

# Basic Strategies for Molecular Docking with Scoring Functions

- ▶ New pharmaceutical targets are being discovered increasingly often
- ▶ Large databases of ligands are being screened for potential drugs
  - ▶ Many tools are being developed for this purpose, but...
  - ▶ Large quantities of data: a 7-dimensional search with simple rigid methods
  - ▶ Scoring functions for pose suitability are complex: generally  $O(n^2)$  at best

## Background & Motivation

Technological advances such as x-ray diffraction, nuclear magnetic resonance imaging, and high-performance graphical computing have made possible the 'in silico' modelling of molecular behaviour. The range of applications for these new tools is vast: the relevance extends at least to the biochemical, medical, and physical sciences.<sup>1</sup>

An important area for research has been that of molecular interactions. Given two molecular structures, it should be possible to quantifiably predict the relative effects they have on each other. In some cases, two molecules might 'bind' to form a stable, equilibrated attachment.

Computational drug discovery uses simulations of large biological molecules ('receptors', often proteins) and small 'ligand' compounds to estimate whether a binding affinity exists. If so, that ligand may be a basis for a drug that inhibits the effects of the target receptor. The process of juxtaposing a ligand and receptor, evaluating this 'pose', and attempting to find a better arrangement is known as 'docking'. Typically, the method is iterated over a large database of potential ligands for any given target, 'screening' out a tractable list for manual investigation.

As more target receptors involved in the pathology of diseases are identified using the improving technology available to medicinal chemists, ligand databases will be screened increasingly frequently, searching for cures.

## Project Context

This project has been supported by InhibOx, Ltd.<sup>2</sup> since 2005, and has used their protein-ligand docking tool *DOx* as a basis for research and development. This uses a genetic algorithm to generate poses of each ligand, and empirical scoring functions to evaluate them. The functions take the form

$$S = \sum_l \sum_r \sum_i F(l, r, i)$$

where  $l$  and  $r$  are ligand and receptor atoms respectively, and  $i$  iterates over the function's interaction types (such as electrostatics, hydrogen bonding, hydrophobic effects, etc. Ligand flexibility is modelled by the use of conformation databases, treating each conformation as a distinct, rigid case.

A series of techniques is being tested, separately at first and later in concert, all building on the basic *DOx* system. The speed and result quality (RMSD: root-mean-square deviations from the correct pose) are being measured to establish an ideal strategy for docking.

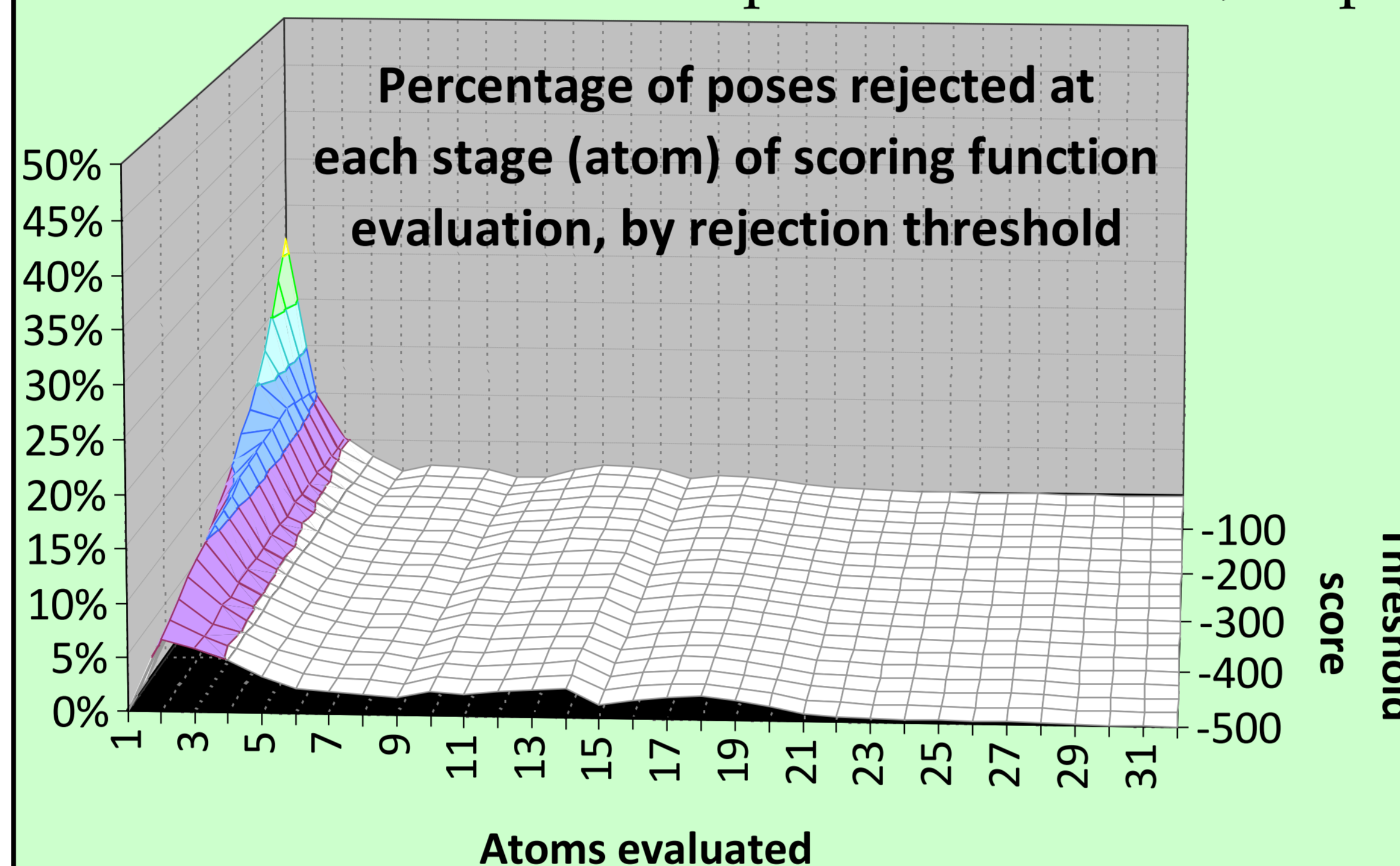
Our most recent work is now redesigning the entire system into a networked parallel execution model, whilst also making the components of the docking process easily switchable.

## Searching and Scoring

- ▶ Common features:
  - ▶ Look-up tables (LUTs) for receptor data
  - ▶ Interpolation of LUT data
- ▶ Additional methods:
  - ▶ Local optimization of results
    - Slow, but improves results
  - ▶ Cache LUT data
    - Avoids full pre-calculation
    - Counter-productive for practical docking
    - Interesting for analysis
  - ▶ Drop duplicate search cases

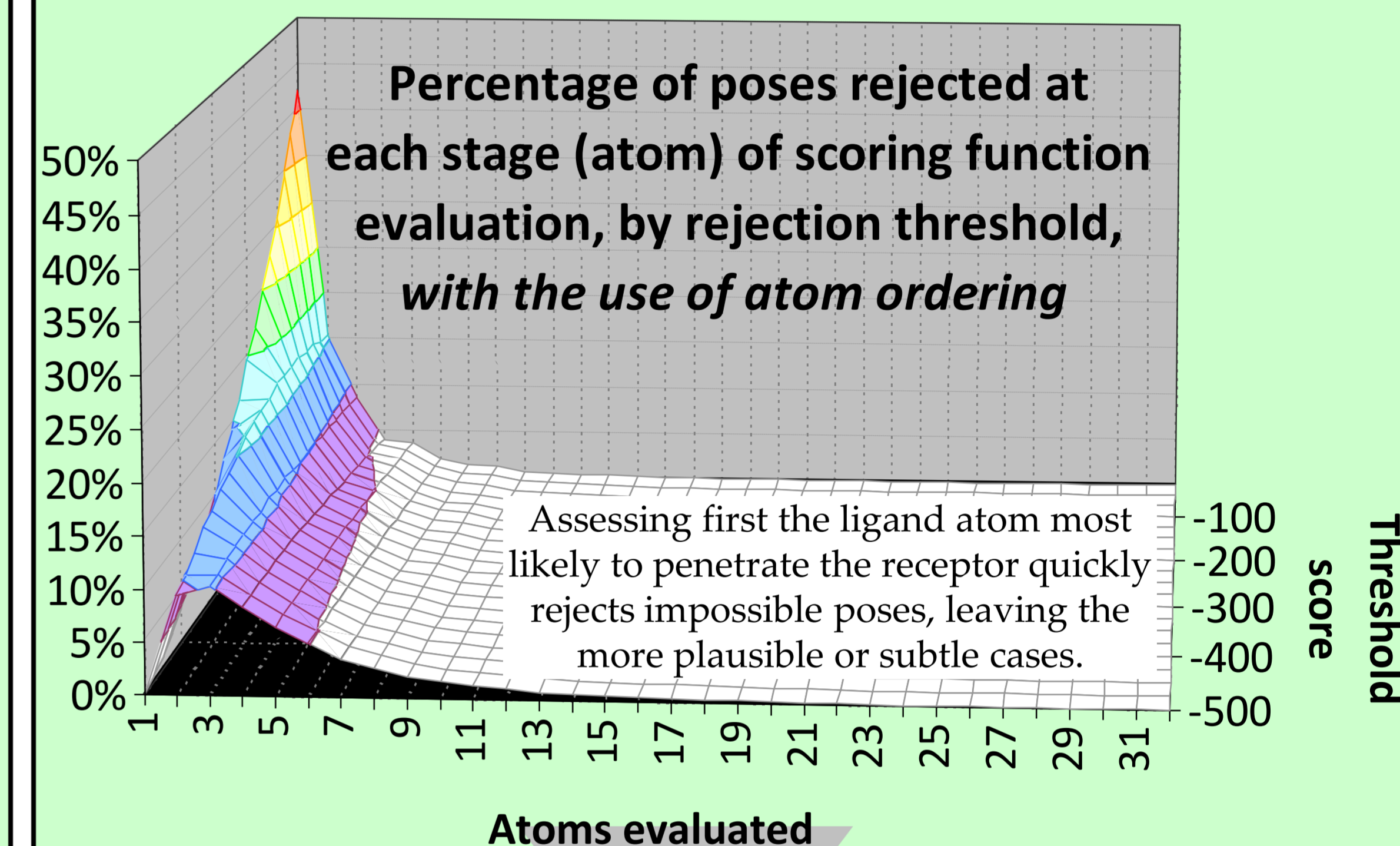
## Early Rejection

If a pose is bad, it doesn't matter how bad. After each atom's contribution is added to the pose's score, check whether it is so poor that it cannot return to an acceptable level. If so, stop.



## Prioritization

A little forethought shrinks the task ahead. If a pose can be rejected easily when its score passes a threshold, sort the atoms such that less desirable contributions will be added first.



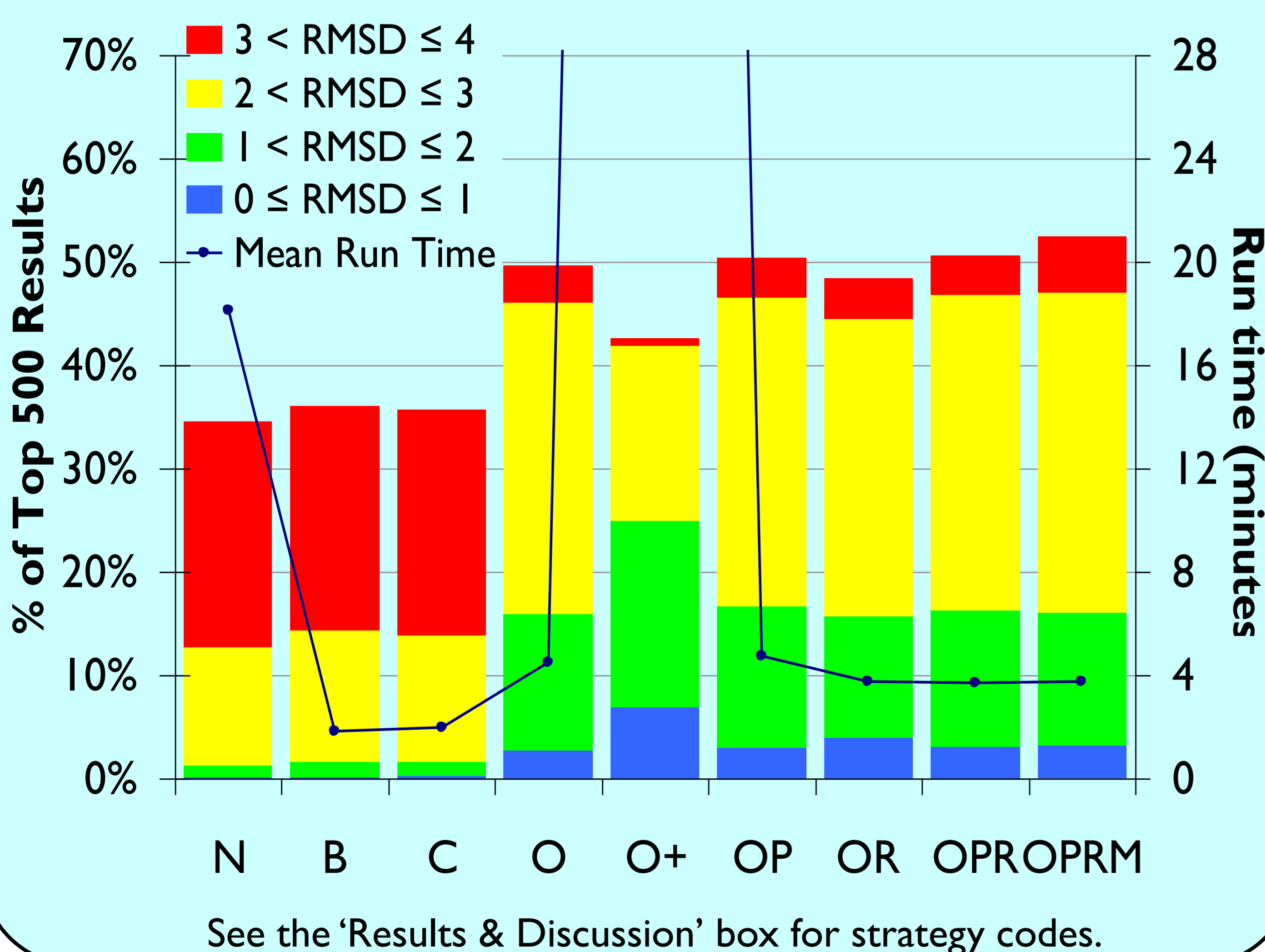
## Pure computer science techniques improve speed and accuracy

### Comparison of Strategies

Separating search into global and local stages for discovery and refinement is useful, but slow if speed is not balanced against efficacy.

Early rejection is a valuable strategy: reductions in run time of at least 20% are possible without loss of results. Analyzing the task may help even more.

Test case: Re-docking the crystal structure of PDB code 1AF2<sup>3</sup> (as shown in the image<sup>4</sup> above) with an ensemble of 52 conformations using the XScore function.



### Results & Discussion

The basic *DOx* system has been adapted into several editions, each applying a different strategy to the docking task. The comparison graphs use codes to identify these strategies: None (no indexing or interpolation), Basic (original version), LUT Caching, local Optimizations, Prioritization, early Rejection, and duplicate case Merging.

It can be seen that the introduction of the local optimization searches does markedly improve the quality of results, but at a significant speed cost. Also, it seems to be of limited use, since the 'O+' results (optimizing every genome of every generation) appeared worse than the simpler 'O' case (only optimizing the final result set) despite the significantly greater run time.

As should be expected, the prioritization (ordering) of atoms for scoring alone made no difference to results, and attracted only a slight time cost. However, the use of early rejection delivered an approximately 47% reduction in run time, with comparable results. On the small molecule used for these tests, and with the very low (frequently-reached) cut-off score, ordering appears to make relatively little difference. Other tests suggest that prioritization is of more benefit when longer or larger ligands are used with a more relaxed rejection threshold.

A follow-up investigation trialled a constraint on rejection: poses that were not rejected after the first  $n$  atoms would no longer be considered for rejection, and scored completely without distraction. This was found to be slightly detrimental to run times, surprisingly; one explanation might be that it defeated compiler loop optimizations.

Duplicate genome merging was intended to make the GA search more efficient. Although up to 20% of cases were discarded for being too similar to others already tried (depending on parameter), no great difference was seen in the results.

#### Docking Tools & Scoring Functions

Tools: AutoDock, DOCK, DockIt, FlexX/E, Flo, FRED, GLIDE, GOLD, LigFit, MOE, MVP

Functions: AutoDock, ChemScore, DrugScore, D-Score, F-Score, G-Score, LigScore, LUDI, PLP, PMF, XScore

These lists are not exhaustive. Several review papers exist, wherein individual references may be found.<sup>5,6</sup>

#### References

1. Skone & Cameron. *Proc. FBIT*. 2007
2. InhibOx Ltd. [www.inhibox.com](http://www.inhibox.com)
3. Xiang, Short, Wolfenden, & Carter. *Biochem.* 1997:36
4. DeLano Scientific LLC. PyMOL, 2006
5. Wang, Lu, & Wang. *J. Med. Chem.* 2003:46
6. Warren et al. *J. Med. Chem.* 2006:49
7. Finn et al. (InhibOx) DrugFinder. *In prep.*

### Future Work

- ▶ Deferred evaluation
- ▶ Parallelization
- ▶ Learning from...
  - ▶ Shape & structure
  - ▶ Previous dockings
- ▶ Modular code design

This development from a single-purpose GA search program into an adaptable docking tool will apply techniques from machine learning, computational geometry, and concurrent processing to maximize efficiency with methods that can be applied across all such tools.

