assessment beyond the Calphas for CASP8 template-based models. *Proteins* **77 Suppl.9** (2009).
2. Krieger, E., Darden, T., Nabuurs, S. B., Finkelstein, A. & Vriend, G. Making optimal use of empirical energy functions: force field parameterization in crystal space. *Proteins* **57**, 678-683 (2004).
3. Lee, J., Scheraga, H. A. & Rackovsky, S. New Optimization Method for Conformational Energy Calculations on Polypeptides: Conformational Space Annealing. *J.Comput.Chem.* **18**, 1222-1232 (1997).
4. Sali, A. & Blundell, T. L. Comparative protein modelling by satisfaction of spatial restraints. *J.Mol.Biol.* **234**, 779-815 (1993).
5. MacKerell, J., A.D., Bashford, D., Bellott, M., Dunbrack, R. L., Evanseck, J. D., Field, M. J., Fischer, S., Gao, J., Guo, H., Ha, S., Joseph-McCarthy, D., Kuchnir, L., Kuczera, K., Lau, F. T. K., Mattos, C., Michnick, S., Ngo, T., Nguyen, D. T., Prodhom, B., Reiher, W. E., Roux, B., Schlenkrich, M., Smith, J. C., Stote, R., Straub, J., Watanabe, M., Wiorkiewicz-Kuczera, J., Yin, D. & Karplus, M. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *J.Phys.Chem.B* **102**, 3586-3616 (1998).
6. Nabuurs, S. B., Nederveen, A. J., Vranken, W., Doreleijers, J. F., Bonvin, A. M., Vuister, G. W., Vriend, G. & Spronk, C. A. DRESS: a database of refined solution NMR structures. *Proteins* **55**, 483-486 (2004).
7. Kuszewski, J., Gronenborn, A. M. & Clore, G. M. Improvements and extensions in the conformational database potential for the refinement of NMR and X-ray structures of proteins and nucleic acids. *J Magn Reson* **125**, 171-177 (1997).
8. Cornell, W. D., Cieplak, P., Bayly, C. I., Gould, I. R., Merz, K. M., Jr., Ferguson, D. M., Spellmeyer, D. C., Fox, T., Caldwell, J. W. & Kollman, P. A. A second generation force field for the simulation of proteins, nucleic acids and organic molecules. *J.Am.Chem.Soc.* **117**, 5179-5197 (1995).
9. Krieger, E., Koraimann, G. & Vriend, G. Increasing the precision of comparative models with YASARA NOVA - a self-parameterizing force field. *Proteins* **47**, 393-402 (2002).
10. Krieger, E. & Vriend, G. Models@Home: distributed computing in bioinformatics using a screensaver based approach. *Bioinformatics* **18**, 315-318 (2002).
11. de Groot, B. L., van Aalten, D. M., Scheek, R. M., Amadei, A., Vriend, G. & Berendsen, H. J. Prediction of protein conformational freedom from distance constraints. *Proteins* **29**, 240-251 (1997).
12. Altschul, S. F., Madden, T. L., Schaeffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389-3402 (1997).
13. Mueckstein, U., Hofacker, I. L. & Stadler, P. F. Stochastic pairwise alignments. *Bioinformatics* **18**, Suppl.2, 153-160 (2002).
14. Qiu, J. & Elber, R. SSALN: An alignment algorithm using structure-dependent substitution matrices and gap penalties learned from structurally aligned protein pairs. *Proteins* **62**, 881-891 (2006).
15. Canutescu, A. A. & Dunbrack, R. L. J. Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein Sci.* **12**, 963-972 (2003).
16. Canutescu, A. A., Shelenkov, A. A. & Dunbrack, R. L. J. A graph-theory algorithm for rapid protein side-chain prediction. *Protein Sci.* **12**, 2001-2014 (2003).
17. Wang, J., Cieplak, P. & Kollman, P. A. How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules? *J.Comp.Chem.* **21**, 1049-1074 (2000).
18. Duan, Y., Wu, C., Chowdhury, S., Lee, M. C., Xiong, G., Zhang, W., Yang, R., Cieplak, P., Luo, R. & Lee, T. A Point-Charge Force Field for Molecular Mechanics Simulations of Proteins. *J.Comp.Chem.* **24**, 1999-2012 (2003).
19. Hooft, R. W. W., Vriend, G., Sander, C. & Abola, E. E. Errors in protein structures. *Nature* **381**, 272-272 (1996).



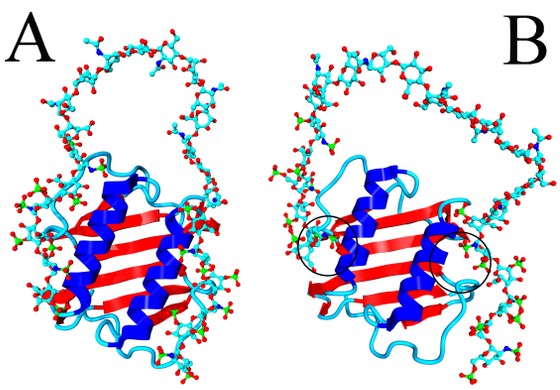


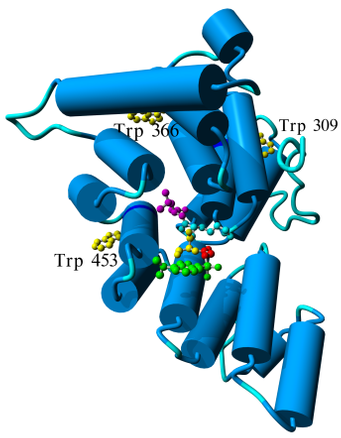


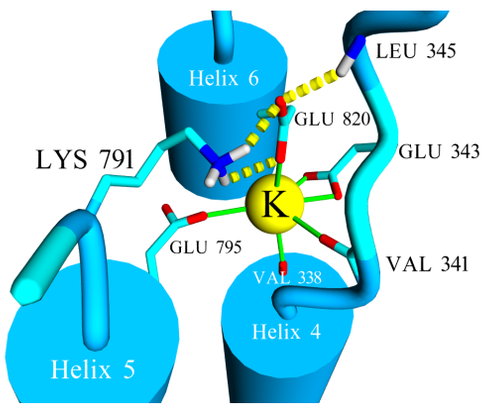



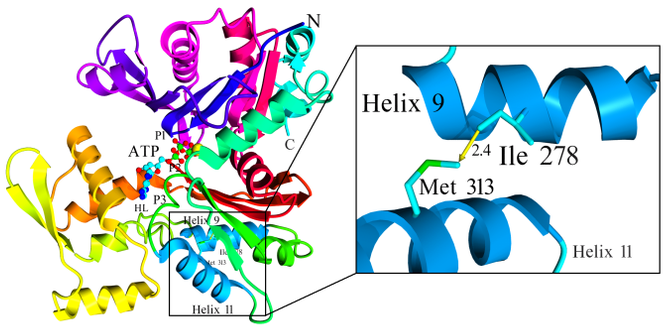
# The applications

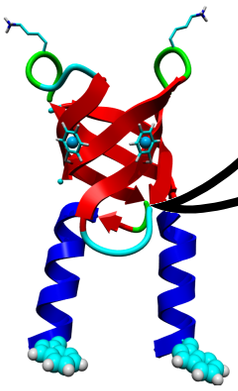
4





In collaboration with Vincenzo Bonifati and Peter Heutink at the Department of Clinical Genetics, Erasmus Medical Center Rotterdam, we modeled DJ-1 (shown above). Vincenzo located this protein while searching for the chromosomal location of an inheritable form of early-onset parkinsonism. This protein of still unknown function was found to contain a Leu > Pro mutation at position 166, which according to the homology model is in the middle of a C-terminal alpha-helix. As proline cannot participate in helical hydrogen bonding, it is a strong helix breaker and consequently destabilizes the protein. It could be shown experimentally that the L166P mutant is degraded much more rapidly than the wild-type DJ-1 protein.

# Mutations in the DJ-1 gene associated with autosomal recessive early-onset parkinsonism

Bonifati V, Rizzu P, van Baren MJ, Schaap O, Breedveld GJ, Krieger E, Dekker MC, Squitieri F, Ibanez P, Joosse M, van Dongen JW, Vanacore N, van Swieten JC, Brice A, Meco G,van Duijn CM, Oostra BA, Heutink P

*Science* **299**, 256-9 (2003)

*Department of Clinical Genetics, Erasmus Medical Center Rotterdam, The Netherlands*

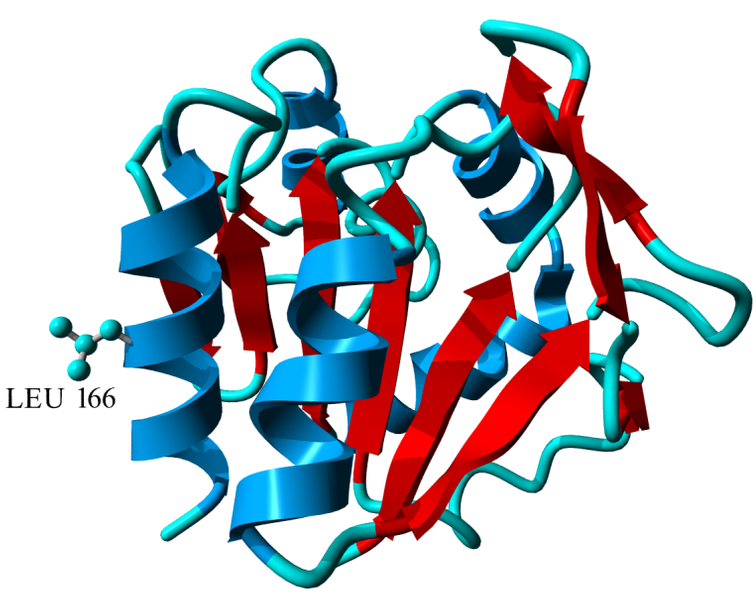
The DJ-1 gene encodes a ubiquitous, highly conserved protein. Here, we show that DJ-1 mutations are associated with PARK7, a monogenic form of human parkinsonism. The function of the DJ-1 protein remains unknown, but evidence suggests its involvement in the oxidative stress response. Our findings indicate that loss of DJ-1 function leads to neurodegeneration. Elucidating the physiological role of DJ-1 protein may promote understanding of the mechanisms of brain neuronal maintenance and pathogenesis of Parkinson's disease.

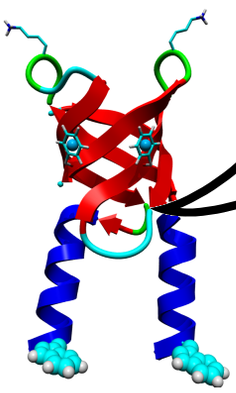
# DJ-1, a novel gene for autosomal recessive, early onset parkinsonism

Bonifati V, Rizzu P, Squitieri F, Krieger E, Vanacore N, van Swieten JC, Brice A, van Duijn CM, Oostra B, Meco G, Heutink P

*Neurol Sci.* **24**, 159-60 (2003)

Four chromosomal loci ( PARK2, PARK6, PARK7, and PARK9) associated with autosomal recessive, early onset parkinsonism are known. We mapped the PARK7 locus to chromosome 1p36 in a large family from a genetically isolated population in the Netherlands, and confirmed this linkage in an Italian family. By positional cloning within the refined PARK7 critical region we recently identified mutations in the DJ-1 gene in the two PARK7-linked families. The function of DJ-1 remains largely unknown, but evidence from genetic studies on the yeast DJ-1 homologue, and biochemical studies in murine and human cell lines, suggests a role for DJ-1 as an antioxidant and/or a molecular chaperone. Elucidating the role of DJ-1 will lead to a better understanding of the pathogenesis of DJ-1-related and common forms of Parkinson's disease.





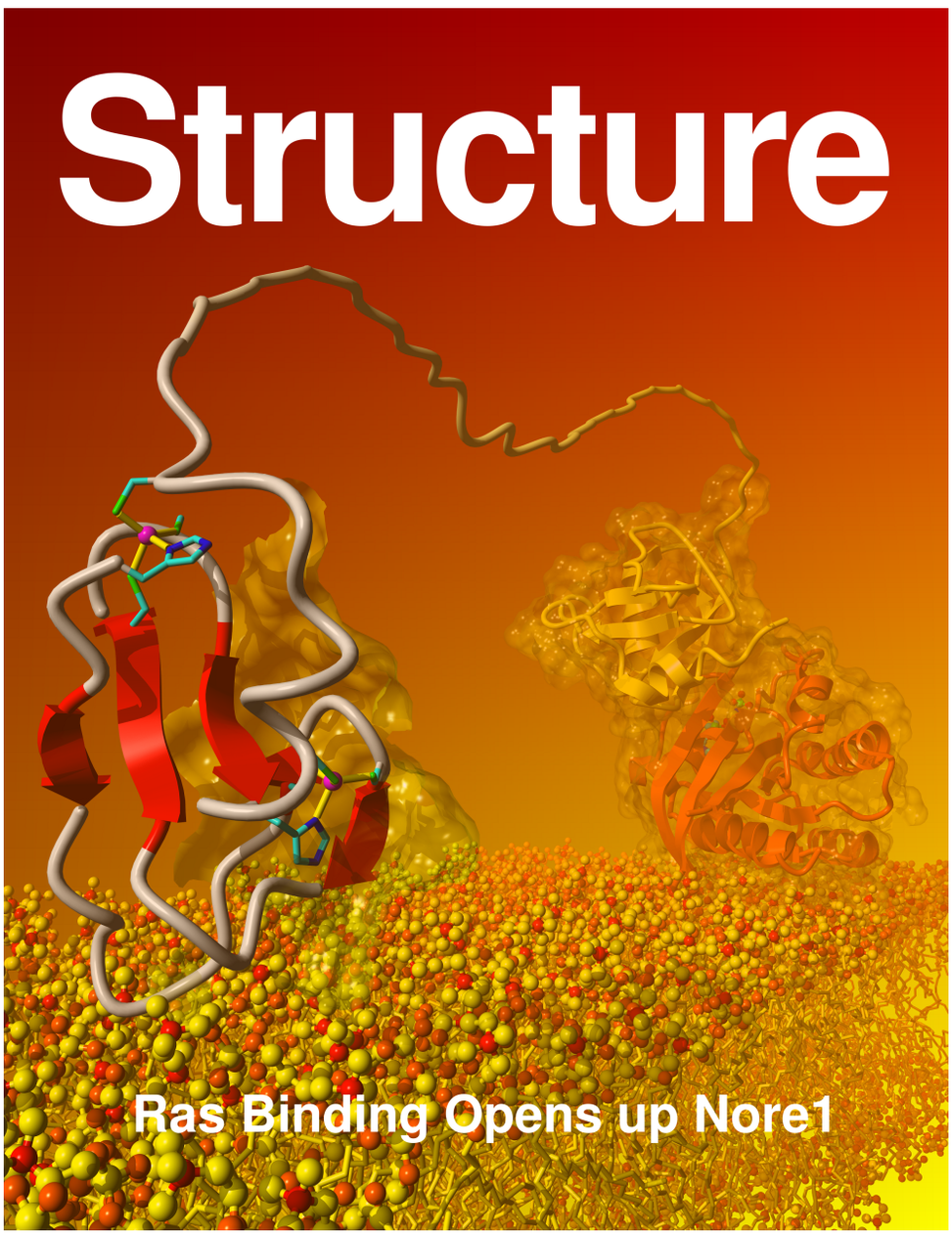
The Ras family contains small GTPases that are central regulators of cell proliferation, cell differentiation and cell death. Their signal transduction is mediated by various effector proteins. In a collaboration with Elena and Stefan Harjes at the Max-Planck Institute of Molecular Physiology in Dortmund, Germany, YASARA's NMR module and knowledge-based refinement force fields were used to help solve the structure of the Nore1 C1 domain, a novel Ras effector capable of inducing apoptosis. The cover illustration shows the C1 domain interacting with a membrane at the front.

# GTP-Ras disrupts the intramolecular complex of C1 and RA domains of Nore1

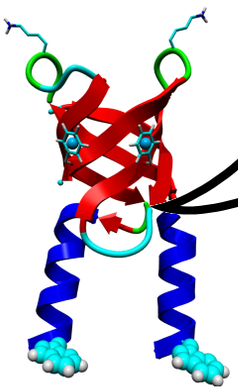
Harjes E, Harjes S, Wohlgemuth S, Müller KH, Krieger E, Herrmann C, Bayer P

*Structure* **14(5)**, 881-8 (2006)

*Molecular and Structural Biophysics, Max-Planck Institute of Molecular Physiology, Otto-Hahn Strasse 11, D-44227 Dortmund, Germany.*




The novel Ras effector mNore1, capable of inducing apoptosis, is a multidomain protein. It comprises a C1 domain homologous to PKC and an RA domain similar to the Ras effectors AF-6 and RalGDS. Here, we determine the affinity of these two domains to the active forms of Ras and Rap1 using isothermal calorimetric titration. The interaction of Ras/Rap1-GTP with the RA domain of mNore1 is weakened significantly by direct binding of the C1 domain to the RA domain. In order to analyze this observation in atomic detail, we solved the C1 solution structure by NMR. By determining chemical shifts and relaxation rates, we can show an intramolecular complex of C1-RA. GTP-Ras titration and binding to RA disrupts this complex and displaces the C1 domain. Once the C1 domain tumbles freely in solution, a lipid binding interface becomes accessible. Furthermore, we provide evidence of phosphatidylinositol 3-phosphate binding of the free C1 domain.



In a joint project with Ida van der Klei at Eukaryotic Microbiology, Groningen Biomolecular Sciences and Biotechnology Institute, we built a model for Pex5p (shown above), the receptor protein for the peroxisomal targeting signal peptide PTS-1. A dansylated PTS-1 derivative is bound to the model. The dansyl group allowed to measure fluorescence resonance energy transfer from surrounding Trp residues. These results could be interpreted based on the model. In addition, a strong dependence of Trp fluorescence on the pH was found, suggesting the existence of different oligomeric Pex5p forms on the acidic inside and the neutral outside of the peroxisomal membrane. This could be the basis of the shuttle mechanism that releases PTS-1 (and hence the translocated protein) on the inside.

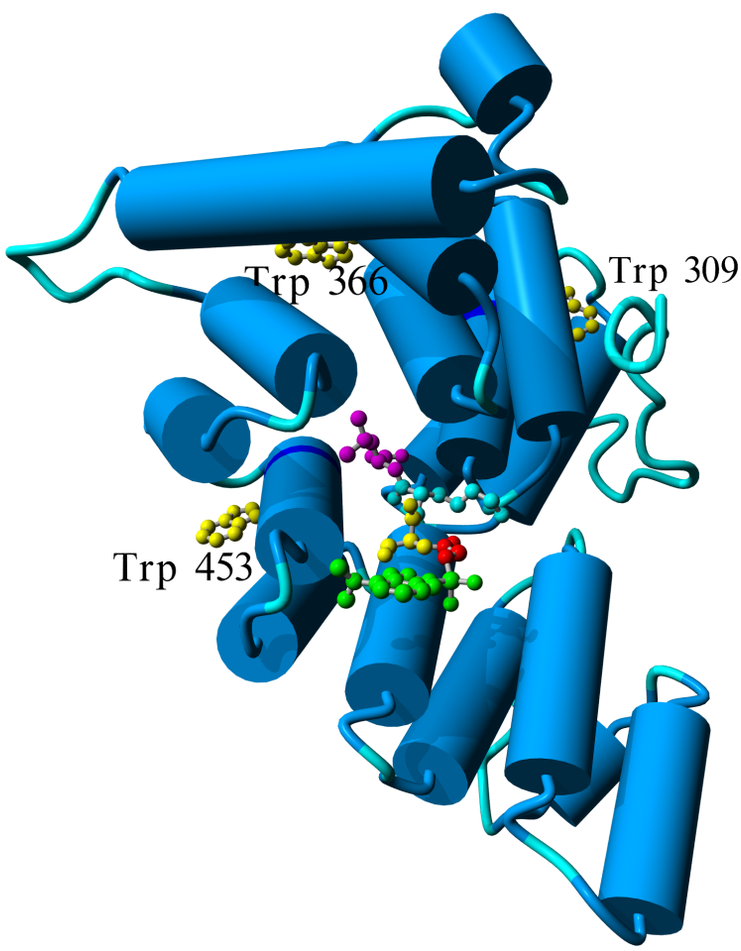
# Fluorescence analysis of the *Hansenula polymorpha* peroxisomal targeting signal-1 receptor, Pex5p

Boteva R[1], Koek A, Visser NV, Visser AJ, Krieger E, Zlateva T, Veenhuis M, van der Klei I[2]

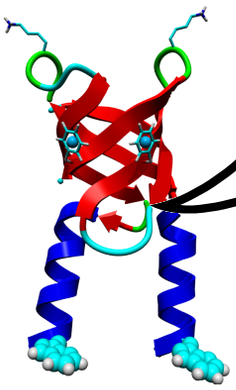
*Eur J Biochem.* **270**; 4332-8 (2003)

[1]*Institute of Molecular Biology, Bulgarian Academy of Sciences, Sofia, Bulgaria.*
[2]*Groningen Biomolecular Sciences and Biotechnology Institute, The Netherlands*

Correct sorting of newly synthesized peroxisomal matrix proteins is dependent on a peroxisomal targeting signal (PTS). So far two PTSs are known. PTS1 consists of a tripeptide that is located at the extreme C terminus of matrix proteins and is specifically recognized by the PTS1-receptor Pex5p. We studied Hansenula polymorpha Pex5p (HpPex5p) using fluorescence spectroscopy. The intensity of Trp fluorescence of purified HpPex5p increased by 25% upon shifting the pH from pH 6.0 to pH 7.2. Together with the results of fluorescence quenching by acrylamide, these data suggest that the conformation of HpPex5p differs at these two pH values. Fluorescence anisotropy decay measurements revealed that the pH affected the oligomeric state of HpPex5p, possibly from monomers/dimers at pH 6.0 to larger oligomeric forms at pH 7.2. Addition of dansylated peptides containing a PTS1, caused some shortening of the average fluorescence lifetime of the Trp residues, which was most pronounced at pH 7.2. Our data are discussed in relation to a molecular model of HpPex5p based on the three-dimensional structure of human Pex5p.




The applications



Together with John van Swieten and Peter Heutink at the Department of Neurology, EMC Rotterdam, we built a model of fibroblast growth factor 14, shown above. A Phe 145 > Ser mutation was discovered in a family affected by spinocerebellar ataxia, an inheritable disease causing a degeneration of certain brain regions responsible for muscle coordination. According to the model, Phe 145 forms an integral part of the hydrophobic protein core, a mutation to Ser leaves an empty hole. Changes of this type are known to destabilize proteins. The expression pattern of fgf14 in mice suggests a role in neuronal development and adult brain function. It is thus very plausible that a reduction in stability can cause the observed phenotype.

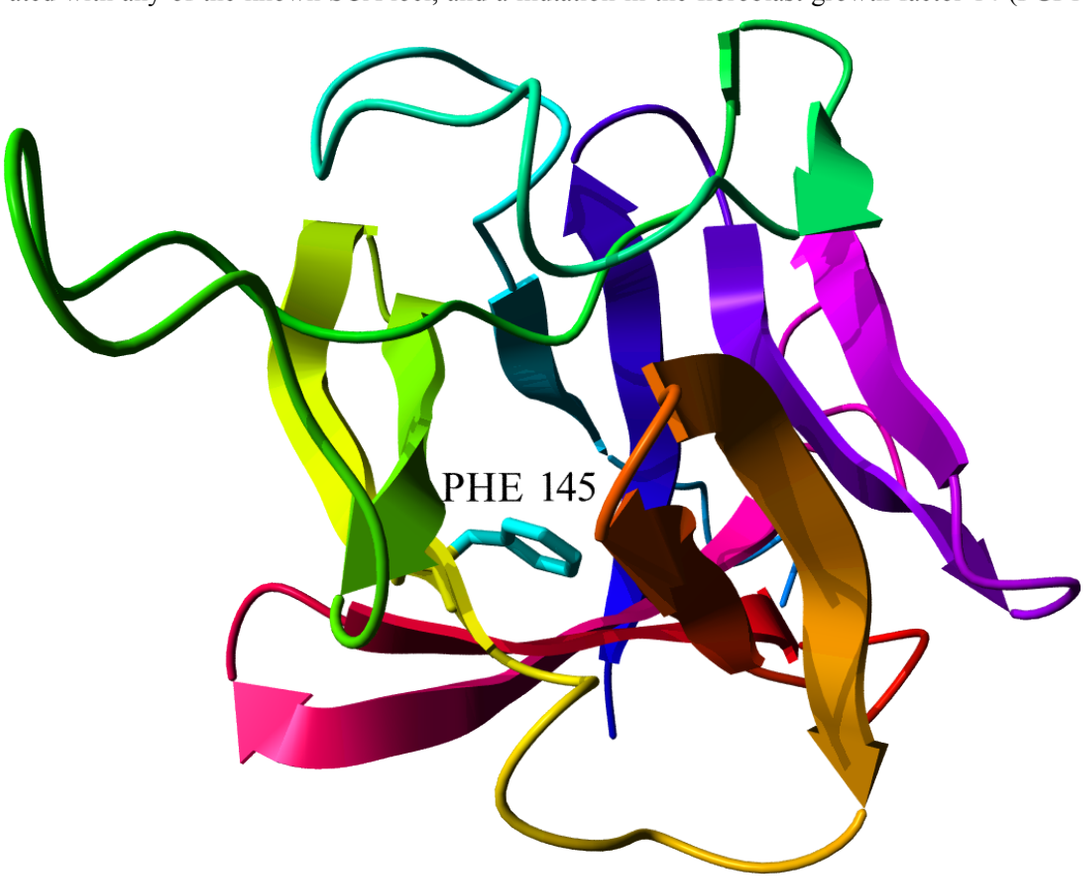
# A mutation in the fibroblast growth factor 14 gene is associated with autosomal dominant cerebellar ataxia

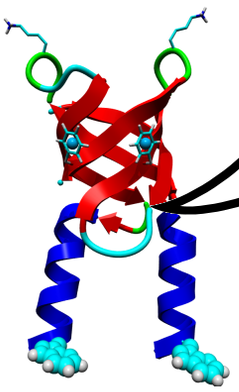
van Swieten JC, Brusse E, de Graaf BM, Krieger E, van de Graaf R, de Koning I, Maat-Kievit A, Leegwater P, Dooijes D, Oostra BA, Heutink P

*Am J Hum Genet.* 72, 191-9 (2003)

*Department of Neurology, Erasmus Medical Center Rotterdam, The Netherlands.*

Hereditary spinocerebellar ataxias (SCAs) are a clinically and genetically heterogeneous group of neurodegenerative disorders for which >=14 different genetic loci have been identified. In some SCA types, expanded tri- or pentanucleotide repeats have been identified, and the length of these expansions correlates with the age at onset and with the severity of the clinical phenotype. In several other SCA types, no genetic defect has yet been identified. We describe a large, three-generation family with early-onset tremor, dyskinesia, and slowly progressive cerebellar ataxia, not associated with any of the known SCA loci, and a mutation in the fibroblast growth factor 14 (FGF14) gene on chromosome 13q34. Our observations are in accordance with the occurrence of ataxia and paroxysmal dyskinesia in Fgf14-knockout mice. As indicated by protein modeling, the amino acid change from phenylalanine to serine at position 145 is predicted to reduce the stability of the protein. The present FGF14 mutation represents a novel gene defect involved in the neurodegeneration of cerebellum and basal ganglia.





In collaboration with Theo Geijtenbeek at the Department of Molecular Cell Biology, VUMC Amsterdam, we modeled the C-terminal C-type lectin domain of the DC-SIGN protein. DC-SIGN plays a crucial role as HIV-1 receptor on the surface of dendritic cells. The interaction with sugars occurs at the left of the two Calcium binding sites shown above. As the HIV-1 target protein gp120 is heavily glycosylated, one would expect it to bind in the same way as natural interaction partners like ICAM-3 – via the attached sugars. Surprisingly this turned out not to be the case: gp120 affinity is not affected by deglycosylation, indicating that there is a second, independent binding site for gp120. Inhibition of this interaction could thus prevent HIV infection while still allowing natural interactions to occur.

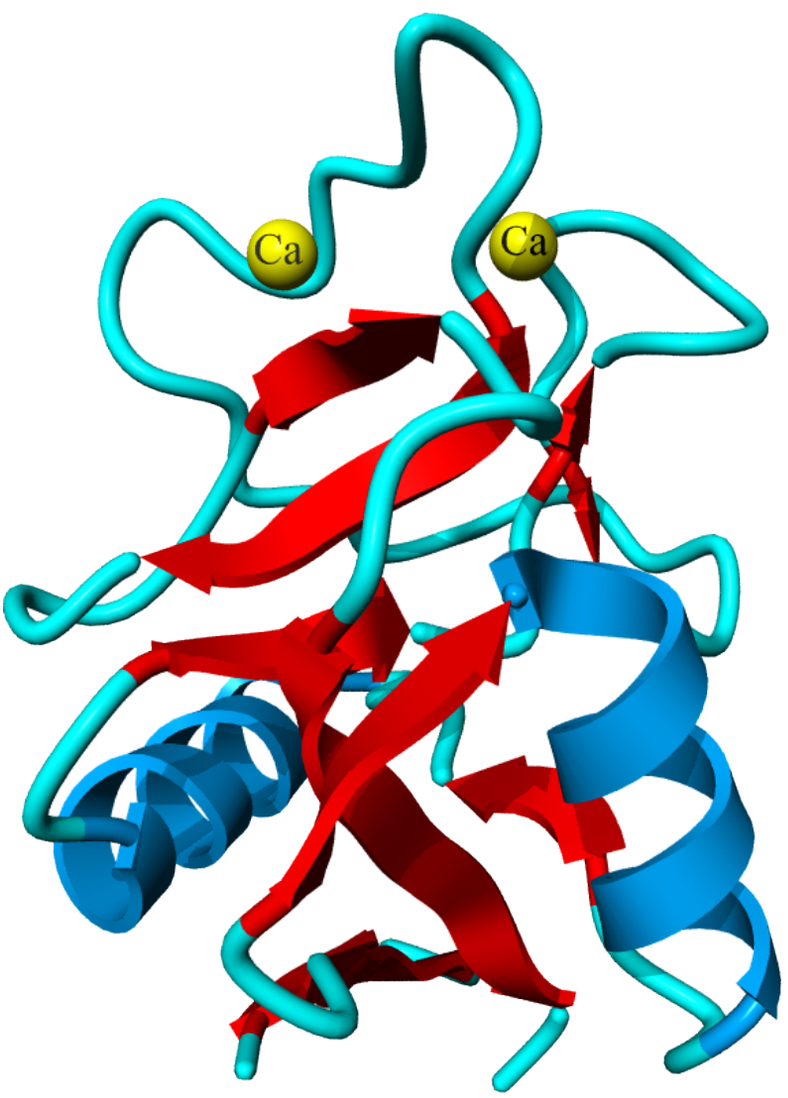
# Identification of different binding sites in the dendritic cell-specific receptor DC-SIGN for intercellular adhesion molecule 3 and HIV-1

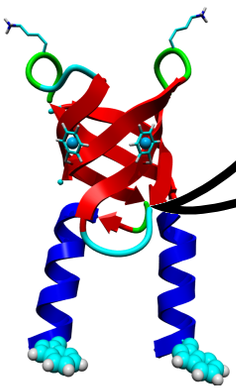
Geijtenbeek TB, van Duijnhoven GC, van Vliet SJ, Krieger E, Vriend G, Figdor CG, van Kooyk Y

*J Biol Chem.* **277**, 11314-20 (2002)

*Department of Molecular Cell Biology, Vrije University Medical Center Amsterdam, The Netherlands*

The novel dendritic cell (DC)-specific human immunodeficiency virus type 1 (HIV-1) receptor DC-SIGN plays a key role in the dissemination of HIV-1 by DC. DC-SIGN is thought to capture HIV-1 at mucosal sites of entry, facilitating transport to lymphoid tissues, where DC-SIGN efficiently transmits HIV-1 to T cells. DC-SIGN is also important in the initiation of immune responses by regulating DC-T cell interactions through intercellular adhesion molecule 3 (ICAM-3). We have characterized the mechanism of ligand binding by DC-SIGN and identified the crucial amino acids involved in this process. Strikingly, the HIV-1 gp120 binding site in DC-SIGN is different from that of ICAM-3, consistent with the observation that glycosylation of gp120, in contrast to ICAM-3, is not crucial to the interaction with DC-SIGN. A specific mutation in DC-SIGN abrogated ICAM-3 binding, whereas the HIV-1 gp120 interaction was unaffected. This DC-SIGN mutant captured HIV-1 and infected T cells in trans as efficiently as wild-type DC-SIGN, demonstrating that ICAM-3 binding is not necessary for HIV-1 transmission. This study provides a basis for the design of drugs that inhibit or alter interactions of DC-SIGN with gp120 but not with ICAM-3 or vice versa and that have a therapeutic value in immunological diseases and/or HIV-1 infections.





In a joint-project with Andreas Kungl at the Institute of Pharmaceutical Chemistry and Pharmaceutical Technology, University of Graz, we docked heparin and interleukin-8 (IL-8). The interaction of heparin and IL-8 is of key importance during inflammation. Experimental results indicated that one single long heparin 24mer might span both binding sites in dimeric IL-8 in a horseshoe-like fashion (Figure A). During a molecular dynamics simulation of such a complex, the interactions concentrated on two symmetry-related hot-spots (circles in Figure B). Key residues in these regions were mutated to alanine *in vitro* and *in silico*. Comparison of the measured and predicted binding energies showed good qualitative agreement.

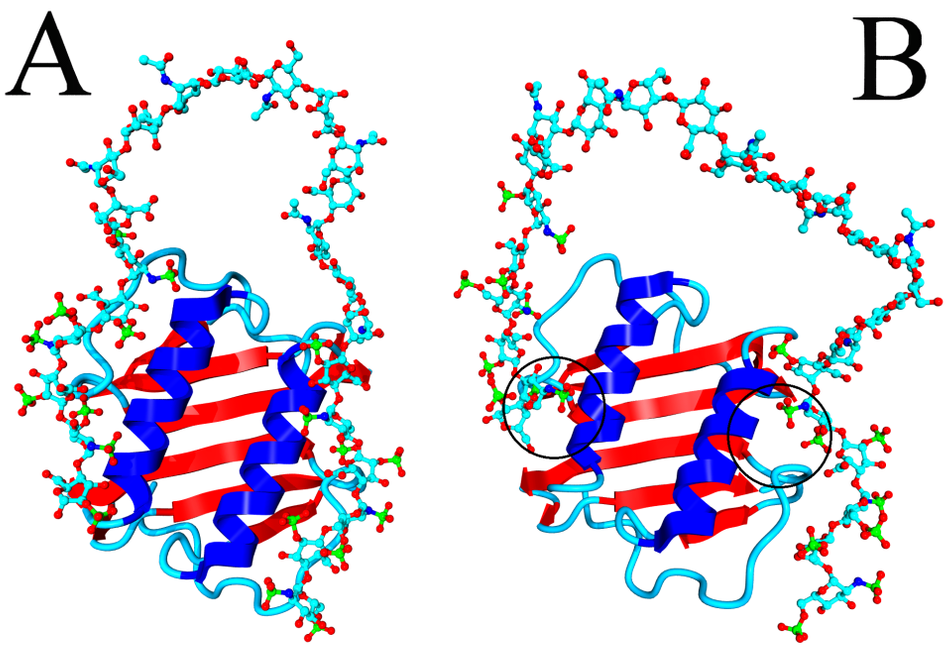
# A structural and dynamical model for the interaction of interleukin-8 and glycosaminoglycans: support from isothermal fluorescence titrations

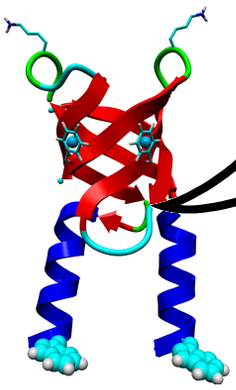
Krieger E, Geretti E, Brandner B, Goger B, Wells TN and Kungl AJ

*Proteins* **54**, 768-775 (2004)

*Institute of Pharmaceutical Chemistry and Pharmaceutical Technology, University of Graz, Austria*

Binding of IL-8 to GAGs on the surface of endothelial cells is crucial for the recruitment of neutrophils to an inflammatory site. Deriving structural knowledge about this interaction from in silico docking experiments has proven difficult due to the high flexibility and the size of GAGs. We therefore developed a docking method that takes into account ligand and protein flexibility by running ~15000 molecular dynamics simulations of the docking event with different initial orientations of the binding partners. The method was shown to successfully reproduce the residues of basic fibroblast growth factor involved in GAG binding. Docking of a heparin hexasaccharide to IL-8 gave an interaction interface involving the basic residues His18, Lys20, Arg60, Lys64, Lys67 and Arg68. By subjecting IL-8 single-site mutants, in which these amino acids were replaced by alanine, to isothermal fluorescence titrations the affinities for heparin were determined to be wtIL-8 > IL-8(H18A) >> IL-8(R68A) > IL-8(K67A) >> IL-8(K20A) > IL-8(R60A) >> IL-8(K64A). A comparison with the binding energies calculated from the model revealed high values for wtIL-8 and the H18A mutant and significantly lower but similar energies for the remaining mutants. Connecting the two fully sulfated hexasaccharides bound to each of the two IL-8 monomers in the dimeric chemokine by an N-acetylated dodecasaccharide gave a complex structure in which the GAG molecule aligned in an parallel fashion to the N-terminal a-helices of IL-8 like a horseshoe. A 5 ns molecular dynamics simulation of this complex confirmed its structural stability and revealed a reorientation in both binding sites where a disaccharide became the central binding unit. Isothermal fluorescence titration experiments using differently sulfated heparin disaccharides confirmed that a single disaccharide can indeed bind IL-8 with high affinity.





Together with Pascal Duijf at the Department of Human Genetics, UMC Nijmegen, we analyzed a newly discovered mutation in the p63 protein linked to ADULT syndrome. p63 is a transcription regulator of crucial importance. Previously described mutations all cluster around the alpha helix that docks in the major DNA groove and thus inhibit DNA binding. The new Arg 298 > Gln mutation is however located on the backside of the DNA binding domain, and was found to activate transcription. As only little is known about the precise arrangement of the activation complex, an explanation at the molecular level is currently not possible.

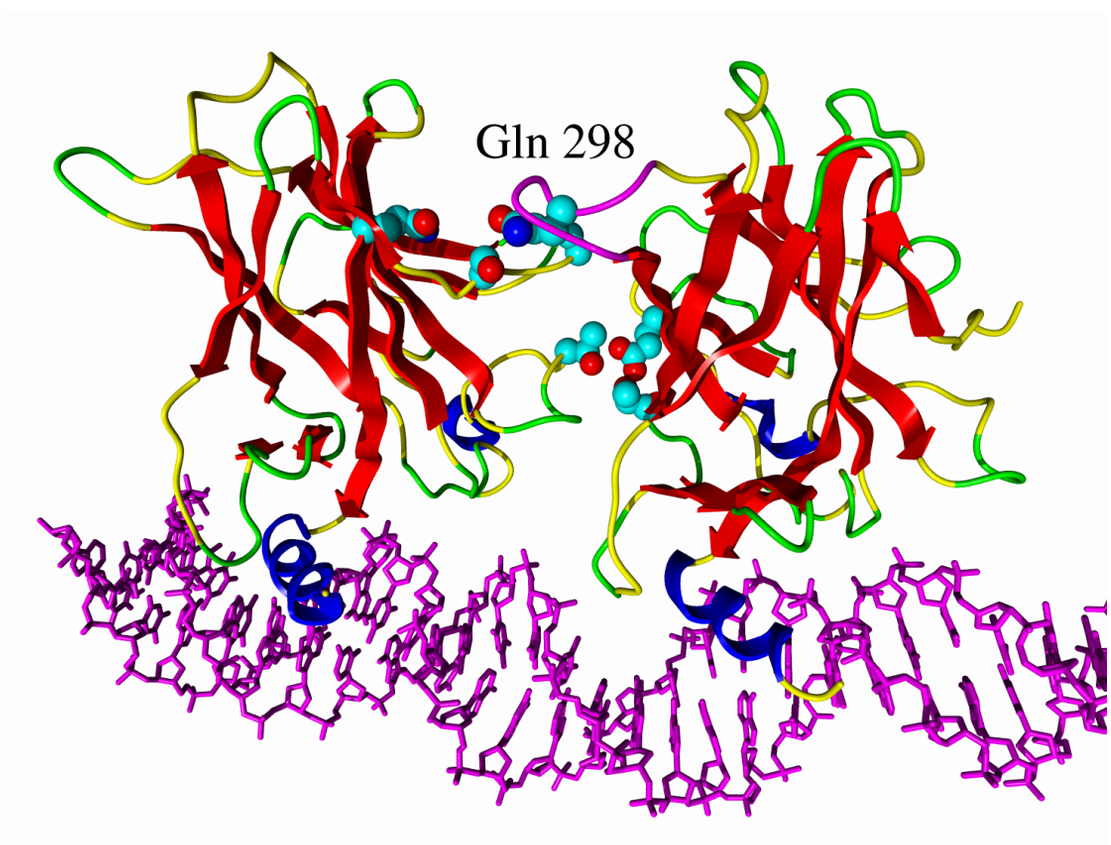
# Gain-of-function mutation in ADULT syndrome reveals the presence of a second transactivation domain in p63

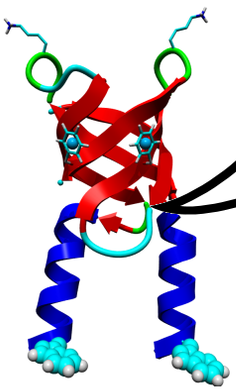
Duijf PH, Vanmolkot KR, Propping P, Friedl W, Krieger E, McKeon F, Dotsch V, Brunner HG, van Bokhoven H

*Hum Mol Genet.* **11**, 799-804 (2002)

*Department of Human Genetics, University Medical Centre Nijmegen, The Netherlands.*

The transcriptional co-activator p63 is of crucial importance for correct development of the limbs, ectodermal appendages (skin, nails, teeth, hair, glands), lip and palate. Mutations in the p63 gene are found in a number of human syndromes, including ectrodactyly-ectodermal dysplasia-cleft lip/palate (EEC) syndrome, limb-mammary syndrome (LMS), Hay-Wells syndrome and in non-syndromic split-hand/split-foot malformation (SHFM). Each syndrome has a specific pattern of mutations with different functional effects in in vitro functional assays. We report a mutation R298Q in acro-dermato-ungual-lacrimal-tooth (ADULT) syndrome, another EEC-like condition. The mutation is located in the DNA binding domain of p63, which harbors almost all EEC associated mutations. However, unlike mutations in EEC syndrome, the R298Q ADULT syndrome mutation does not impair DNA binding. Rather, the mutation confers novel transcription activation capacity on the DeltaN-p63gamma isoform, which normally does not possess such activity. These results confirm that ADULT syndrome is a clinically as well as molecularly distinct member of the expanding p63 mutation family of human malformation syndromes. Our results further show that p63 contains a second transactivation domain which is normally repressed and can become activated by mutations in the DNA binding domain of p63.





In collaboration with Jan Koenderink and Jan Joep de Pont at the Department of Biochemistry, Nijmegen Center for Molecular Life Sciences, we built a model for the E2 form of H,K-ATPase. The model revealed one strong potassium binding site in close proximity to a salt-bridge between Lys 791 and Glu 820. A large number of mutations was analyzed with respect to this model, most interesting of which is a E820Q exchange. In the model, this mutation disrupts the salt-bridge and lets Lys 791 take the place of the potassium ion. This would explain the experimental finding, that the mutant enzyme functions also in the absence of potassium.

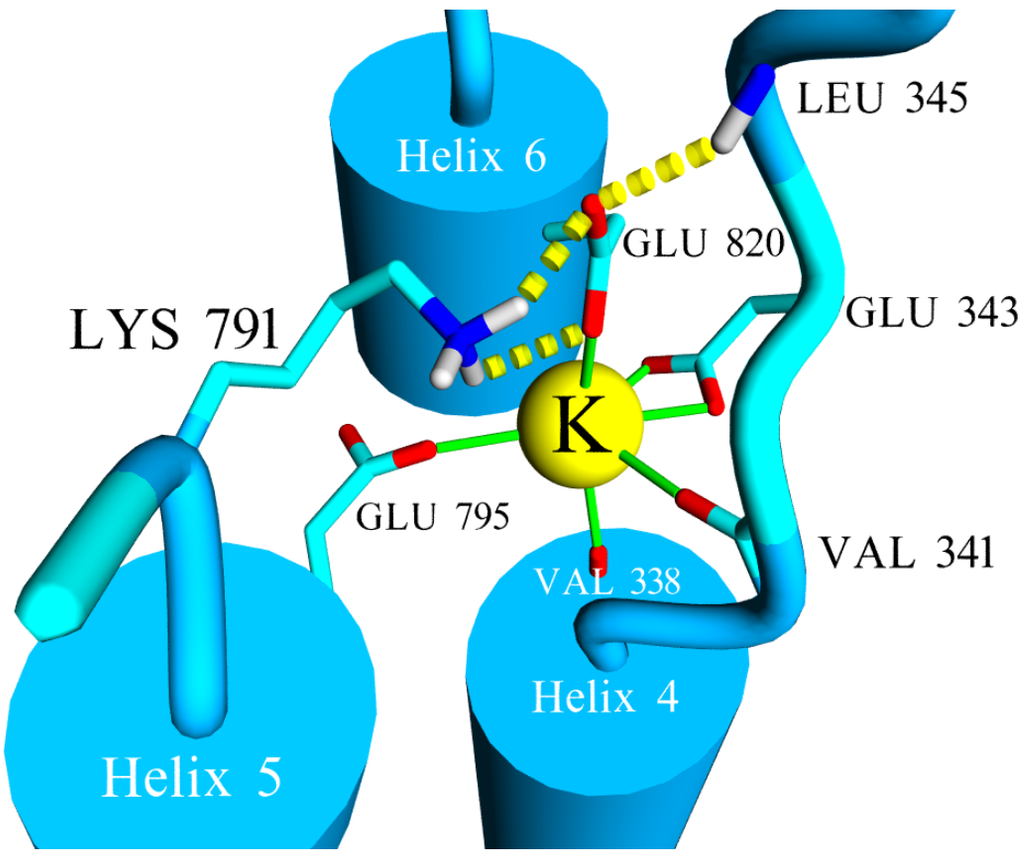
# A conformational specific interhelical salt bridge is essential for the E2 preference of gastric H,K-ATPase

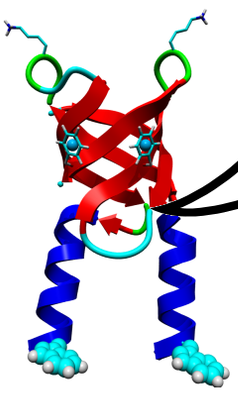
Koenderink JB, Swarts HGP, Willems PHGM, Krieger E and De Pont JJHHM

*J Biol Chem.* **279**, 16417-16427 (2004)

*Department of Biochemistry, Nijmegen Center for Molecular Life Sciences, The Netherlands*

Homology modeling of gastric H,K-ATPase based on E2- model of sarco(endo)plasmic reticulum Ca2+-ATPase revealed the presence of both a single high-affinity binding site for K+ and a E2-form specific salt bridge between Glu820 (M6) and Lys791 (M5). Replacement of Lys791 by an Ala residue significantly reduced the K+ affinity, without altering the E2-preference of the enzyme. Replacement of Glu820 by a Gln residue rendered the enzyme active in the absence of K+ with preference for the E1-conformation. The double K791A-E820Q mutant had no ATPase activity but showed a preference for the E2-conformation reaction as measured by the effect of specific inhibitors on the phosphorylation reaction. Modeling of the E820Q mutant revealed that the head group of the Lys residue together with a water molecule fills the K+-binding pocked, thus explaining the K+-independent activity of this mutant. These data indicate that the salt bridge is essential for high-affinity K+ binding and E2-preference of H,K-ATPase. Moreover, its breakage provides a structural explanation for the K+-insensitive activity and E1-preference of the E820Q mutant.





Together with Herman Swarts, Jan Koenderink and Jan Joep de Pont at the Department of Biochemistry, NCMLS, Radboud University Nijmegen, the Netherlands, the analysis of gastric H,K-ATPase was continued with the focus on the highly conserved residue Asn 792. In the homology model, Asn 792 is located next to a potassium binding pocket, but not directly involved in binding. Instead it mediates the hydrogen-bonding network of the pocket-forming residues, and consequently significant changes are observed in Asn 792 point-mutants. Energy calculations of these mutants correlated well with experiment.

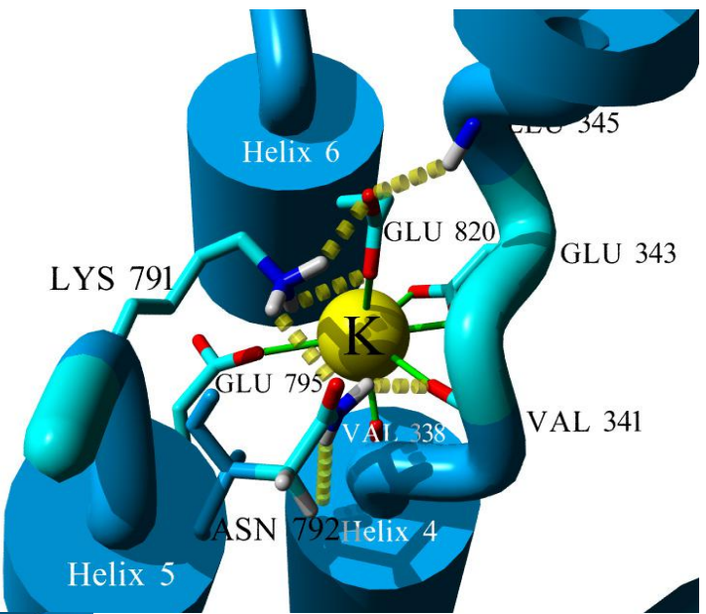
# Asn792 participates in the hydrogen bond network around the K+-binding pocket of gastric H,K-ATPase

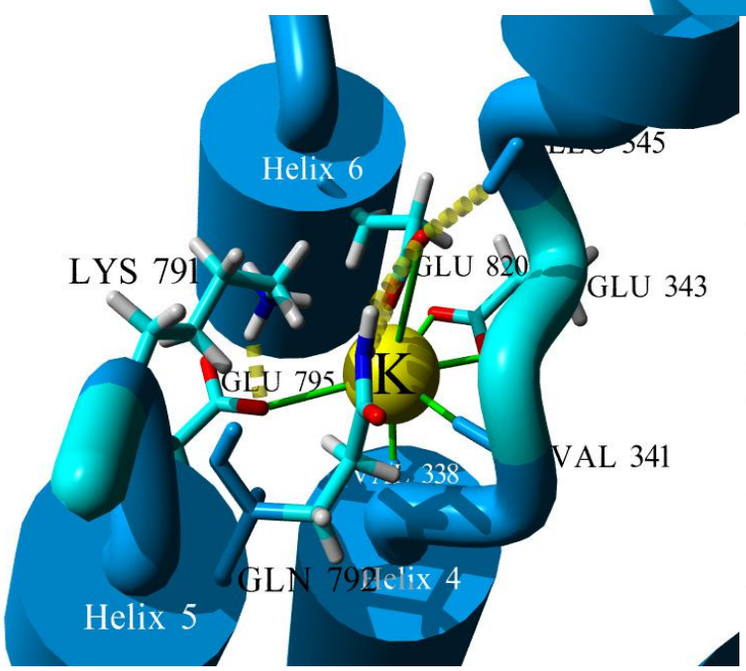
Swarts HG, Koenderink JB, Willems PH, Krieger E, De Pont JJ

*J Biol Chem.* **280(12),** 11488-94 (2005)

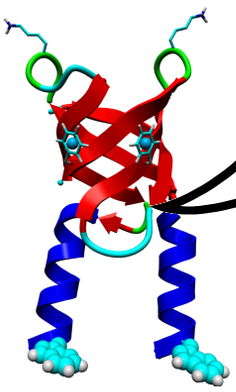
*Department of Biochemistry, Nijmegen Centre for Molecular Life Sciences, P. O. Box 9101, 6500 HB Nijmegen, The Netherlands.*

Asn792 present in M5 of gastric H,K-ATPase is highly conserved within the P-type ATPase family. A direct role in K+ binding was postulated for Na,K-ATPase but was not found in a recent model for gastric H,K-ATPase (Koenderink, J. B., Swarts, H. G. P., Willems, P. H. G. M., Krieger, E., and De Pont, J. J. H. H. M. (2004) J. Biol. Chem. 279, 16417-16424). Therefore, its role in K+ binding and E1/E2 conformational equilibrium in gastric H,K-ATPase was studied by site-directed mutagenesis and expression in Sf9 cells. N792Q and N792A, but not N792D and N792E, had a markedly reduced K+ affinity in both the ATPase and dephosphorylation reactions. In addition, N792A shifted the conformational equilibrium to the E1 form. In double mutants, the effect of N792A on K+ sensitivity was overruled by either E820Q (K(+)-independent activity) or E343D (no dephosphorylation activity). Models were made for the




mutants based on the E2 structure of Ca(2+)-ATPase. In the wild-type model the acid amide group of Asn792 has hydrogen bridges to Lys791, Ala339, and Val341. Comparison of the effects of the various mutants suggests that the hydrogen bridge between the carbonyl oxygen of Asn792 and the amino group of Lys791 is essential for the K+ sensitivity and the E2 preference of wild-type enzyme. Moreover, there was a high positive correlation (r = 0.98) between the in silico calculated energy difference of the E2 form (mutants versus wild type) and the experimentally measured IC50 values for vanadate, which reflects the direction of the E2<-->E1 conformational equilibrium. These data strongly support the validity of the model in which Asn792 participates in the hydrogen bond network around the K(+)-binding pocket.



Li Yan Qiu and the ATPase team at the Department of Biochemistry, NCMLS, Radboud University Nijmegen, managed to transfer the ability to bind the cardiac glycoside Ouabain between two ATPases by mutating just seven amino acids. The structural basis was analyzed with YASARA, combined with a Flexx docking study by Gijs Schafenaar. While high Ouabain doses are extremely toxic and have been used as an arrow poison, low doses are used to treat heart-related problems like low blood pressure. The JBC cover shows the model embedded in a membrane.

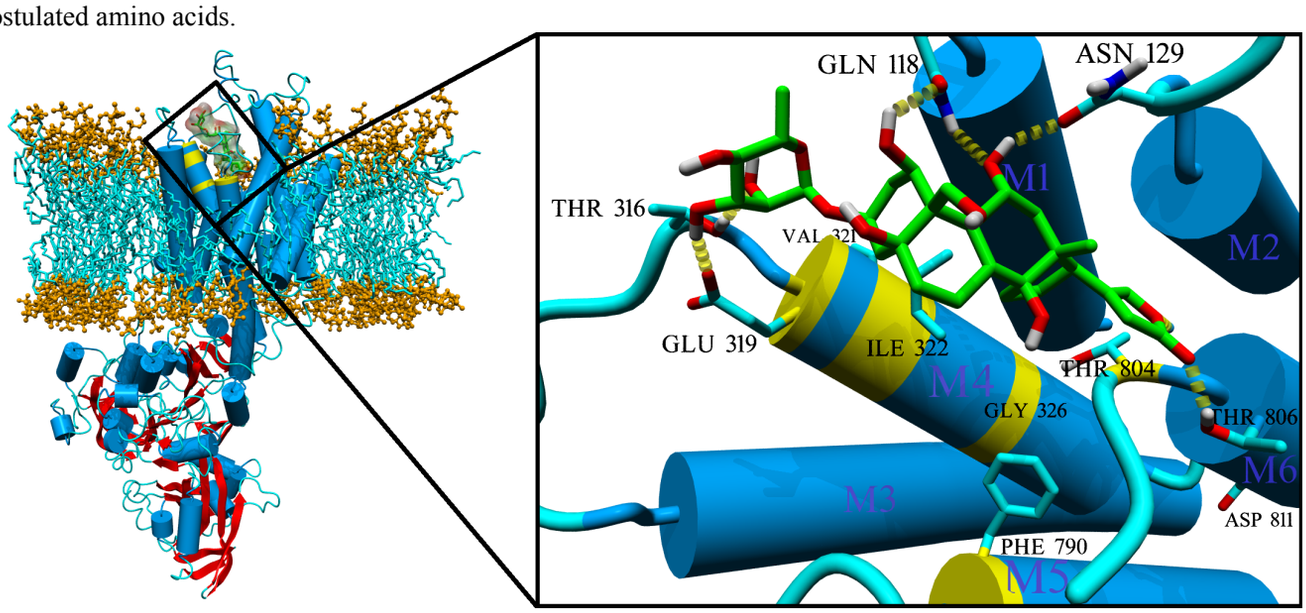
# Reconstruction of the complete ouabain-binding pocket of Na,K-ATPase in gastric H,K-ATPase by substitution of only seven amino acids

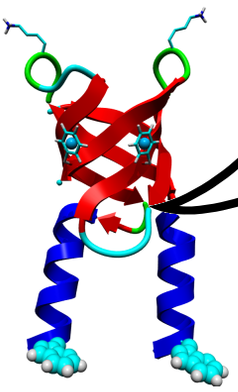
Qiu LY, Krieger E, Schaftenaar G, Swarts HG, Willems PH, De Pont JJ, Koenderink JB.

J Biol Chem. **280(37)**, 32349-55 (2005)

*Department of Biochemistry, Nijmegen Centre for Molecular Life Sciences, Radboud University Nijmegen Medical Centre, Netherlands.*


Although cardiac glycosides have been used as drugs for more than 2 centuries and their primary target, the sodium pump (Na,K-ATPase), has already been known for 4 decades, their exact binding site is still elusive. In our efforts to define the molecular basis of digitalis glycosides binding we started from the fact that a closely related enzyme, the gastric H,K-ATPase, does not bind glycosides like ouabain. Previously, we showed that a chimera of these two enzymes, in which only the M3-M4 and M5-M6 hairpins were of Na,K-ATPase, bound ouabain with high affinity (Koenderink, J. B., Hermsen, H. P. H., Swarts, H. G. P., Willems, P. H. G. M., and De Pont, J. J. H. H. M. (2000) Proc. Natl. Acad. Sci. U. S. A. 97, 11209-11214). We also demonstrated that only three amino acids (Phe(783), Thr(797), and Asp(804)) present in the M5-M6 hairpin of Na,K-ATPase were sufficient to confer high affinity ouabain binding to a chimera which contained in addition the M3-M4 hairpin of Na,K-ATPase (Qiu, L. Y., Koenderink, J. B., Swarts, H. G., Willems, P. H., and De Pont, J. J. H. H. M. (2003) J. Biol. Chem. 278, 47240-47244). To further pinpoint the ouabain-binding site here we used a chimera-based loss-of-function strategy and identified four amino acids (Glu(312), Val(314), Ile(315), Gly(319)), all present in M4, as being important for ouabain binding. In a final gain-of-function study we showed that a gastric H,K-ATPase that contained Glu(312), Val(314), Ile(315), Gly(319), Phe(783), Thr(797), and Asp(804) of Na,K-ATPase bound ouabain with the same affinity as the native enzyme. Based on the E(2)P crystal structure of Ca(2+)-ATPase we constructed a homology model for the ouabain-binding site of Na,K-ATPase involving all seven amino acids as well as several earlier postulated amino acids.





In a joint project with Hannie Kremer and Erwin van Wijk at the Department of Otorhino-laryngology, UMC Nijmegen, we investigated a Thr > Ile mutation in the gamma actin 1 gene of patients with hearing loss. The δ-methyl group of Ile 278 was found to bump into Met 313, without any space for the side chains to reorient. We concluded that this mutation is serious enough to explain the observed phenotype, especially as the loops attached to helices 9 and 11 are involved in actin polymerization and ATP binding. Surprisingly, an actin orthologue in *D.discoideum* contains an Ile residue at this position too. But in this case, a correlated Thr > Ala mutation provides space for the larger Ile side chain.

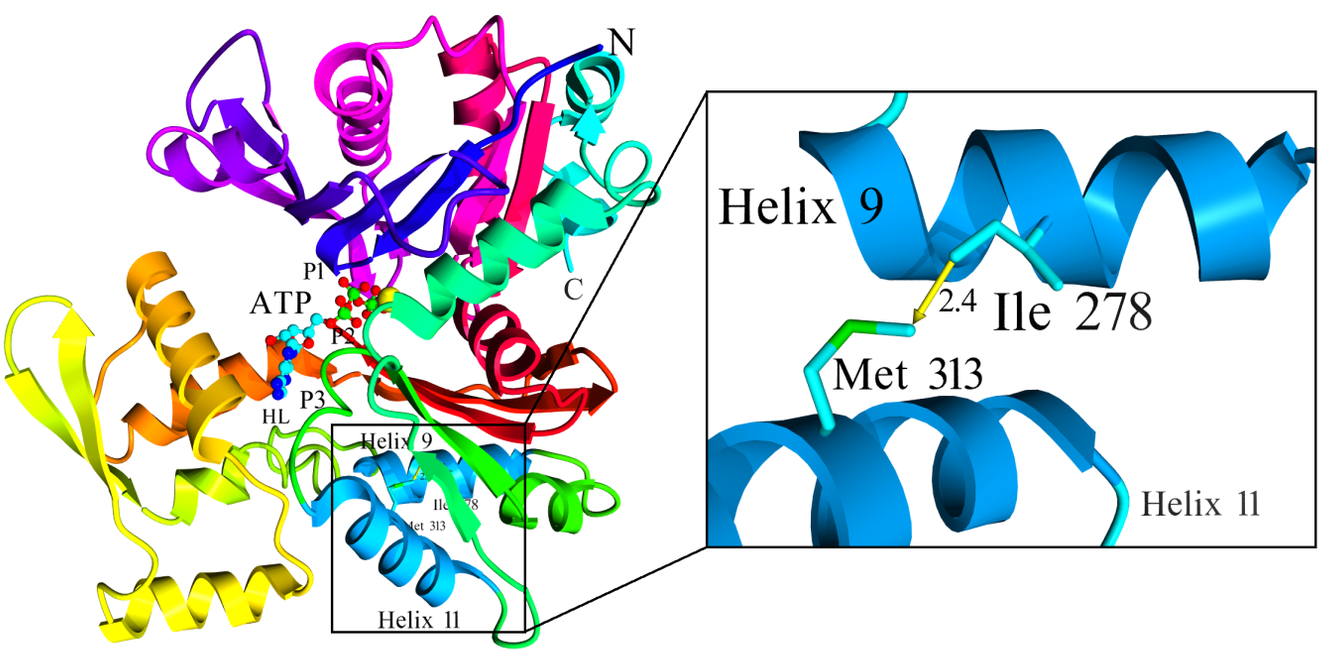
# A mutation in the gamma actin 1 (ACTG1) gene causes autosomal dominant hearing loss

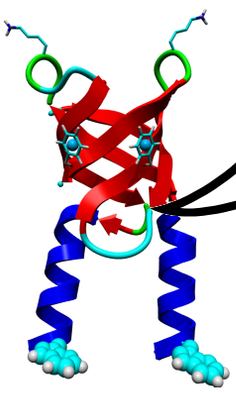
van Wijk E, Krieger E, Kemperman MH, de Leenheer EMR, Huygen PLM, Cremer CWRJ, Cremers FPM, Kremer H

*J Med Genet.* **40,** 879-884 (2004)

*Department of Otorhinolaryngology, UMC Nijmegen, The Netherlands*

Linkage analysis in a multigenerational family with autosomal dominant hearing loss yielded a chromosomal localization of the underlying genetic defect in the DFNA20/26 locus at 17q25-qter. The 6-cM critical region harbored the gamma 1 actin (*ACTG1*) gene which was considered an attractive candidate gene because actins constitute important structural elements of the inner ear hair cells. We identified a Thr278Ile mutation in helix 9 of the modeled protein structure. The alteration of residue Thr278 is predicted to have a small but significant effect on the gamma 1 actin structure due to its close proximity to a methionine residue at position 313 in helix 11. Moreover, the Thr278 residue is highly conserved throughout eukaryotic evolution. Using a known actin structure the mutation could be predicted to impair actin polymerization. The progression of DFNA20/26 associated hearing loss is similar to DFNA1, DFNA17 and DFNA22 associated hearing impairment. The latter are all caused by mutations in genes encoding proteins functionally related to actin. The severity of the deafness in this DFNA20/26 family matches the hearing loss observed in (young) patients with Usher syndrome type 1 (USH1). USH1 is caused by autosomal recessive mutations in genes encoding proteins which are indicated to form protein complexes involved in stereocilia structure, cohesion and anchorage. These findings strongly suggest that the Thr278Ile mutation in *ACTG1* represents the first disease causing germline mutation in a cytoplastic actin isoform.





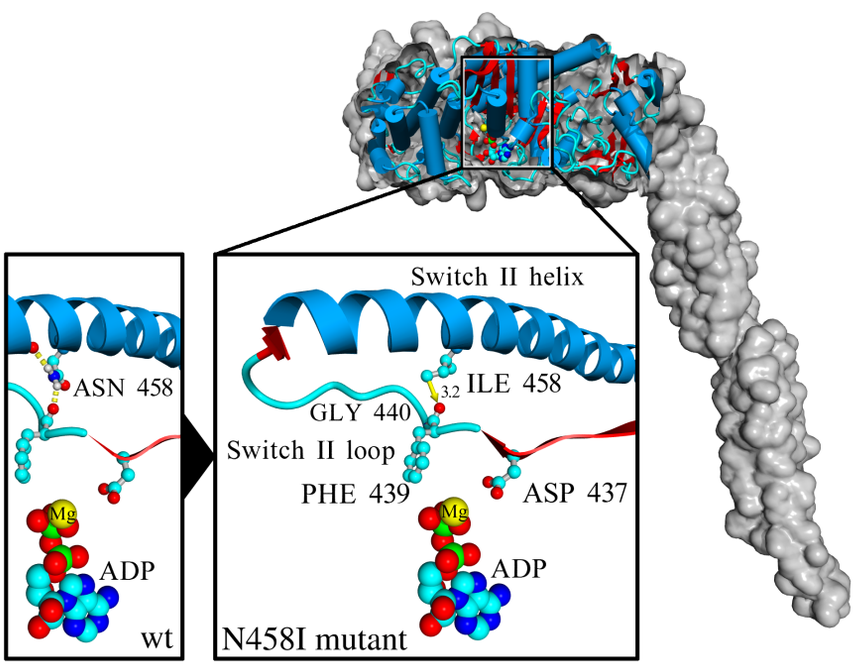
In a second collaboration project with Mirjam Luijendijk, Erwin van Wijk and Hannie Kremer, we built a model of myosin VIIa to analyze a mutation found to impair hearing. Myosin VIIa is expressed in the inner ear, where it was shown to be involved in cross-linking adjacent stereocilia. In the wild-type enzyme, residue Asn 458 attaches the switch II loop to the following helix via two hydrogen bonds to backbone oxygens. One of these belongs to Phe 439 which in turn is followed by Gly 440. This residue was found experimentally to be the central hinge when the myosin head domains reorient during the power stroke. Ile 458 disturbs this interaction, indicating that myosin VIIa does not just play a passive structural role.

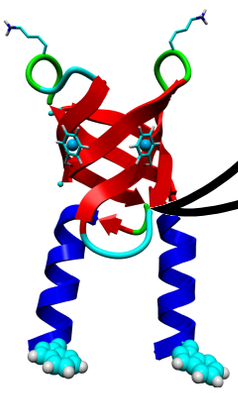
# Identification and molecular modeling of a mutation in the motor head domain of myosin VIIA in a family with autosomal dominant hearing impairment

Luijendijk MWJ, van Wijk E, Krieger E, Bischoff AMLC, Cremers CWRJ, Cremers FPM, Kremer H, Pennings RJE, Weekamp H, Cruysberg JRM, Huygen PLM and Brunner HG

*Hum Genet.* **115**, 149-156 (2004)

Myosin VIIA is an unconventional myosin that has been implicated in Usher syndrome type 1B, nonsyndromic autosomal recessive hearing impairment (DFNB2) and autosomal dominant hearing impairment (DFNA11). Unconventional myosins are actin-based motor molecules that transduce chemical energy derived from ATP into a force enabling them to move along actin filaments. The structure of the myosin VIIA protein is highly conserved in vertebrates and invertebrates. Here, we present a family (W02-011) with nonsyndromic autosomal dominant hearing impairment that clinically resembles the previously published DFNA11 family. The affected family members from the Dutch family show a flat audiogram at young ages and only modest progression, most clear at the high frequencies Patients also suffer from minor vestibular symptoms. Linkage analysis indeed yielded a maximum two-point lodscore of 3.43 for marker D11S937 located within 1 cM of the myosin VIIA gene. The 49 exons and splice sites of the myosin VIIA gene were sequenced and 11 nucleotide variations were found. Ten nucleotide changes represent benign intronic variants, silent exon mutations or non-pathologic amino acid substitutions. One variant, a c.1373 A>T transversion that is heterozygously present in all affected family members and absent in 300 healthy individuals, results in a Asn458Ile amino acid substitution. Asn458 is located in a region of the myosin VIIA motor domain that is highly conserved in different classes of myosins and myosins of different species. To evaluate whether the Asn458Ile mutation is indeed responsible for the hearing impairment, a molecular model of myosin VIIA was built based on the known structure of the myosin II heavy chain from *Dictyostelium discoideum*. In this model, the isoleucine residue at position 458 can't form a hydrogen bond with phenylalanine at position 439 in the switch II loop and because of its larger side chain compared to asparagine, pushes the switch II loop in the myosin VIIA motor domain towards the ATPase binding pocket. This could possibly disrupt ATP/ADP binding as well as impair the myosin power stroke, which would have a severe effect on the function of the myosin VIIA protein.





In another joint-project with Ersan Kalay and Hannie Kremer at the Department of Human Genetics, Radboud University Nijmegen Medical Centre, YASARA was used to analyze a point-mutation in myosin XVA. While myosins are best known for their ATP-dependent function as motor proteins in muscle contraction, they take part in many other processes, most of which involve an interaction with actin proteins. Myosin 15A for example is located in the inner ear and required for the assembly of stereocilia that sense the sound waves. Consequently, the Gly1831Val mutation described here is linked to hearing loss.

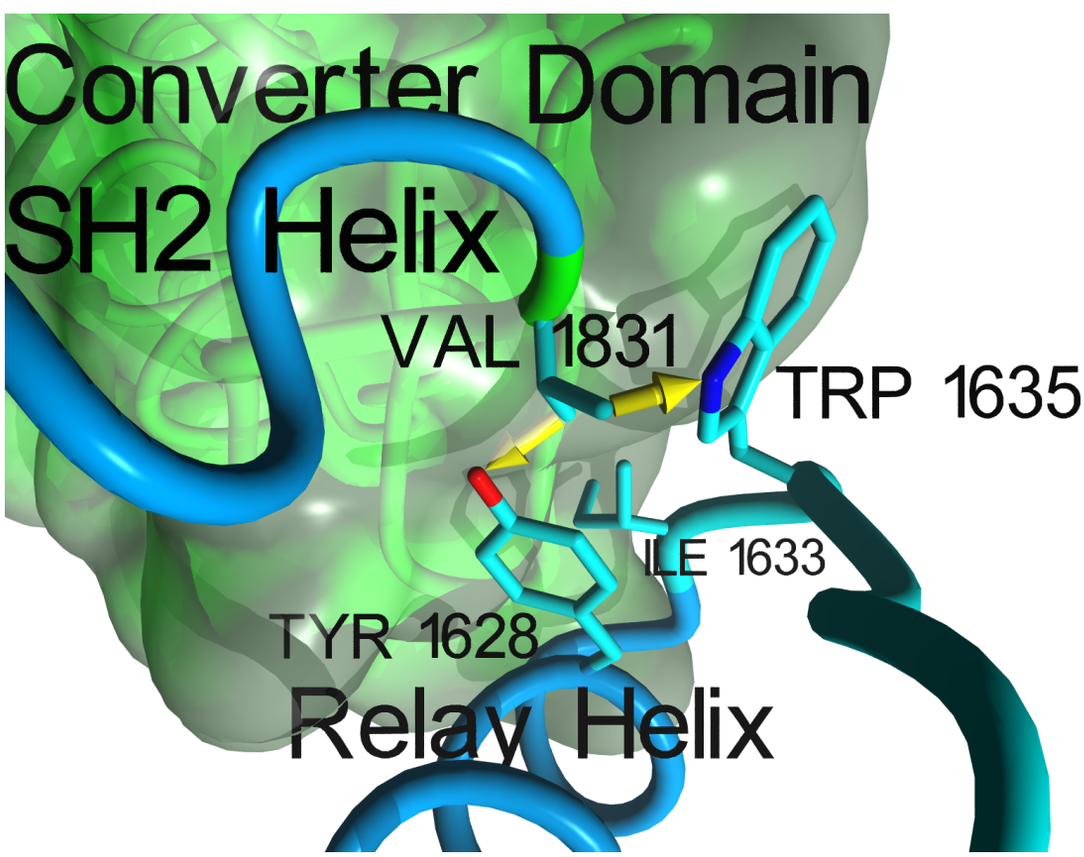
# MYO15A (DFNB3) mutations in Turkish hearing loss families and functional modeling of a novel motor domain mutation

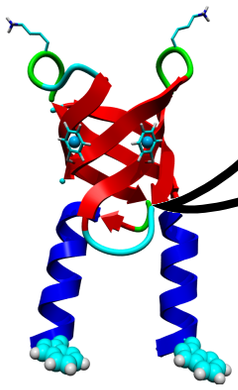
Kalay E, Uzumcu A, Krieger E, Caylan R, Uyguner O, Ulubil-Emiroglu M, Erdol H, Kayserili H, Hafiz G, Başerer N, Heister AJ, Hennies HC, Nürnberg P, Başaran S, Brunner HG, Cremers CW, Karaguzel A, Wollnik B, Kremer H

*Am J Med Genet A.* **143A(20)**, 2382-9 (2007)

*Department of Human Genetics, Radboud University Nijmegen Medical Centre, Nijmegen, The Netherlands. ersankalay@hotmail.com*

Myosin XVA is an unconventional myosin which has been implicated in autosomal recessive nonsyndromic hearing impairment (ARNSHI) in humans. In Myo15A mouse models, vestibular dysfunction accompanies the autosomal recessive hearing loss. Genomewide homozygosity mapping and subsequent fine mapping in two Turkish families with ARNSHI revealed significant linkage to a critical interval harboring a known deafness gene MYO15A on chromosome 17p13.1-17q11.2. Subsequent sequencing of the MYO15A gene led to the identification of a novel missense mutation, c.5492G-->T (p.Gly1831Val) and a novel splice site mutation, c.8968-1G-->C. These mutations were not detected in additional 64 unrelated ARNSHI index patients and in 230 Turkish control chromosomes. Gly1831 is a conserved residue located in the motor domains of the different classes of myosins of different species. Molecular modeling of the motor head domain of the human myosin XVa protein suggests that the Gly1831Val mutation inhibits the powerstroke by reducing backbone flexibility and weakening the hydrophobic interactions necessary for signal transmission to the converter domain.





After three articles on ion pumps, the collaboration with Jan Koenderink switched to a different class of membrane transporters: the secretory protein MRP4, which can pump a large number of compounds out of the cell, including antibiotics and cytostatics. Azza El-Sheikh at the Department of Pharmacology and Toxicology, Radboud University Nijmegen Medical Centre, The Netherlands, constructed several mutants that affected the transport efficiency for the second messenger cGMP and the cytostatic methotrexate. Homology modeling with YASARA showed that the mutated side-chains face the central pore of MPR4 and are most likely part of the substrate binding pocket.

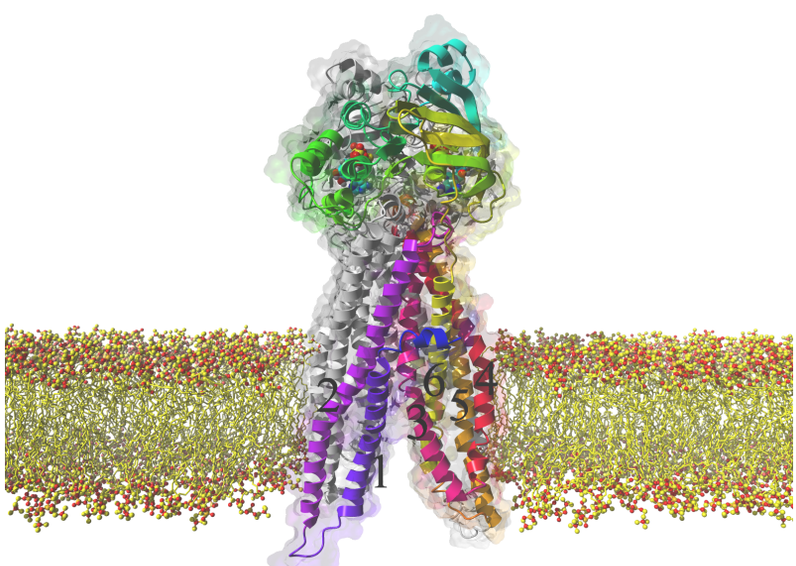
# Functional role of arginine 375 in transmembrane helix 6 of multidrug resistance protein 4 (MRP4/ABCC4)

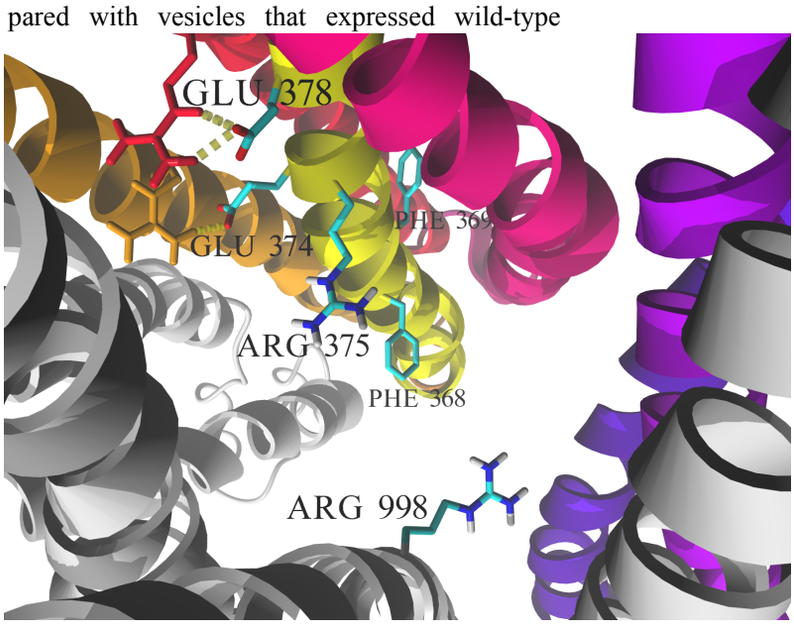
El-Sheikh AA, van den Heuvel JJ, Krieger E, Russel FG, Koenderink JB

*Mol Pharmacol.* **74(4)**, 964-71 (2008)

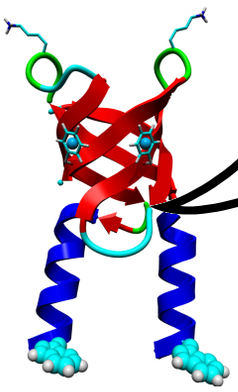
*Department of Pharmacology and Toxicology, Radboud University Nijmegen Medical Centre, Nijmegen Centre for Molecular Life Sciences, P.O. Box 9101, 6500 HB Nijmegen, The Netherlands.*

Multidrug resistance protein (MRP) 4 transports a variety of endogenous and xenobiotic organic anions. MRP4 is widely expressed in the body and specifically localized to the renal apical proximal tubule cell membrane, where it mediates the excretion of these compounds into urine. To characterize the MRP4 substrate-binding site, the amino acids Phe368, Phe369, Glu374, Arg375, and Glu378 of transmembrane helix 6, and Arg998 of helix 12, localized in the intracellular half of the central pore, were mutated into the corresponding amino acids of MRP1 and MRP2. Membrane vesicles isolated from human embryonic kidney 293 cells overexpressing these mutants showed significantly reduced methotrexate (MTX) and cGMP transport activity compared with vesicles that expressed wild-type




MRP4. The only exception was substitution of Arg375 with serine, which had no effect on cGMP transport but significantly decreased the affinity of MTX. Substitution of the same amino acid with a positively charged lysine returned the MTX affinity to that of the wild type. Furthermore, MTX inhibition of MRP4-mediated cGMP transport was noncompetitive, and the inhibition constant was increased by introduction of the R375S mutation. A homology model of MRP4 showed that Arg375 and Arg998 face right into the central aqueous pore of MRP4. We conclude that positively charged amino acids in transmembrane helices 6 and 12 contribute to the MRP4 substrate-binding pocket.



Together with Peter van den Berghe and Leo Klomp at the Department of Metabolic and Endocrine Diseases, University Medical Center Utrecht, The Netherlands, YASARA was used to build homology models and investigate mutations in another ion-transporting transmembrane ATPase: ATP7B, which exports copper ions in liver and brain cells and thus helps to balance the copper level. Malfunction of ATP7B caused by mutations in its gene leads to copper accumulation, which in turn causes a wide range of symptoms known as Wilson's disease.

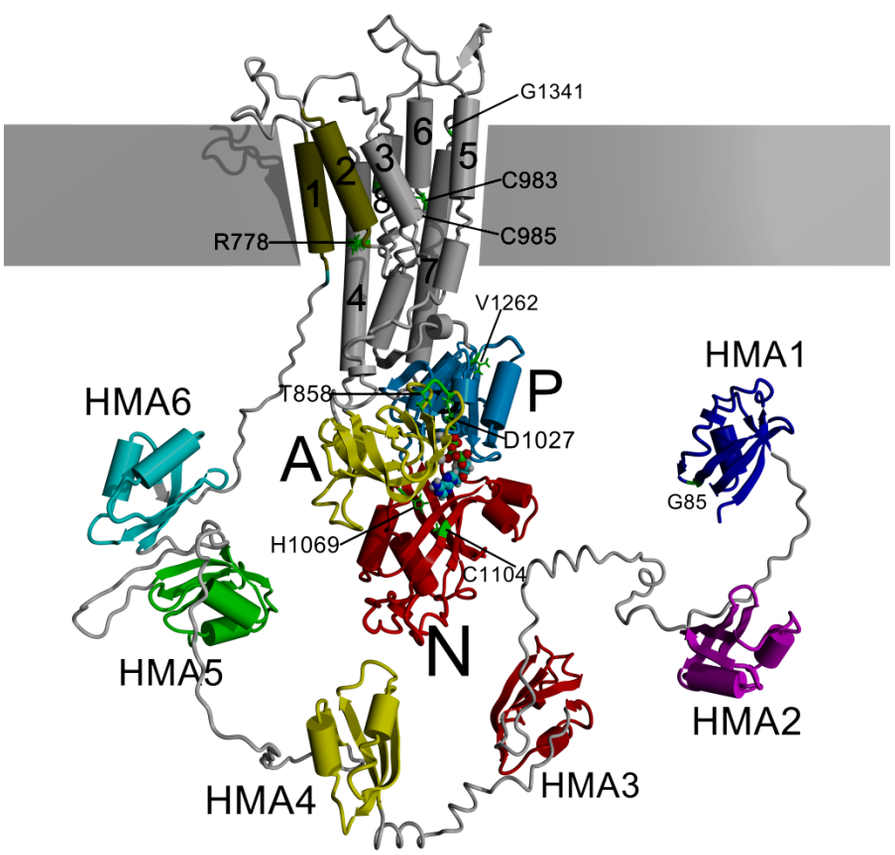
# Reduced expression of ATP7B affected by Wilson disease-causing mutations is rescued by pharmacological folding chaperones 4-phenylbutyrate and curcumin

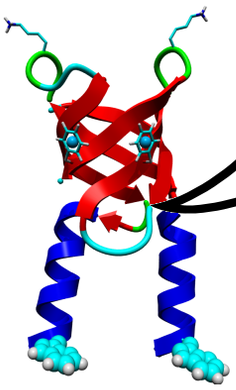
van den Berghe PV, Stapelbroek JM, Krieger E, de Bie P, van de Graaf SF, de Groot RE, van Beurden E, Spijker E, Houwen RH, Berger R, Klomp LW

*Hepatology.* **50(6),** 1783-95 (2009)

*Department of Metabolic and Endocrine Diseases, University Medical Center Utrecht, Utrecht, The Netherlands.*

Wilson disease (WD) is an autosomal recessive copper overload disorder of the liver and basal ganglia. WD is caused by mutations in the gene encoding ATP7B, a protein localized to the trans-Golgi network that primarily facilitates hepatic copper excretion. Current treatment comprises reduction of circulating copper by zinc supplementation or copper chelation. Despite treatment, a significant number of patients have neurological deterioration. The aim of this study was to investigate the possibility that defects arising from some WD mutations are ameliorated by drug treatment aimed at improvement of protein folding and restoration of protein function. This necessitated systematic characterization of the molecular consequences of distinct ATP7B missense mutations associated with WD. With the exception of p.S1363F, all mutations tested (p.G85V, p.R778L, p.H1069Q, p.C1104F, p.V1262F, p.G1343V, and p.S1363F) resulted in reduced ATP7B protein expression, whereas messenger RNA abundance was unaffected. Retention of mutant ATP7B in the endoplasmic reticulum, increased protein expression, and normalization of localization after culturing cells at 30 degrees C, and homology modeling suggested that these proteins were misfolded. Four distinct mutations exhibited residual copper export capacity, whereas other mutations resulted in complete disruption of copper export by ATP7B. Treatment with pharmacological chaperones 4-phenylbutyrate (4-PBA) and curcumin, a clinically approved compound, partially restored protein expression of most ATP7B mutants.





Yet Another Scientific ATPase Research Article was written with Lieke van der Velden and Stan van de Graaf at the Department of Metabolic and Endocrine Diseases, University Medical Center Utrecht, Netherlands. ATP8B1 is a P-type ATPase putatively transporting phosphatidylethanolamine and phosphatidylserine from one side of the membrane to the other. YASARA helped to analyze the structural basis of several ATP8B1 mutations linked to cholestasis, a disease where the digestion of lipids is impaired since bile/gall is not produced correctly in the liver and thus cannot transport the lipids to the duodenum for excretion.

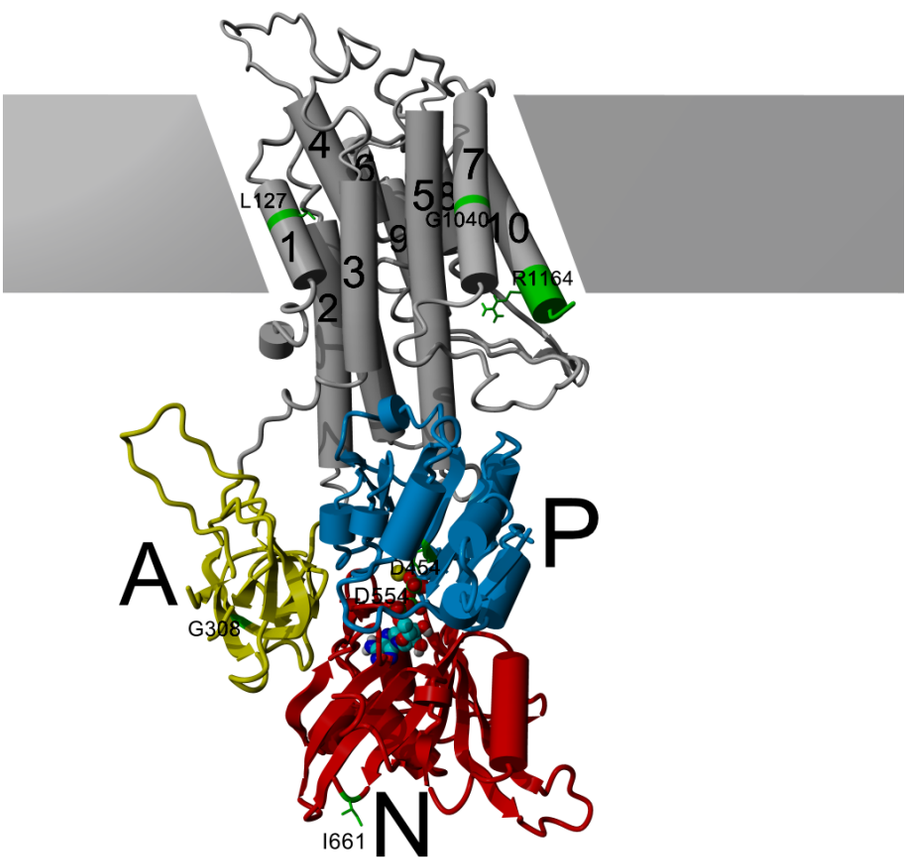
# Folding defects in P-type ATP8B1 associated with hereditary cholestasis are ameliorated by 4-phenylbutyrate

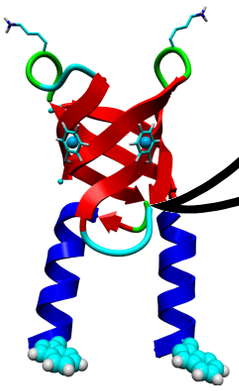
van der Velden LM, Stapelbroek JM, Krieger E, van den Berghe PV, Berger R, Verhulst PM, Holthuis JC, Houwen RH, Klomp LW, van de Graaf SF

*Hepatology.* **51(1)**, 286-96 (2010)

*Department of Metabolic and Endocrine Diseases, University Medical Center (UMC) Utrecht, The Netherlands.*

Deficiency in P-type ATP8B1 is a severe and clinically highly variable hereditary disorder that is primarily characterized by intrahepatic cholestasis. It presents either as a progressive (progressive familial intrahepatic cholestasis type 1 [PFIC1]) or intermittent (benign recurrent intrahepatic cholestasis type 1 [BRIC1]) disease. ATP8B1 deficiency is caused by autosomal recessive mutations in the gene encoding ATP8B1, a putative aminophospholipid-translocating P-type adenosine triphosphatase. The exact pathogenesis of the disease is elusive, and no effective pharmacological therapy is currently available. Here, the molecular consequences of six distinct ATP8B1 missense mutations (p.L127P, p.G308V, p.D454G, p.D554N, p.I661T, and p.G1040R) and one nonsense mutation (p.R1164X) associated with PFIC1 and/or BRIC1 were systematically characterized. Except for the p.L127P mutation, all mutations resulted in markedly reduced ATP8B1 protein expression, whereas messenger RNA expression was unaffected. Five of seven mutations resulted in (partial) retention of ATP8B1 in the endoplasmic reticulum. Reduced protein expression was partially restored by culturing the cells at 30 degrees C and by treatment with proteasomal inhibitors, indicating protein misfolding and subsequent proteosomal degradation. Protein misfolding was corroborated by predicting the consequences of most mutations onto a homology model of ATP8B1. Treatment with 4-phenylbutyrate, a clinically approved pharmacological chaperone, partially restored defects in expression and localization of ATP8B1 substitutions G308V, D454G, D554N, and in particular I661T, which is the most frequently identified mutation in BRIC1.





In contrast to the transmembrane ATPases described on the previous pages, the Drg1 protein from Saccharomyces cerevisiae belongs to ATPases Associated with various Activities, the AAA family, just like the human p97 protein. AAA-ATPases form hexameric rings and act as chaperones. The group of Helmut Bergler at the Institute of Molecular Biosciences, Karl-Franzens-University Graz, Austria, identified Drg1 as the target of diazaborine, an antimycotic that inhibits ribosome assembly. A homology model built with YASARA helped to analyze mutations known to induce resistance to diazaborine (hence the name Drg1, Diazaborine Resistance Gene 1).

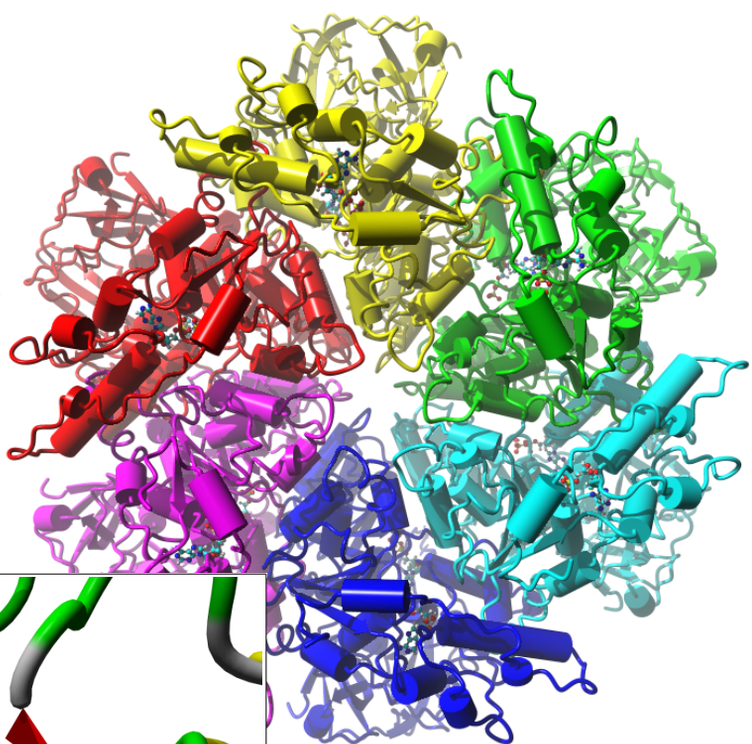
# The drug diazaborine blocks ribosome biogenesis by inhibiting the AAA-ATPase Drg1

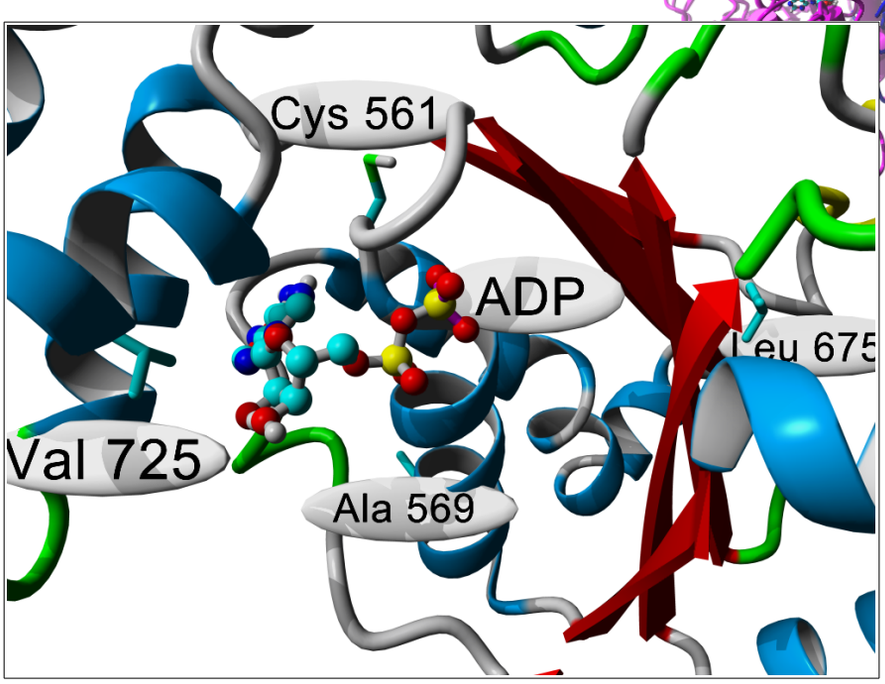
Loibl M, Klein I, Prattes M, Schmidt C, Kappel L, Zisser G, Gungl A, Krieger E, Pertschy B, Bergler H

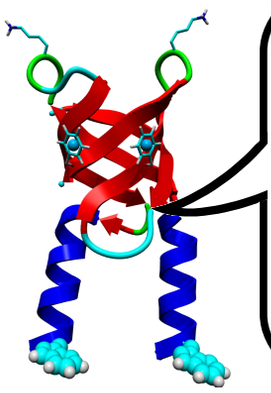
*J Biol Chem.* **289(7)**, 3913-22 (2014)

*Institut für Molekulare Biowissenschaften, Karl-Franzens-Universität Graz, A-8010 Graz, Austria*

The drug diazaborine is the only known inhibitor of ribosome biogenesis and specifically blocks large subunit formation in eukaryotic cells. However, the target of this drug and the mechanism of inhibition were unknown. Here we identify the AAA-ATPase Drg1 as a target of diazaborine. Inhibitor binding into the second AAA domain of Drg1 requires ATP loading and results in inhibition of ATP hydrolysis in this site. As a consequence the physiological activity of Drg1, i.e. the release of Rlp24 from pre-60S particles, is blocked, and further progression of cytoplasmic preribosome maturation is prevented. Our results identify the first target of an inhibitor of ribosome biogenesis and provide the mechanism of inhibition of a key step in large ribosomal subunit formation.







Leber congenital amaurosis is an inherited eye disease causing blindness due to the malfunction of photoreceptors in the retina. One protein affected is the photoreceptor RPGRIP1. Most known mutations can be found in two C2 domains, beta-sandwiches of eight strands which often bind calcium ions and help targeting proteins to cell membranes. In collaboration with Ronald Roepman at the Department of Human Genetics, Radboud University Nijmegen, Netherlands, YASARA helped to model the C-terminal C2 domain and found a potential calcium binding site whose disruption is likely to cause the disease by disabling the interaction with another protein named NPHP4.

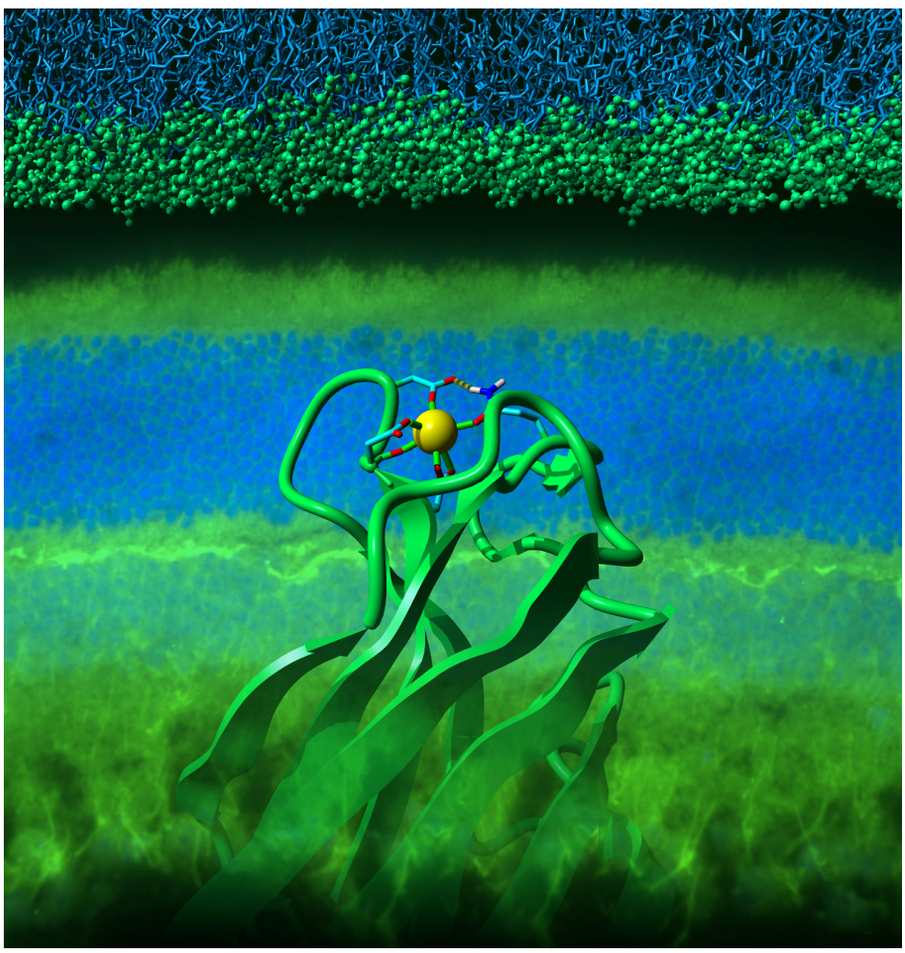
# Interaction of nephrocystin-4 and RPGRIP1 is disrupted by nephronophthisis or Leber congenital amaurosis-associated mutations

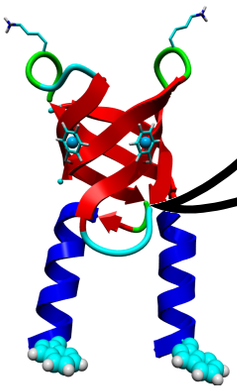
Roepman R, Letteboer SJ, Arts HH, van Beersum SE, Lu X, Krieger E, Ferreira PA, Cremers FP

*Proc Natl Acad Sci U S A.* **102(51),** 18520-5 (2005)

*Department of Human Genetics, Radboud University Nijmegen Medical Centre, P.O. Box 9101, 6500 HB, Nijmegen, The Netherlands*

RPGR-interacting protein 1 (RPGRIP1) is a key component of cone and rod photoreceptor cells, where it interacts with RPGR (retinitis pigmentosa GTPase regulator). Mutations in RPGRIP1 lead to autosomal recessive congenital blindness [Leber congenital amaurosis (LCA)]. Most LCA-associated missense mutations in RPGRIP1 are located in a segment that encodes two C2 domains. Based on the C2 domain of novel protein kinase C epsilon (PKC epsilon), we built a 3D-homology model for the C-terminal C2 domain of RPGRIP1. This model revealed a potential Ca2+-binding site that was predicted to be disrupted by a missense mutation in RPGRIP1, which was previously identified in an LCA patient. Through yeast two-hybrid screening of a retinal cDNA library, we found this C2 domain to specifically bind to nephrocystin-4, encoded by NPHP4. Mutations in NPHP4 are associated with nephronophthisis and a combination of nephronophthisis and retinitis pigmentosa called Senior-Løken syndrome (SLSN). We show that RPGRIP1 and nephrocystin-4 interact strongly in vitro and in vivo, and that they colocalize in the retina, matching the panretinal localization pattern of specific RPGRIP1 isoforms. Their interaction is disrupted by either mutations in RPGRIP1, found in patients with LCA, or by mutations in NPHP4, found in patients with nephronophthisis or SLSN. Thus, we provide evidence for the involvement of this disrupted interaction in the retinal dystrophy of both SLSN and LCA patients.





Another cause of Leber congenital amaurosis are mutations in CRB1 (homolog 1 of Drosophila's crumbs protein), which is likely at the core of a protein complex that controls the development of polarity in the eye. Another member of this complex is MPP5. Using two-hybrid screening, Albena Kantardzhieva at the Netherlands Ophthalmic Research Institute in Amsterdam could identify another member of the complex, MPP4, whose GUK domain binds the SH3 domain of MPP5. MPP4 and MPP5 are homologs, both have GUK and SH3 domains, which can either bind cis or trans. YASARA was used to estimate the binding energies of the four different MPP interactions types (MPP4*2, MPP5*2, MPP4-MPP5, MPP5-MPP4).

# MPP5 recruits MPP4 to the CRB1 complex in photoreceptors.

Kantardzhieva A, Gosens I, Alexeeva S, Punte IM, Versteeg I, Krieger E, Neefjes-Mol CA, den Hollander AI, Letteboer SJ, Klooster J, Cremers FP, Roepman R, Wijnholds J.

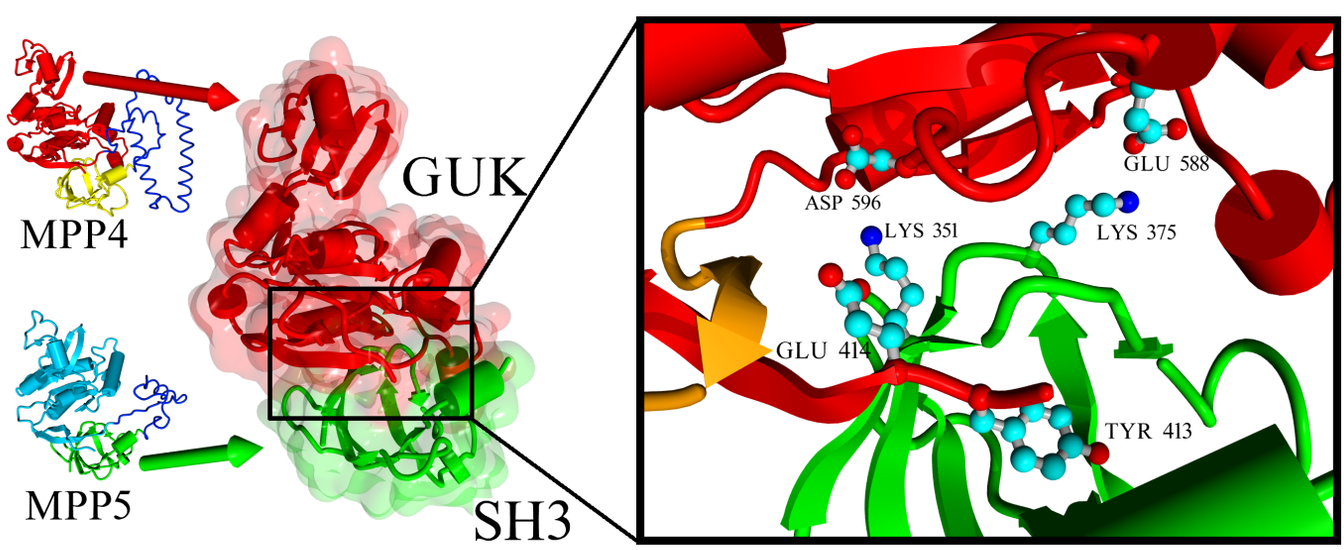
*Invest Ophthalmol Vis Sci.* **46**, 2192-201 (2005)

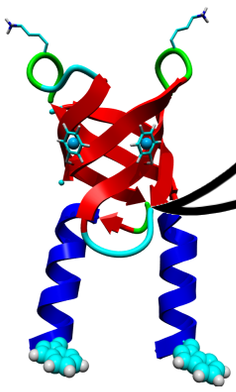
**PURPOSE:** Mutations in the human Crumbs homologue 1 (CRB1) gene are a frequent cause of Leber congenital amaurosis (LCA) and various forms of retinitis pigmentosa. CRB1 is thought to organize an intracellular protein scaffold in the retina that is involved in photoreceptor polarity. This study was focused on the identification, subcellular localization, and binding characteristics of a novel member of the protein scaffold connected to CRB1.

**METHODS:** To dissect the protein scaffold connected to CRB1, the yeast two-hybrid approach was used to screen for interacting proteins. Glutathione S-transferase (GST) pull-down analysis and immunoprecipitation were used to verify protein-protein interactions. The subcellular localization of the proteins was visualized by immunohistochemistry and confocal microscopy on human retinas and immunoelectron microscopy on mouse retinas.

**RESULTS:** A novel member of the scaffold connected to CRB1, called membrane palmitoylated protein (MPP) subfamily member 4 (MPP4), a membrane-associated guanylate kinase (MAGUK) protein, was identified. MPP4 was found to exist in a complex with CRB1 through direct interaction with the MPP subfamily member MPP5 (PALS1). 3D homology modeling provided evidence for a mechanism that regulates the recruitment of both homo- and heterodimers of MPP4 and -5 proteins to the complex. Localization studies in the retina showed that CRB1, MPP5, and MPP4 colocalize at the outer limiting membrane (OLM).

**CONCLUSIONS:** These data imply that MPP4 and -5 have a role in photoreceptor polarity and, by association with CRB1, pinpoint the cognate genes as functional candidate genes for inherited retinopathies.





In a continued collaboration project with Ilse Gosens and Ronald Roepman at the Department of Human Genetics, Radboud University Nijmegen Medical Centre, Netherlands, YASARA analyzed the dimerization potential of additional MPP family members. The Membrane Palmitoylated Proteins are membrane-associated guanylate kinase homologs (MAGUKs) that are involved in the positioning of neurons in the retina via the Crumbs complex, mutations can lead to retinal patterning defects. Yeast two-hybrid screening revealed MPP1as interaction partner of MPP5, modeling suggested that the dimerization occurs again via the SH3 and GUK domains, just like in the previous article.

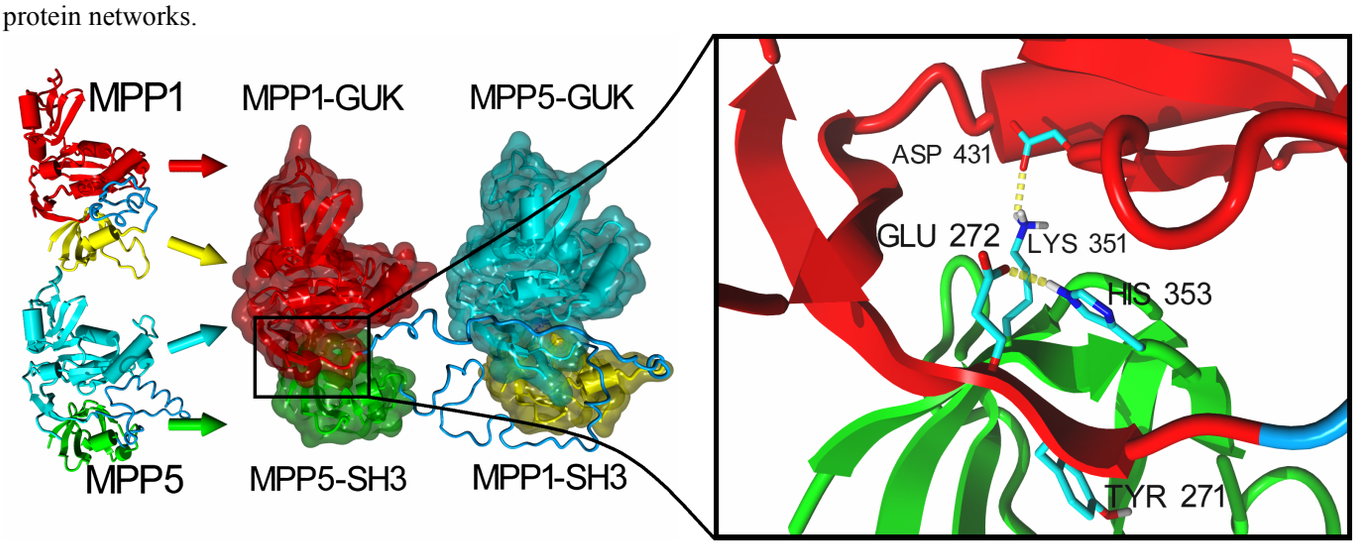
# MPP1 links the Usher protein network and the Crumbs protein complex in the retina

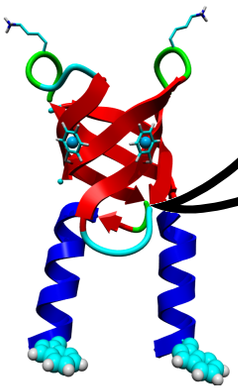
Gosens I, van Wijk E, Kersten FF, Krieger E, van der Zwaag B, Märker T, Letteboer SJ, Dusseljee S, Peters T, Spierenburg HA, Punte IM, Wolfrum U, Cremers FP, Kremer H, Roepman R

*Hum Mol Genet.* **16**,1993-2003 (2007)

*Department of Human Genetics, Radboud University Nijmegen Medical Centre, Nijmegen, The Netherlands.*

The highly ordered distribution of neurons is an essential feature of a functional mammalian retina. Disruptions in the apico-basal polarity complexes at the outer limiting membrane (OLM) of the retina are associated with retinal patterning defects in vertebrates. We have analyzed the binding repertoire of MPP5/Pals1, a key member of the apico-basal Crumbs polarity complex, that has functionally conserved counterparts in zebrafish (nagie oko) and Drosophila (Stardust). We show that MPP5 interacts with its MAGUK family member MPP1/p55 at the OLM. Mechanistically, this interaction involves heterodimerization of both MAGUK modules in a directional fashion. MPP1 expression in the retina throughout development resembles the expression of whirlin, a multi-PDZ scaffold protein and an important organizer in the Usher protein network. We demonstrate that both proteins interact strongly by both a classical PDZ domain-to-PDZ binding motif (PBM) mechanism, and a mechanism involving internal epitopes. MPP1 and whirlin colocalize in the retina at the OLM, at the outer synaptic layer and at the basal bodies and the ciliary axoneme. In view of the known roles of the Crumbs and Usher protein networks, our findings suggest a novel link of the core developmental processes of actin polymerization and establishment/maintenance of apico-basal cell polarity through MPP1. These processes, essential in neural development and patterning of the retina, may be disrupted in eye disorders that are associated with defects in these protein networks.





The vacuole is a membrane-confined cell compartment present mostly in plants and fungi and used mainly for waste disposal. In *S.cerevisiae*, a number of vacuolar sorting proteins (VPSs) are responsible for the transport between vacuole and Golgi. Five VPSs (5,17,26,29,35) form the retromer complex, which associates with the endosome, a special form of vacuole created during endocytosis. In collaboration with Ester Damen and Jeroen van Leeuwen at the Department of Cell Biology, Radboud University Nijmegen, the Netherlands, YASARA built a model of the Vps29 catalytic site and helped to select point mutants for further analysis.

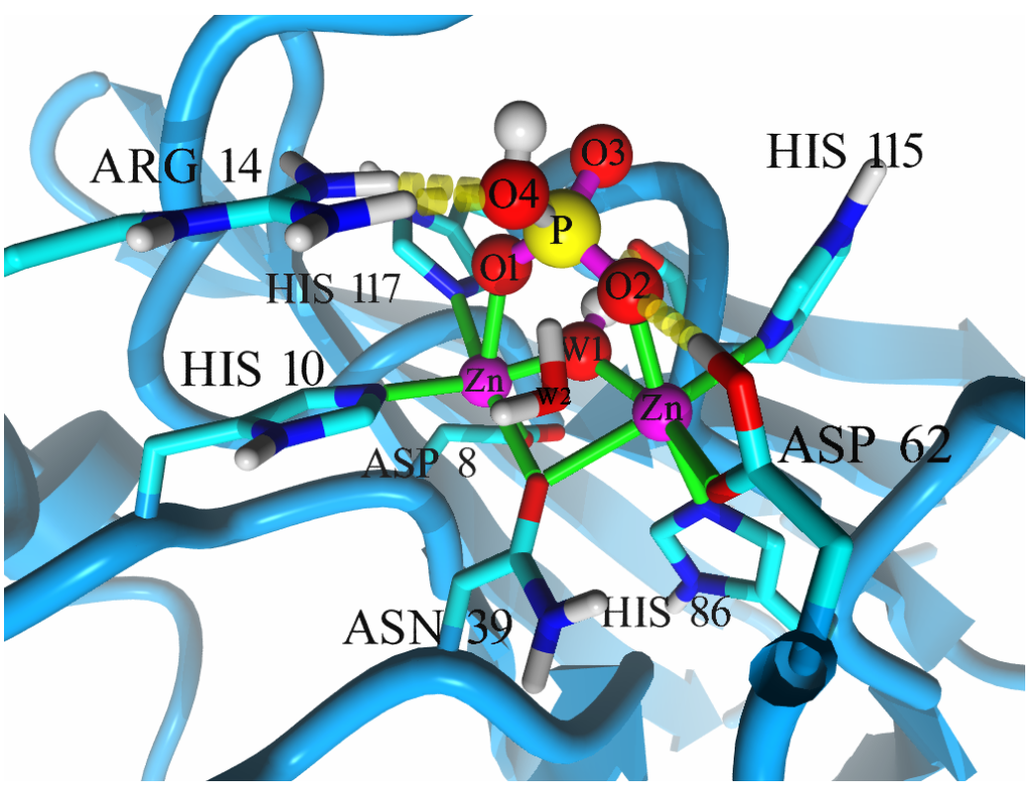
# The human Vps29 retromer component is a metallo-phosphoesterase for a cation-independent mannose 6-phosphate receptor substrate peptide

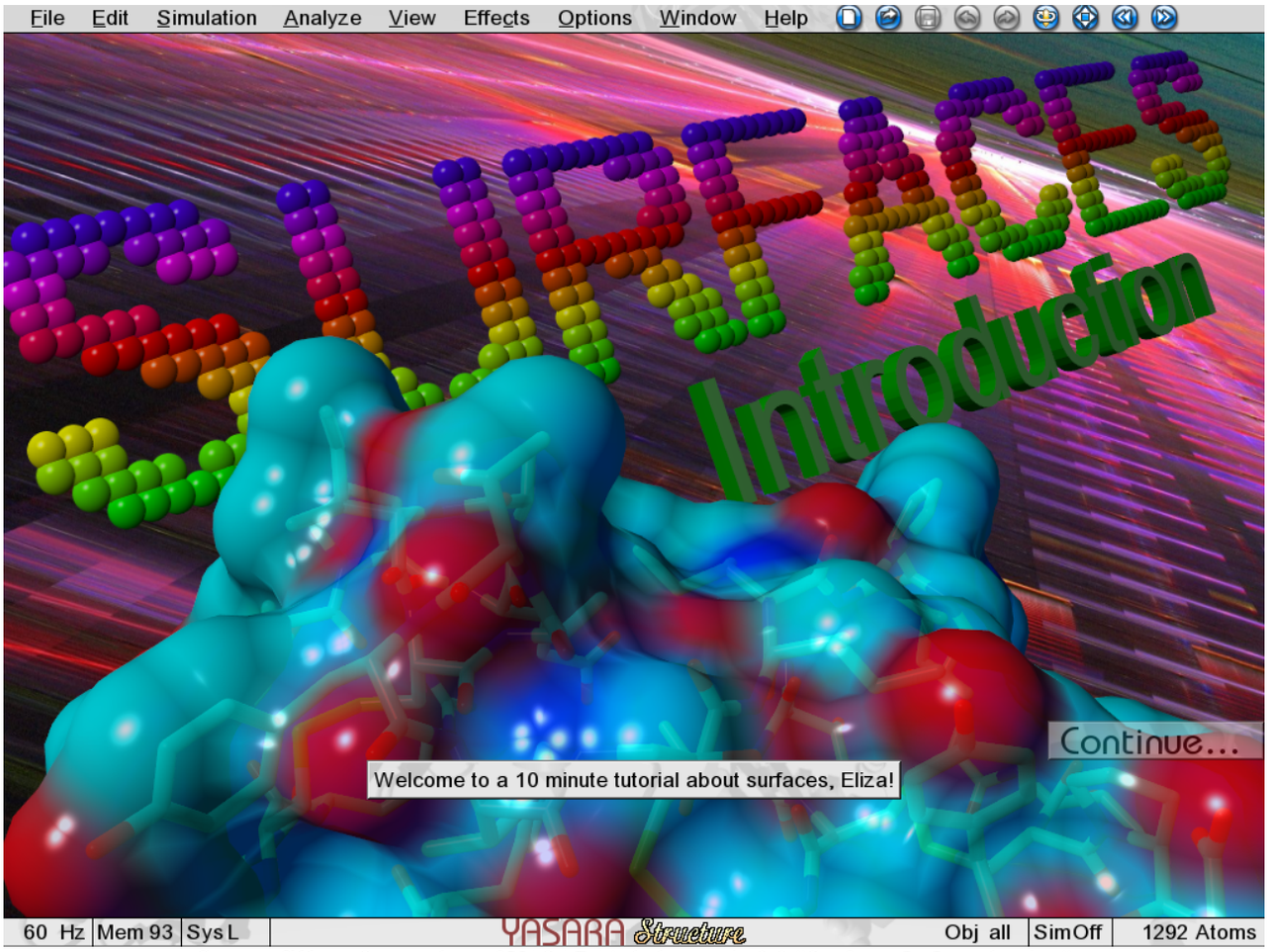
Damen E, Krieger E, Nielsen JE, Eygensteyn J, van Leeuwen JE

*Biochem J.* **398(3),** 399-409 (2006)

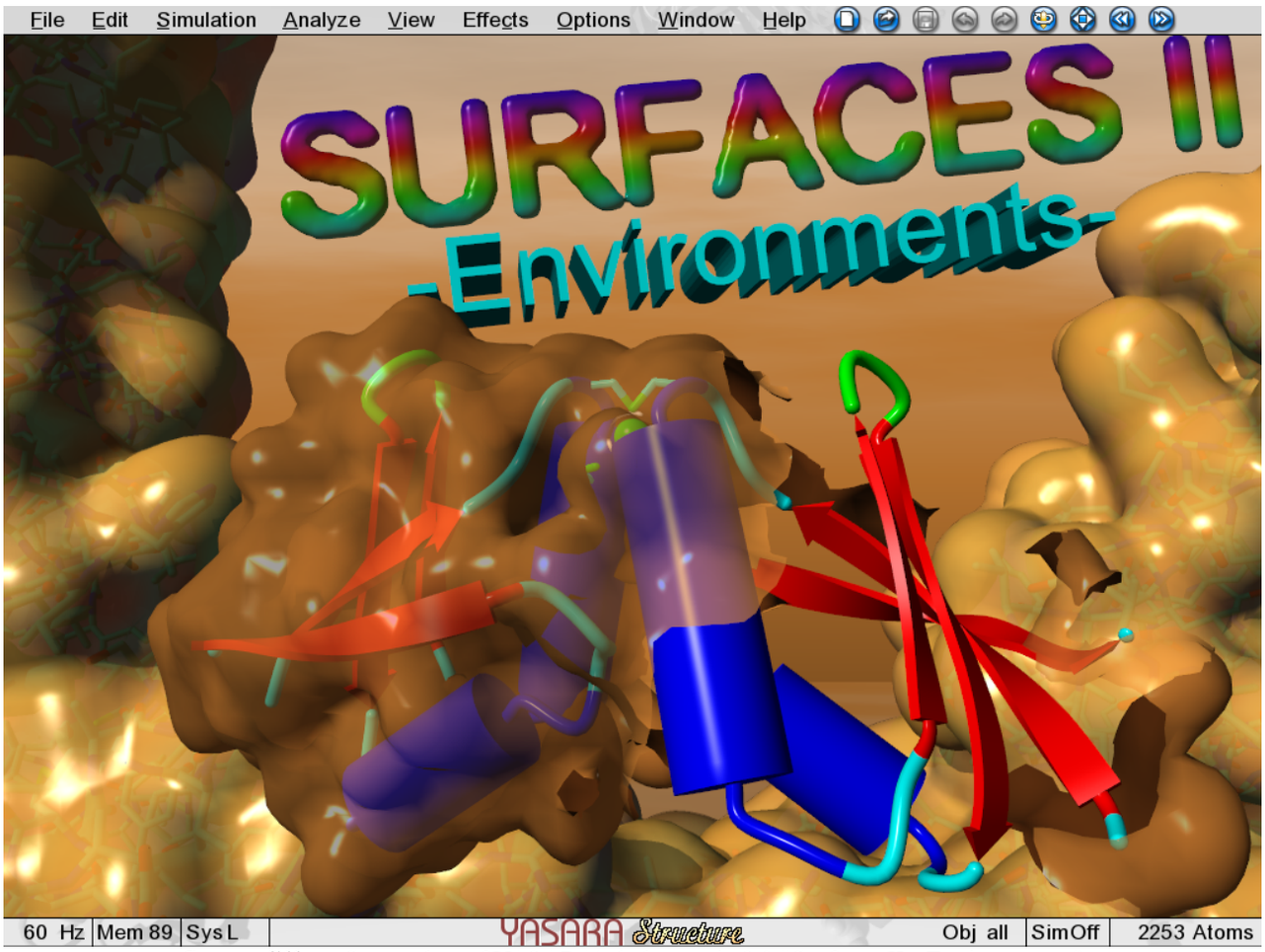
*Department of Cell Biology, Faculty of Sciences, Radboud University Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands.*

The retromer complex is involved in the retrograde transport of the CI-M6PR (cation-independent mannose 6-phosphate receptor) from endosomes to the Golgi. It is a hetero-trimeric complex composed of Vps26 (vacuolar sorting protein 26), Vps29 and Vps35 proteins, which are conserved in eukaryote evolution. Recently, elucidation of the crystal structure of Vps29 revealed that Vps29 contains a metallo-phosphoesterase fold [Wang, Guo, Liang, Fan, Zhu, Zang, Zhu, Li, Teng, Niu et al. (2005) J. Biol. Chem. 280, 22962-22967; Collins, Skinner, Watson, Seaman and Owen (2005) Nat. Struct. Mol. Biol. 12, 594-602]. We demonstrate that recombinant hVps29 (human Vps29) displays in vitro phosphatase activity towards a serine-phosphorylated peptide, containing the acidic-cluster dileucine motif of the cytoplasmatic tail of the CI-M6PR. Efficient dephosphorylation required the additional presence of recombinant hVps26 and hVps35 proteins, which interact with hVps29. Phosphatase activity of hVps29 was greatly decreased by alanine substitutions of active-site residues that are predicted to co-ordinate metal ions. Using inductively coupled plasma MS, we demonstrate that recombinant hVps29 binds zinc. Moreover, hVps29-dependent phosphatase activity is greatly reduced by non-specific and zinc-specific metal ion chelators, which can be completely restored by addition of excess ZnCl2. The binuclear Zn2+ centre and phosphate group were modelled into the hVps29 catalytic site and pKa calculations provided further insight into the molecular mechanisms of Vps29 phosphatase activity. We conclude that the retromer complex displays Vps29-dependent in vitro phosphatase activity towards a serinephosphorylated acidic-cluster dileucine motif that is involved in endosomal trafficking of the CI-M6PR. The potential significance of these findings with respect to regulation of transport of cycling trans-Golgi network proteins is discussed.

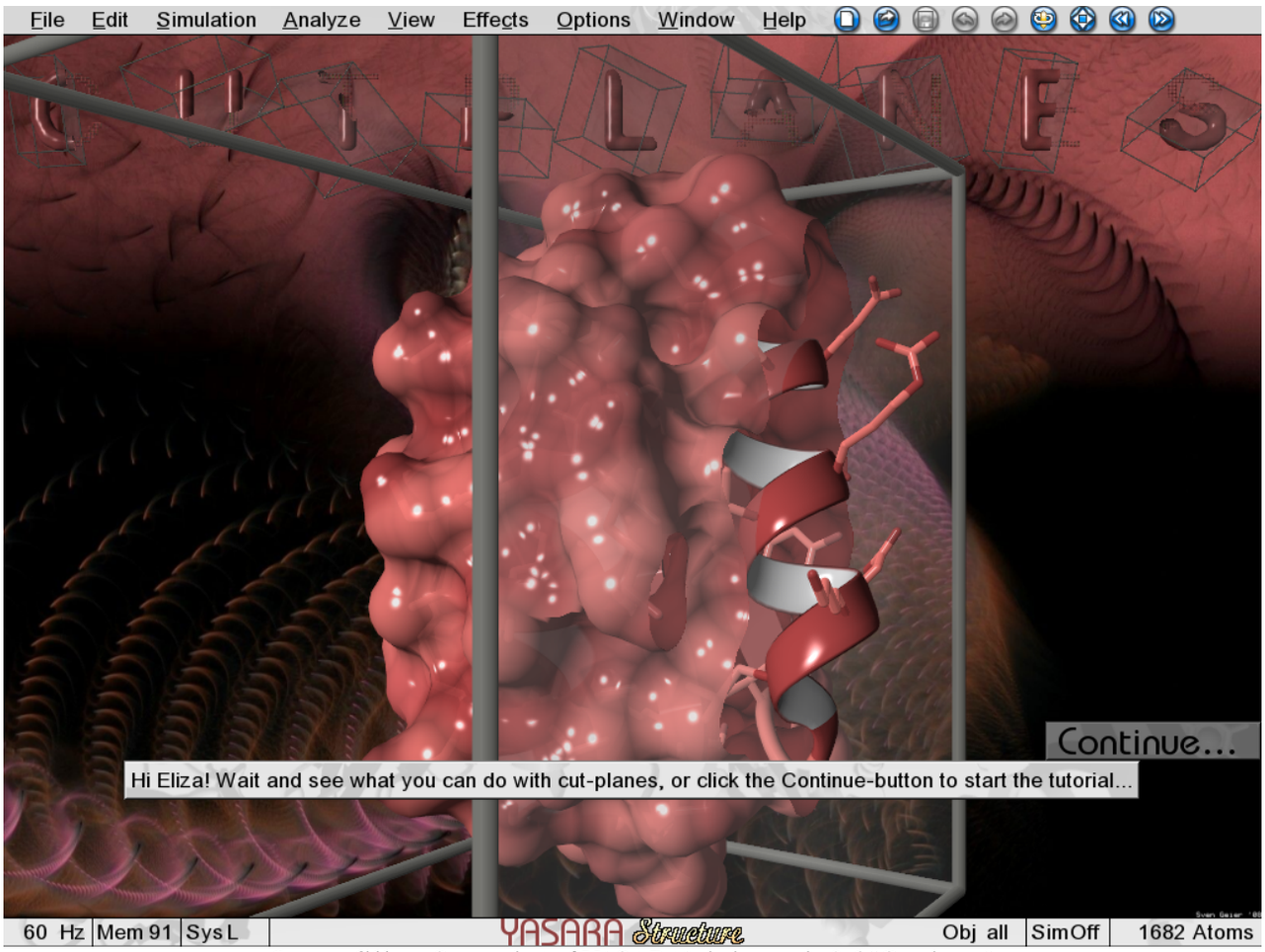




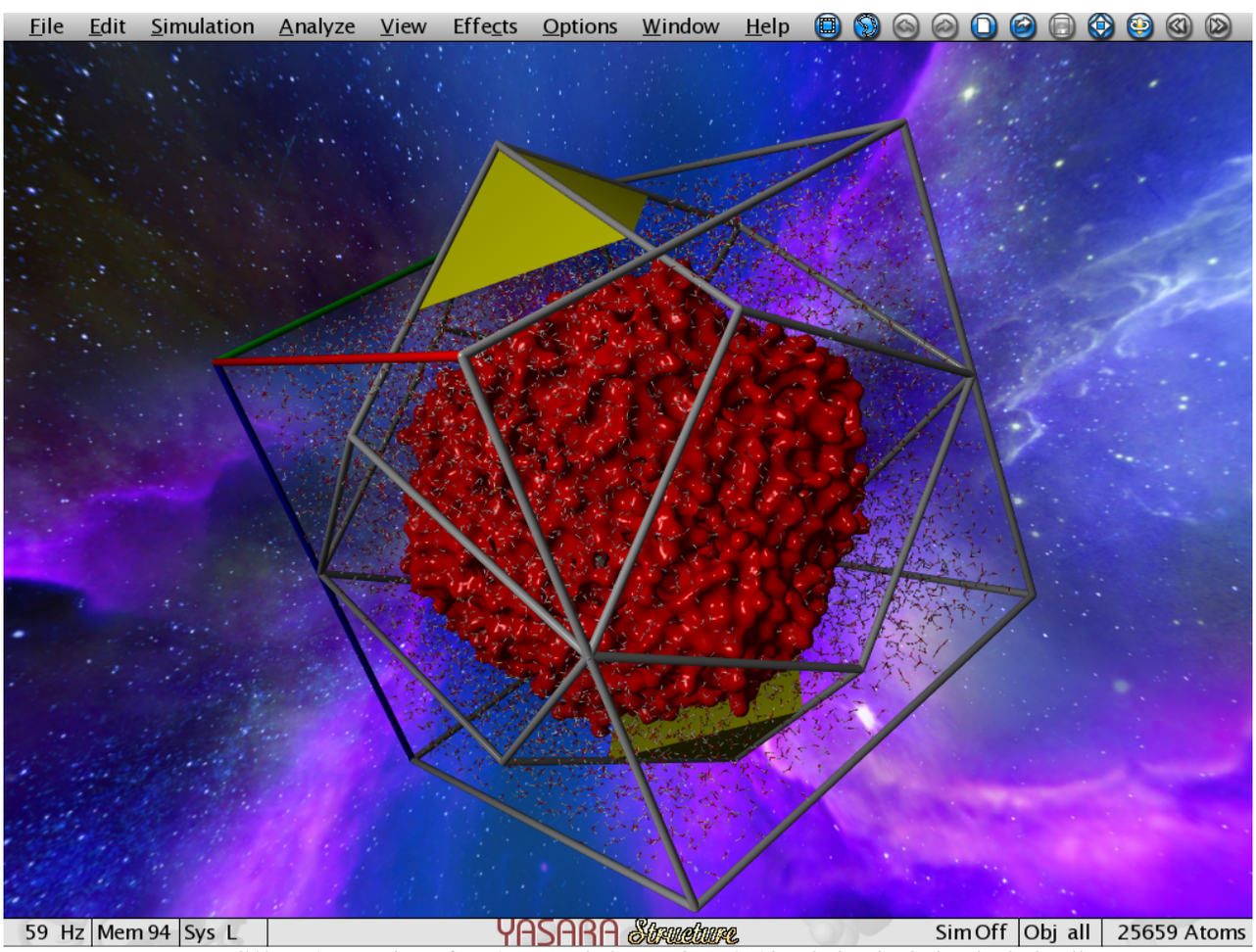

**Space filler:** Screenshot of YASARA's help movie 2.2 "Surfaces I - Introduction"




**Space filler:** Screenshot of YASARA's help movie 2.3 "Surfaces II - Environments"




**Space filler:** Screenshot of YASARA's help movie 2.6 "Cutplanes"




**Space filler:** Screenshot of YASARA's help movie 3.6 "Simulation in dodecahedral cells"