ScalaLife

Competence Center: Providing tailored support to the Life Science Computational Communities

CAPACITIES

e-infrastructure

ScalaLife Project and Partners

Project started in September 2010

Leibniz-Rechenzentrum, Germany

Oxford e-Research Center, United Kingdom

Synective Labs, Sweden

Royal Institute of Technology, Sweden

PDC Center for High Performance Computing at KTH, Sweden

Institute for Research in BioMedicine, Spain

Barcelona Supercomputing Center, Spain

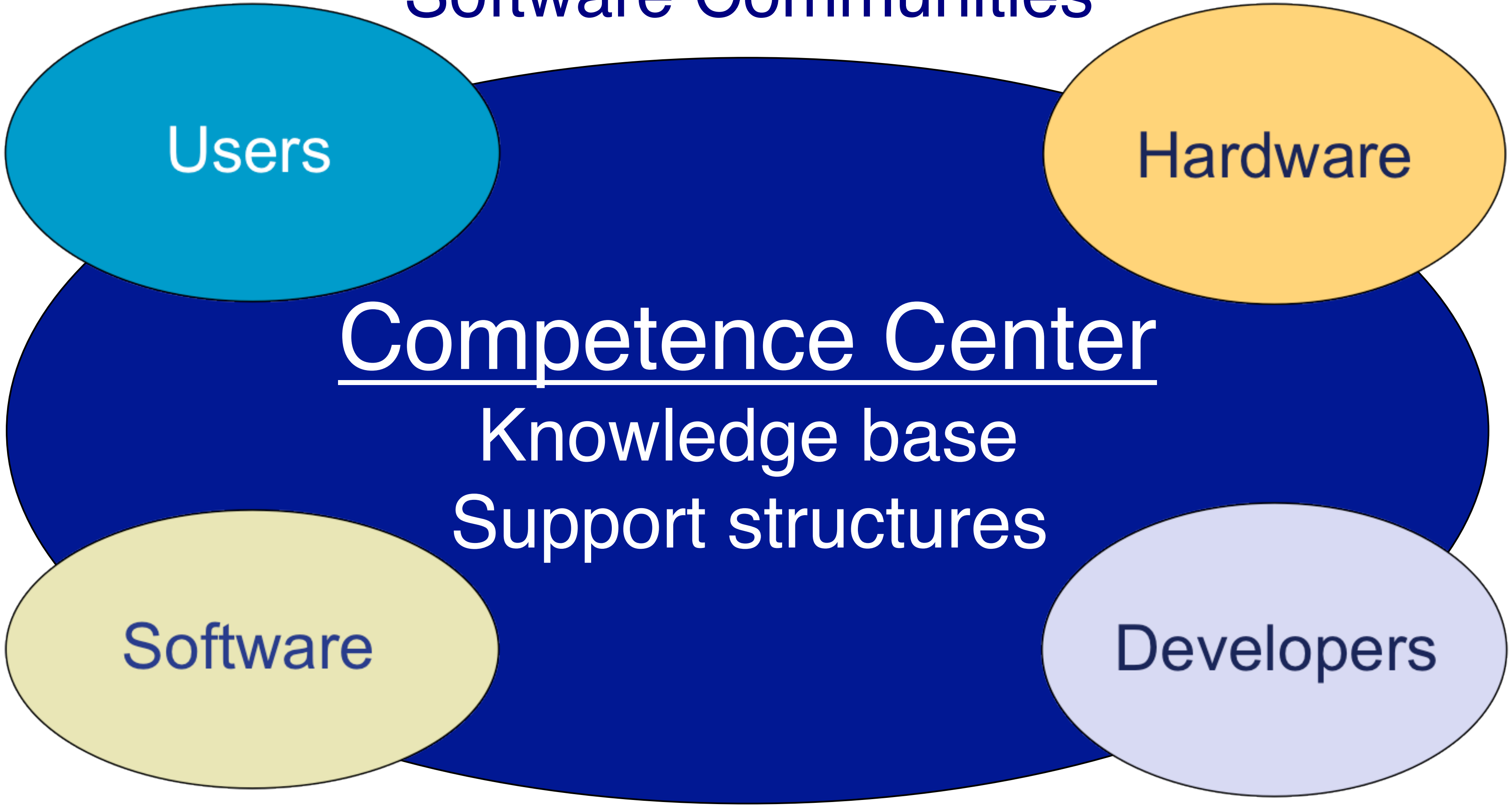
ScalaLife Objectives

- Hierarchical **parallelization** & **ensemble** computing
- **Pilot codes**: DALTON (QM), GROMACS (MD), DISCRETE (CG)
- Standards for **file formats**

- Capture and document **best practices**
- Establish a **Competence Center** for computational Life Science

Competence Center

"**One-stop-shop**" for Life Science Software Communities



Users

Hardware

Competence Center

Knowledge base

Support structures

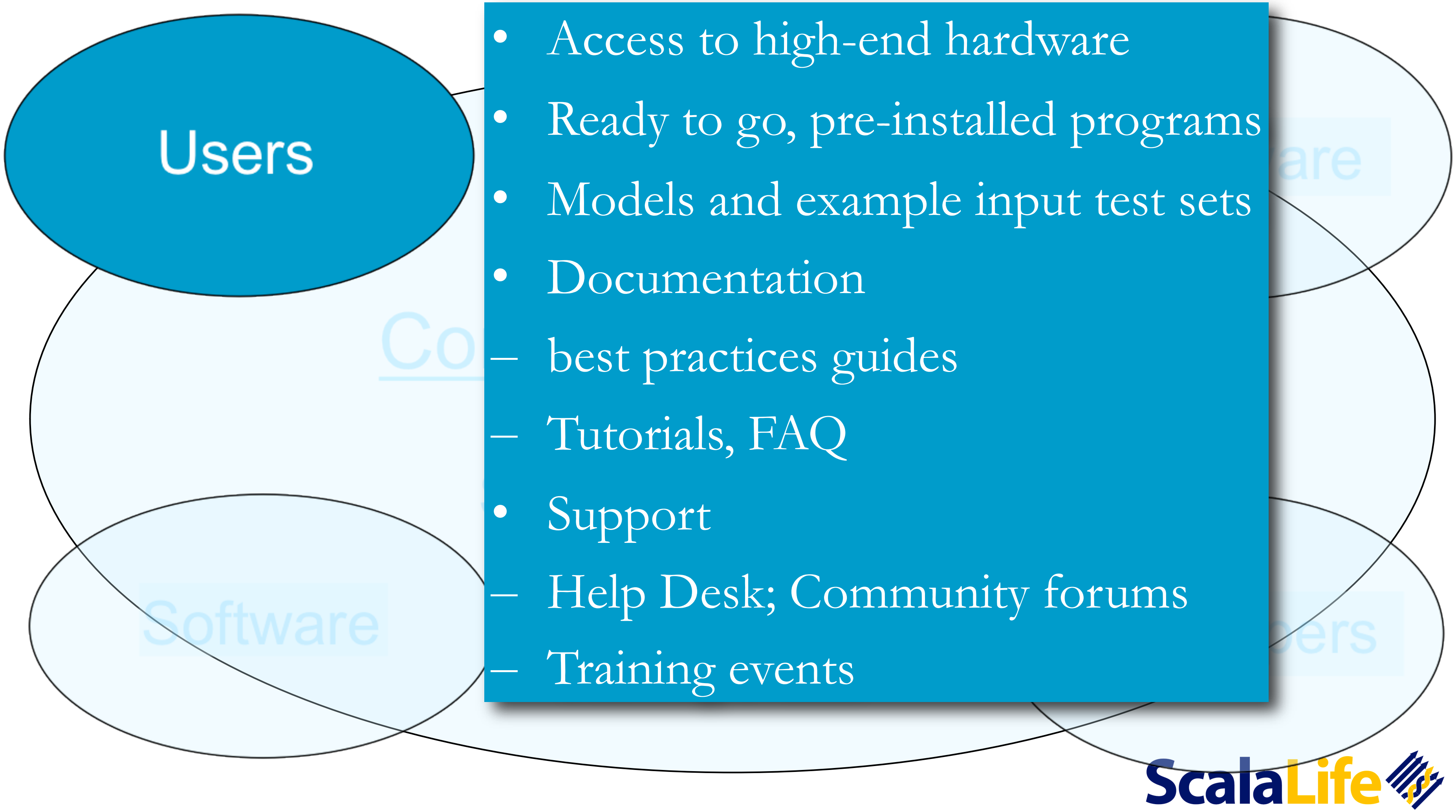
Software

Developers

www.ScalaLife.eu

Competence Center

"**One-stop-shop**" for Life Science Software Communities

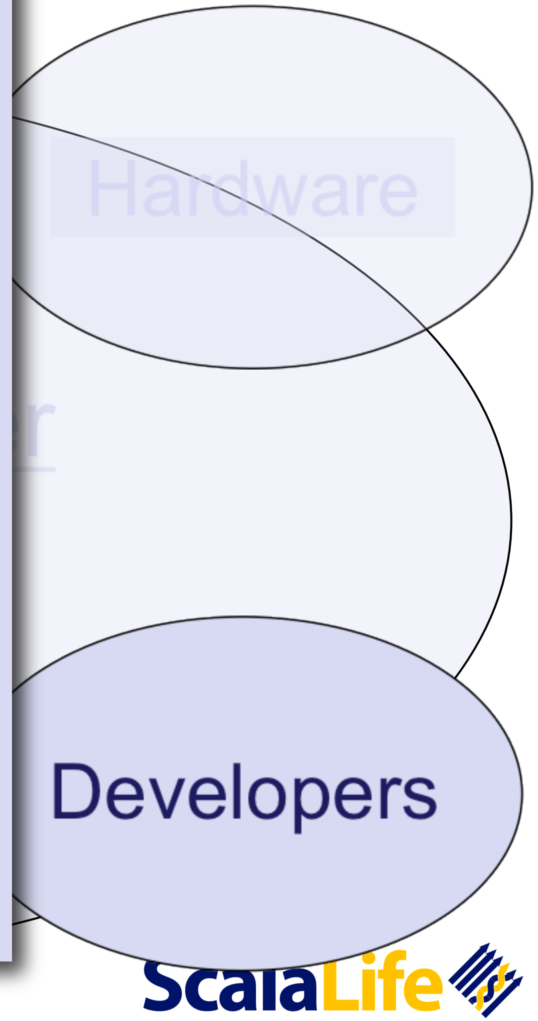


- Access to high-end hardware
- Ready to go, pre-installed programs
- Models and example input test sets
- Documentation
  – best practices guides
  – Tutorials, FAQ
- Support
  – Help Desk; Community forums
  – Training events

Competence Center

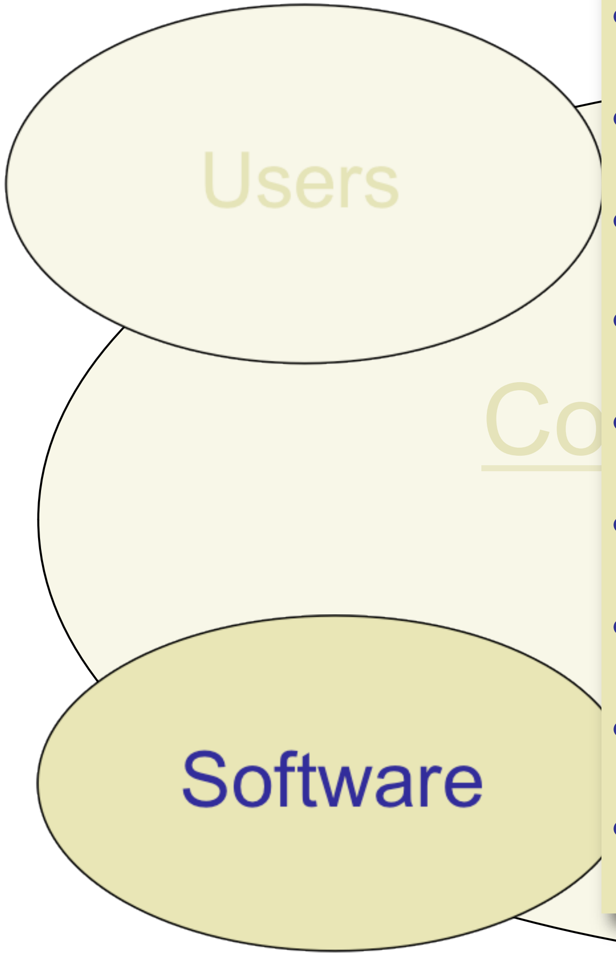
“**One-stop-shop**” for Life Science

- Applications
  - Latest source code releases
  - test sets; best programming practices guides; performance reports on different architectures
- Development tools: Debuggers; Performance Tools etc.
  - Documentation
- Algorithms: descriptions, success stories, implementations
  - Performance analysis



Competence Center

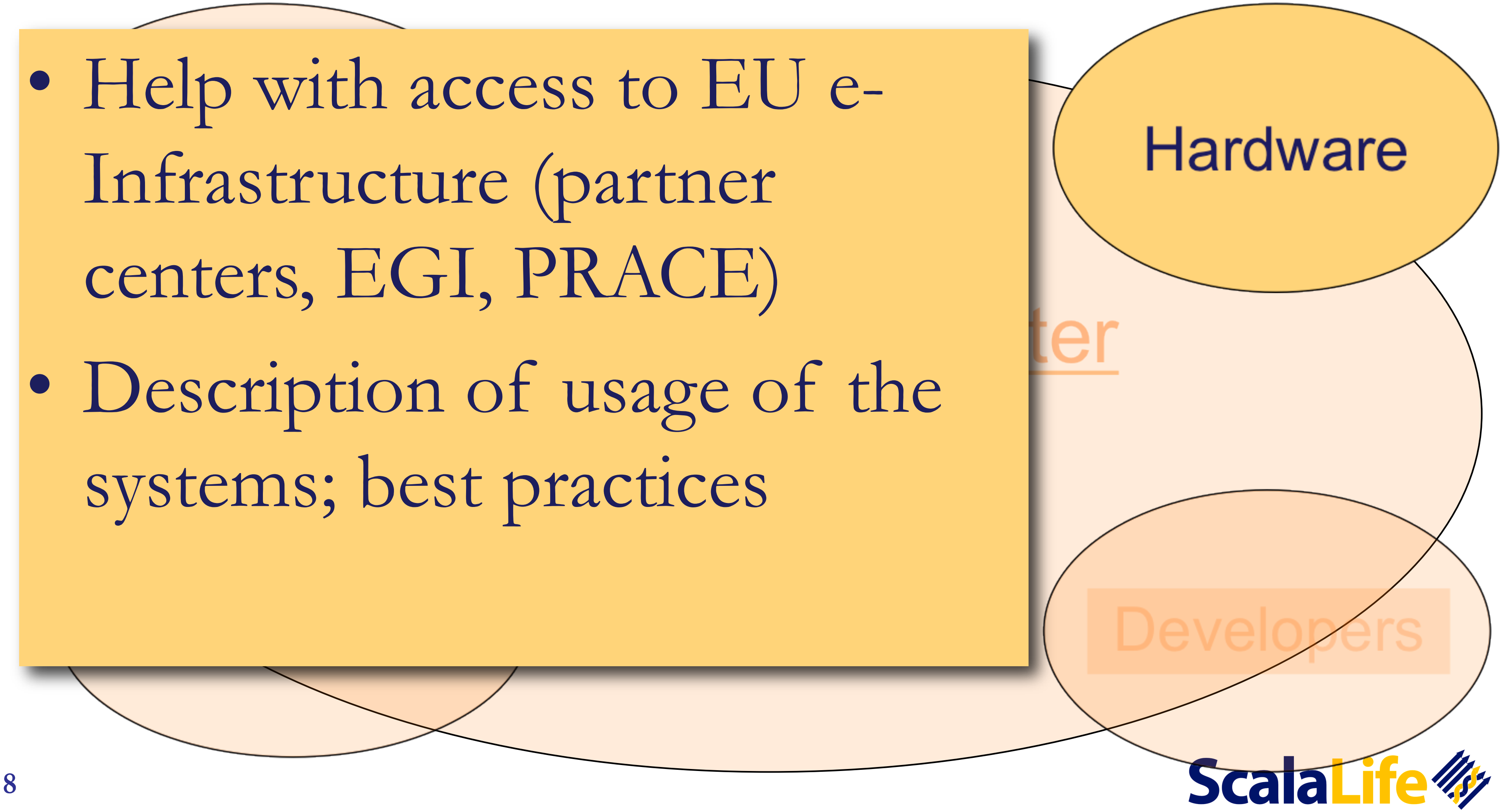
"**One-stop-shop**" for Life Science Software Communities



- Pilot applications:
- GROMACS (MD)
- DALTON (QM)
- DISCRETE (DMD)
- New applications
- ERGO (QM)
- MUSIC (neuroinformatics)
- XMIPP (medical imaging)...
- Development tools

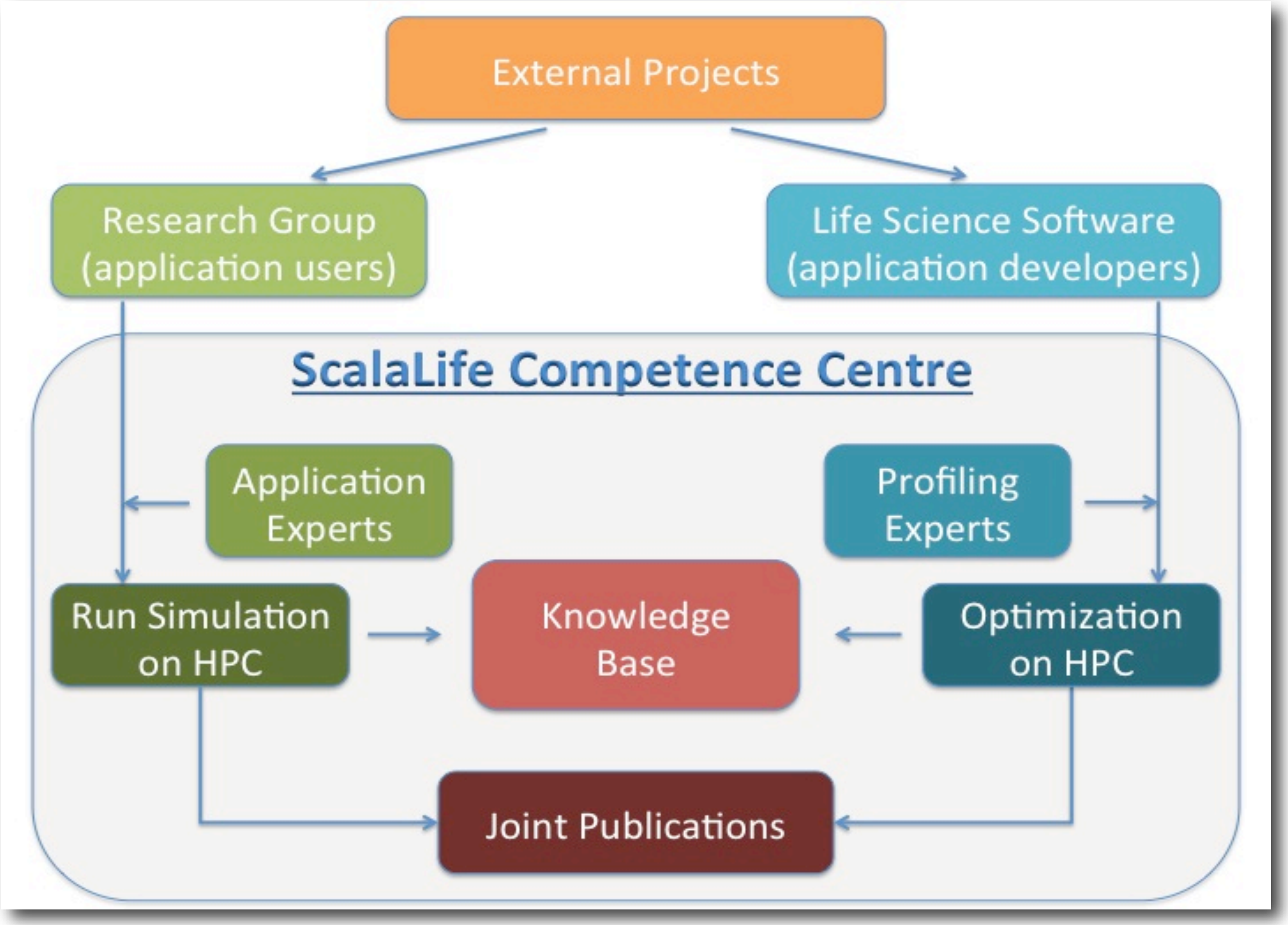
Competence Center

"**One-stop-shop**" for Life Science Software Communities



- Help with access to EU e-Infrastructure (partner centers, EGI, PRACE)
- Description of usage of the systems; best practices

Competence Centre Structure



External Projects

Research Group (application users)

Life Science Software (application developers)

ScalaLife Competence Centre

Application Experts

Profiling Experts

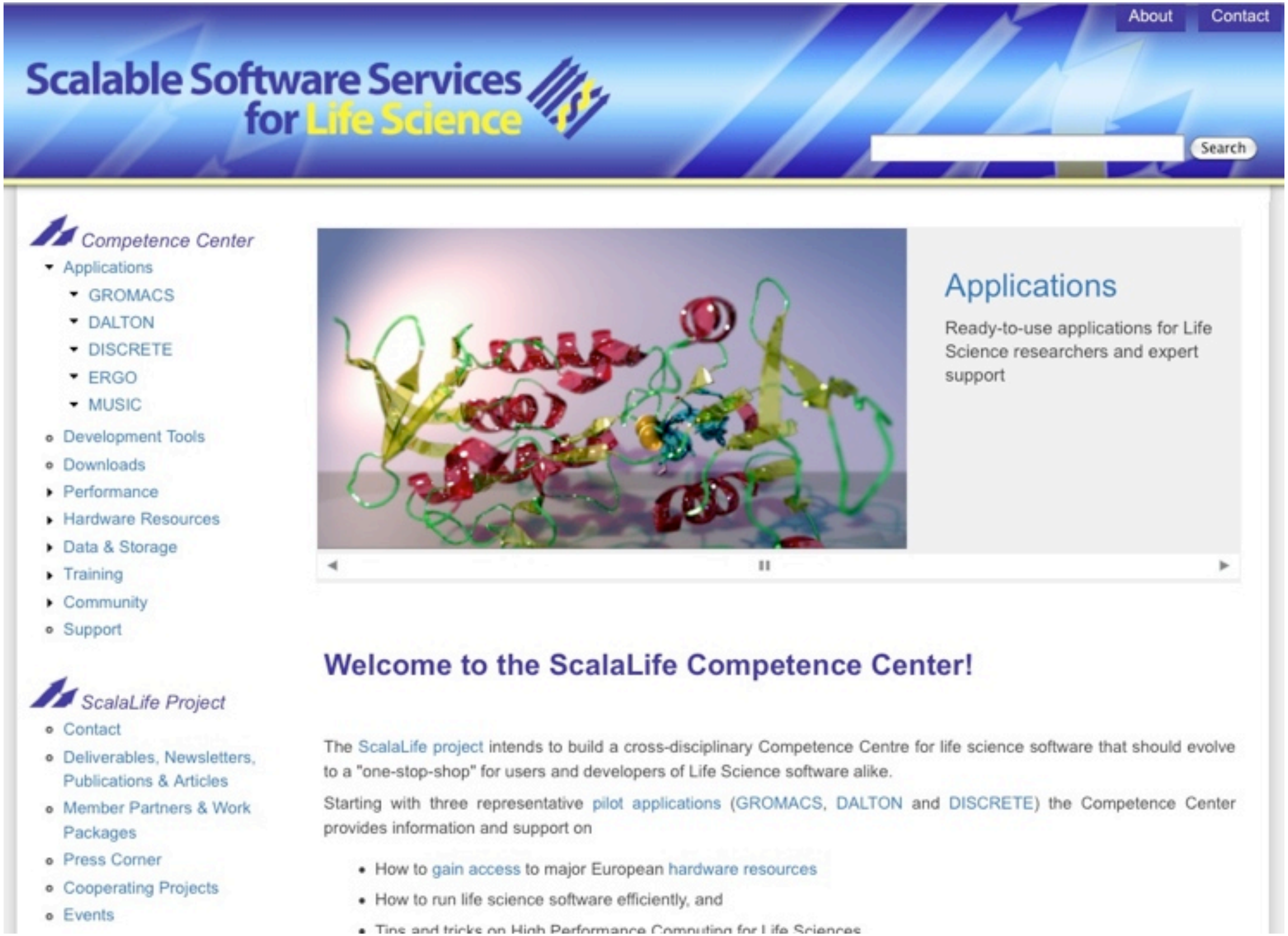
Run Simulation on HPC

Knowledge Base

Optimization on HPC

Joint Publications

www.ScalaLife.eu



Applications

Ready-to-use applications for Life Science researchers and expert support

Welcome to the ScalaLife Competence Center!

The ScalaLife project intends to build a cross-disciplinary Competence Centre for life science software that should evolve to a "one-stop-shop" for users and developers of Life Science software alike.

Starting with three representative pilot applications (GROMACS, DALTON and DISCRETE) the Competence Center provides information and support on

- How to gain access to major European hardware resources
- How to run life science software efficiently, and
- Tips and tricks on High Performance Computing for Life Sciences

SCALALIFE MAIN CODES:

DALTON: electronic structure of molecules
using quantum mechanics calculations, QM/MM

GROMACS: molecular dynamics simulations (classical mechanics)

DISCRETE: discrete molecular dynamics (DMD) (classical mechanics)

DISCRETE:
DMD SIMULATIONS OF PROTEINS AND PROTEIN COMPLEXES

Developers: *Agusti Emperador* and *Josep Lluis Gelpi*

Institute for Research in Biomedicine, Barcelona

Standard molecular dynamics (MD)

Integrate the equations of motion
at each *timestep* (Δt = 2 fs)

$$F = -\frac{dV}{dr} = ma$$

$$a = \frac{dv}{dt}$$

$$v = \frac{dr}{dt}$$

500.000.000 integrations needed
to generate 1 μs of trajectory!

$$V = E_{bonded} + E_{non-bonded}$$

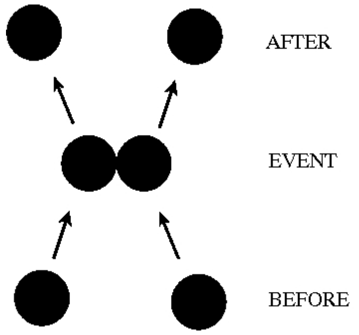
Δt such that forces over particles change
smoothly between timesteps.
Lightest particles (H) move faster,
Limiting the value of Δt

Discrete molecular dynamics (DMD)

Particles move with constant velocity…

$$r_i(t+t_c) = r_i(t) + v_i(t)t_c$$

…until a collision occurs



Transfer of linear momentum
upon a collision

$$\begin{aligned} m_i v_i &= m_i v_i' + \Delta p \\ m_j v_j + \Delta p &= m_j v_j' \end{aligned}$$

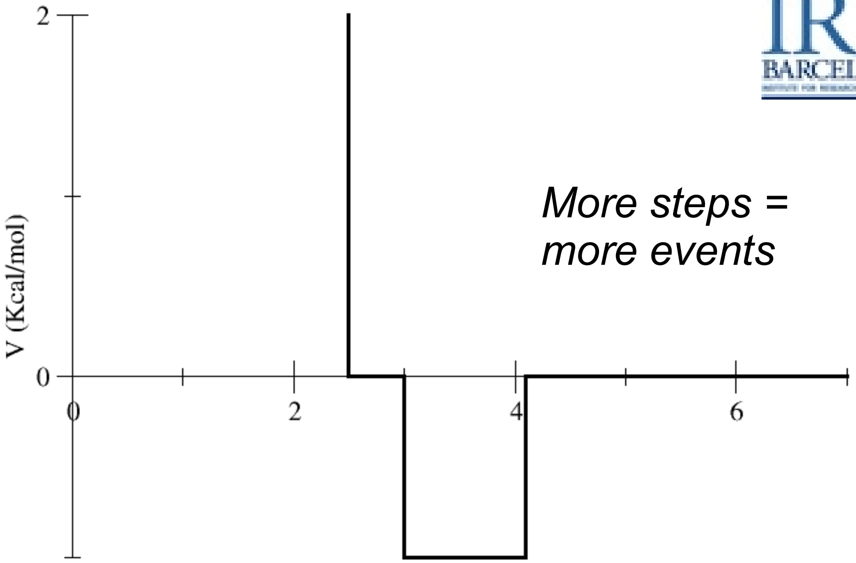
The velocities of particles i, j change
No need to integrate equations of motion.
Frequency of collisions increases with N:
Δt = constant/N

Discrete molecular dynamics

Collision (event):
Conservation of linear momentum
Conservation of energy when entering a region with different potential energy
(Emperador et al, Proteins **78** 83 (2010))

$$m_i v_i + m_j v_j = m_i v_i' + m_j v_j'$$

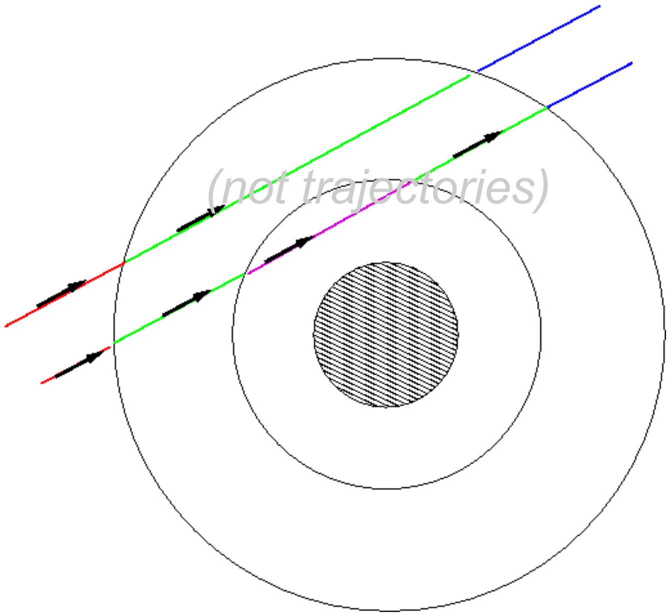
$$\frac{1}{2} m_i v_{i^2} + \frac{1}{2} m_j v_{j^2} = \frac{1}{2} m_i v_i'^2 + \frac{1}{2} m_j v_j'^2 + \Delta V$$



More steps = more events

Transferred linear momentum

$$\Delta p = \frac{m_i m_j}{m_i + m_j} \left\{ \sqrt{(v_j - v_i)^2 - 2\frac{m_i + m_j}{m_i m_j} \Delta V} - (v_j - v_i) \right\}$$



(not trajectories)

If ΔV>0, the particles can overcome the potential step as long as

$$\Delta V < \frac{m_1 m_2}{2(m_1 + m_2)} (v_j - v_i)^2$$

Otherwise, the particles keep inside the well and

$$\Delta p = \frac{2 m_i m_j}{m_i + m_j} (v_i - v_j)$$

The program, instructions for installation and manual at

http://www.scalalife.eu/content/discrete-1

Inputs for DISCRETE:

1- Topology file: chemical structure of the molecule (bonds, angles, dihedrals)

2- Coordinate file

These are generated by the setup program, whose inputs are:

1- PDB file of the protein structure

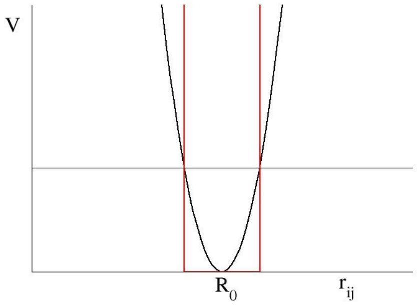
2- Library of aminoacids

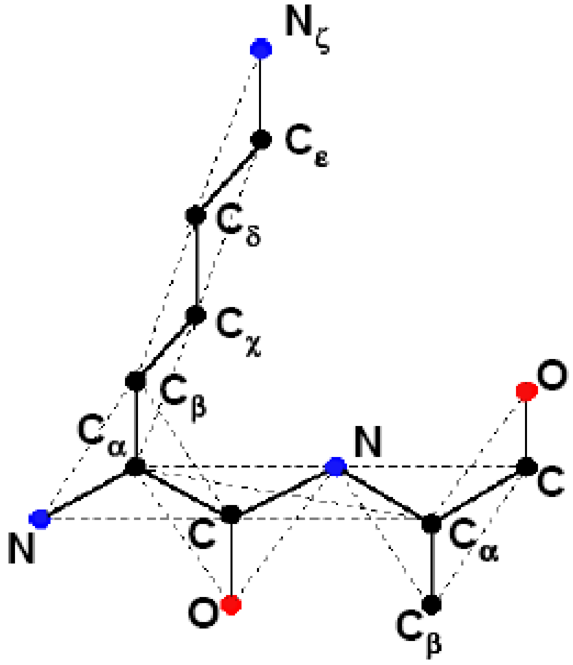
3- Interaction potential parameters

MOLECULAR STRUCTURE AND TOPOLOGY IN DMD

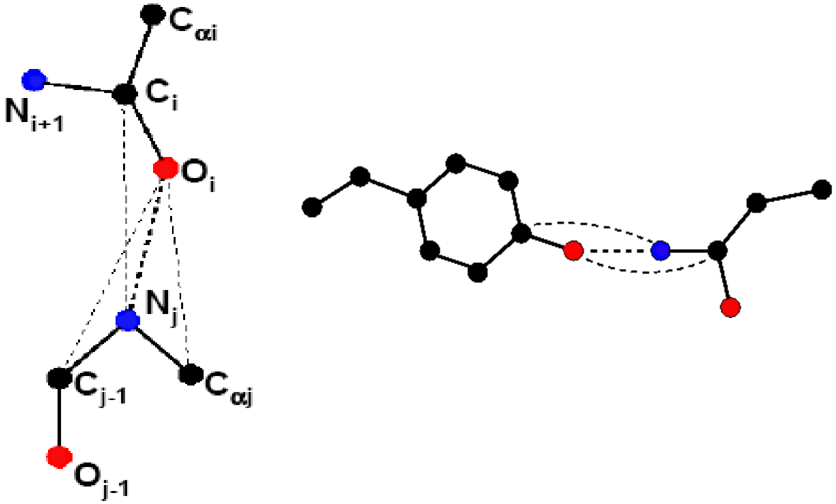
In DMD all the potentials depend on particle-particle distances (two-body potentials) It is not possible to define forces depending on angles of dihedrals. Pseudobonds have to be used to fix bond angles and dihedrals

Covalent bond,pseudobond





LYS-ALA



Hydrogen bonds (HB) are dipole-dipole interactions, therefore dependent on the angle. Pseudobonds are used to keep the correct angle.

INTERACTION POTENTIALS

Solvent considered implicitly via a solvation term

Interactions considered (nonbonded potential energy terms)

1- Van der Waals

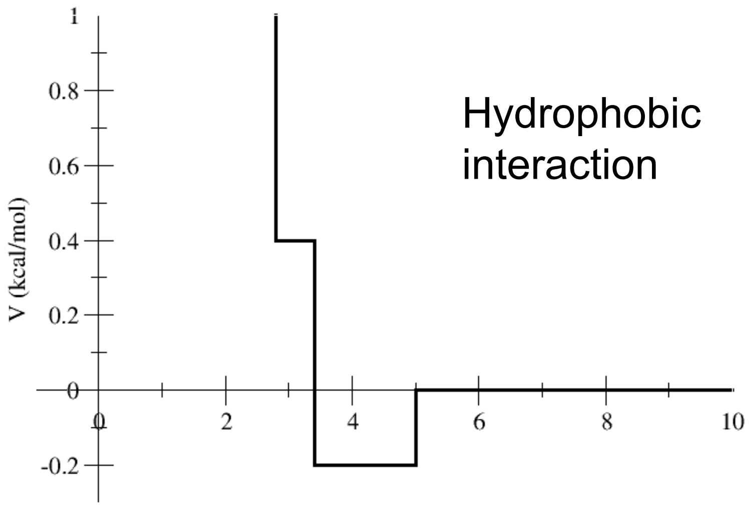
2- Electrostatic

3- Implicit solvation. EEF1 potential energy function.
Lazaridis and Karplus, Proteins **35**, 133 (1999)

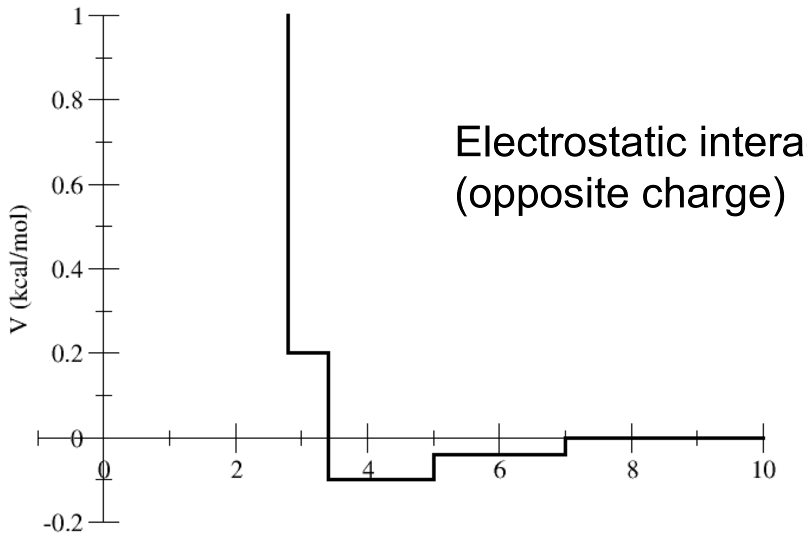
Secondary structure is restricted
(otherwise would be lost due to the simplicity of the two-step potentials;
minimum number of steps to increase speed)

DMD representation of the interaction potential terms

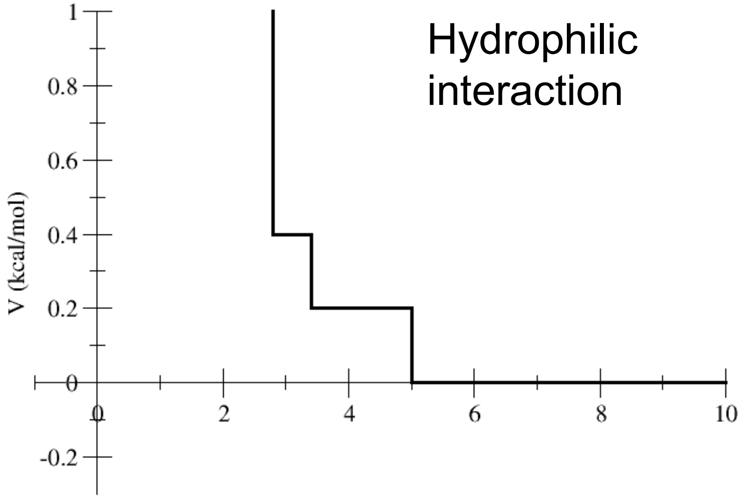
We want to use the minimum number of steps to reduce the number of events (faster simulation)



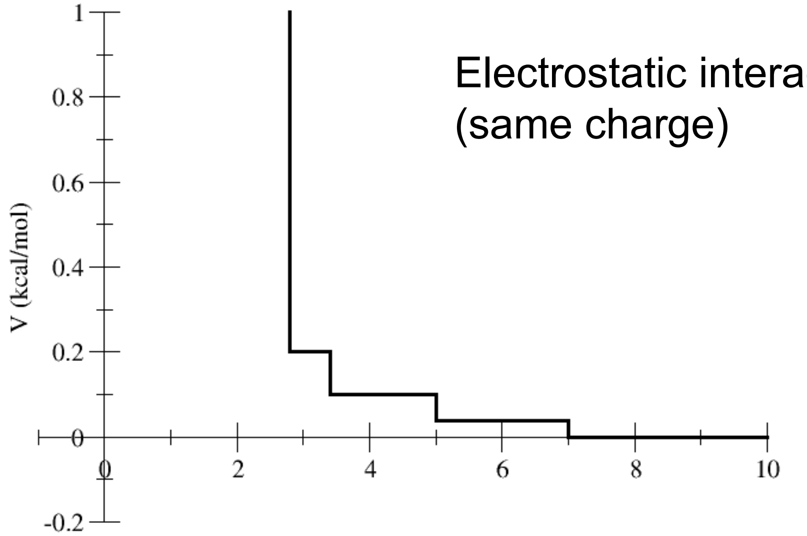
Hydrophobic interaction



Electrostatic interaction (opposite charge)



Hydrophilic interaction



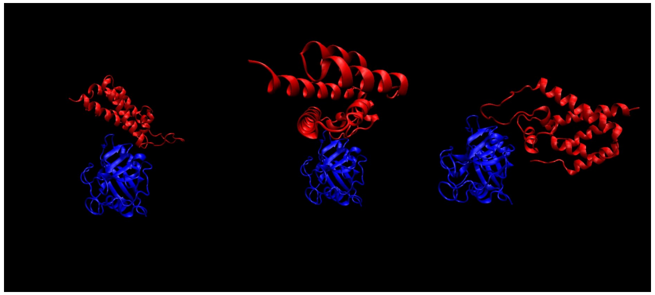
Electrostatic interaction (same charge)

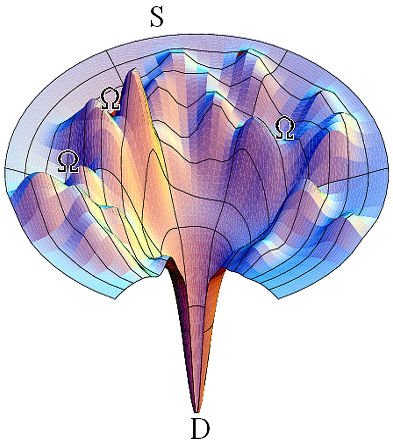
THE PROTEIN-PROTEIN DOCKING PROBLEM

Where is the interface?

To have a fast prediction method: consider proteins as rigid bodies, generate many docking conformations and score them.

Sampling in a 6D space (receptor fixed, translation (3) + rotation (3) of the ligand).
Each rigid docking pose minimized





Sources of error:

1- Insufficient sampling of rigid docking conformations (rugged free energy landscape)

2- Inaccuracies in the scoring function (that predicts the free energy, not the potential energy)

3- Ignoring protein flexibility (proteins are not rigid and can undergo conformational changes upon binding)

Why protein-protein docking?

It is an ideal case for a serial simulation program: many rigid body docking conformations have to be refined to take into account its flexibility

The refinement is made through a DMD simulation of each of the docking conformations

Only the interface between the two proteins (different interface in each configuration) is included in the simulation: the number of particles remains below 1000 (most of times)

A protein-protein docking analysis with flexibility included via DMD can be made in less that 1h in a cluster. Each node running the DMD simulation of a docking conformation

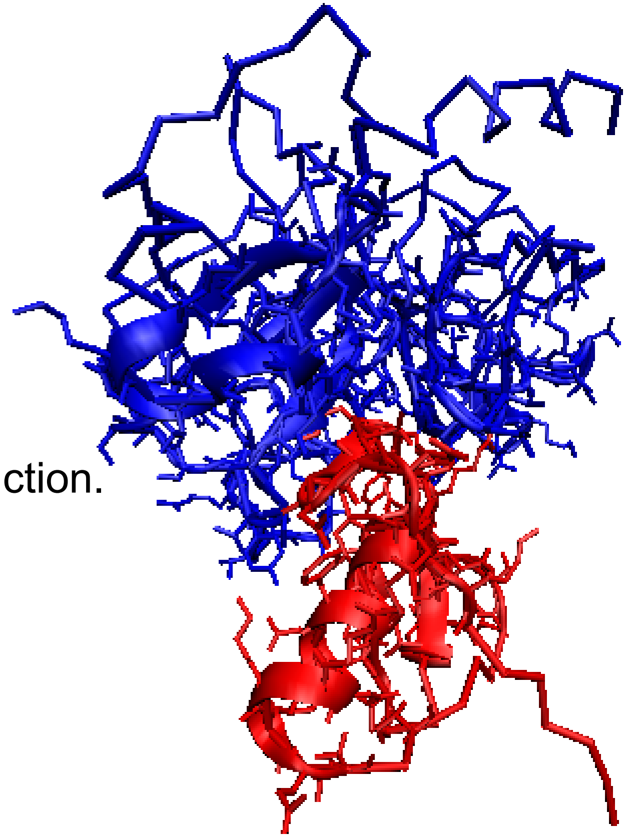
FOCUSING ON PROTEIN FLEXIBILITY: STRUCTURAL REFINEMENT

Receptor and ligand structures may change upon binding.
This should affect the **binding energy.**

Structural changes in the interface during the simulation are expected to **approach the experimental interface** and improve the binding energy of the near-native poses

We run a **discrete molecular dynamics (DMD)** simulation for each rigid docking pose.

As starting point for the DMD simulations, we choose the **100 top-ranked configurations** (out of 10000 generated with FTDOCK)
Ranking of docking poses with **pyDock**, a highly optimized scoring function.
(Cheng et al., *Proteins* **68** 503 (2007))



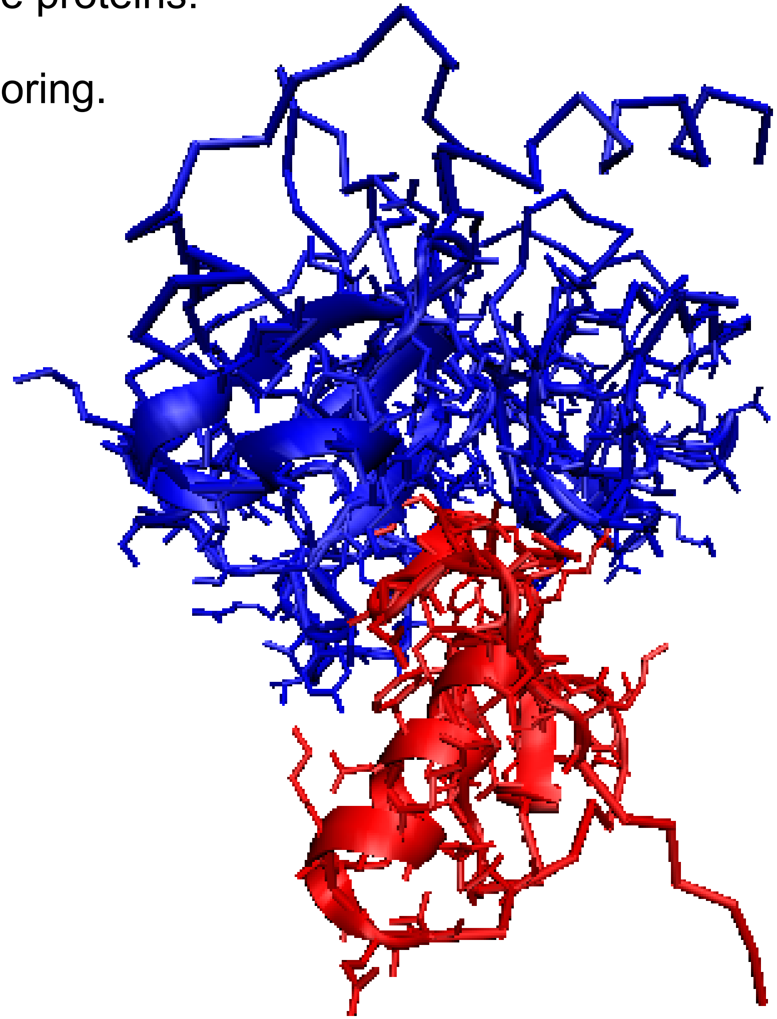
We use a **multiscale** representation of the protein:
all the atoms in the interface, only CA elsewhere.
Only the atoms at the interface contribute to the scoring of the docking conformation.

Less particles makes the simulation much faster.

MULTISCALE MODEL

DISCRETE uses a **multiscale** representation of the proteins:
all the atoms in the interface, only CA elsewhere.
Only the atoms at the interface contribute to the scoring.

Less particles makes the simulation much faster.



Position restraints between the CA out of the interface to fix the protein structure

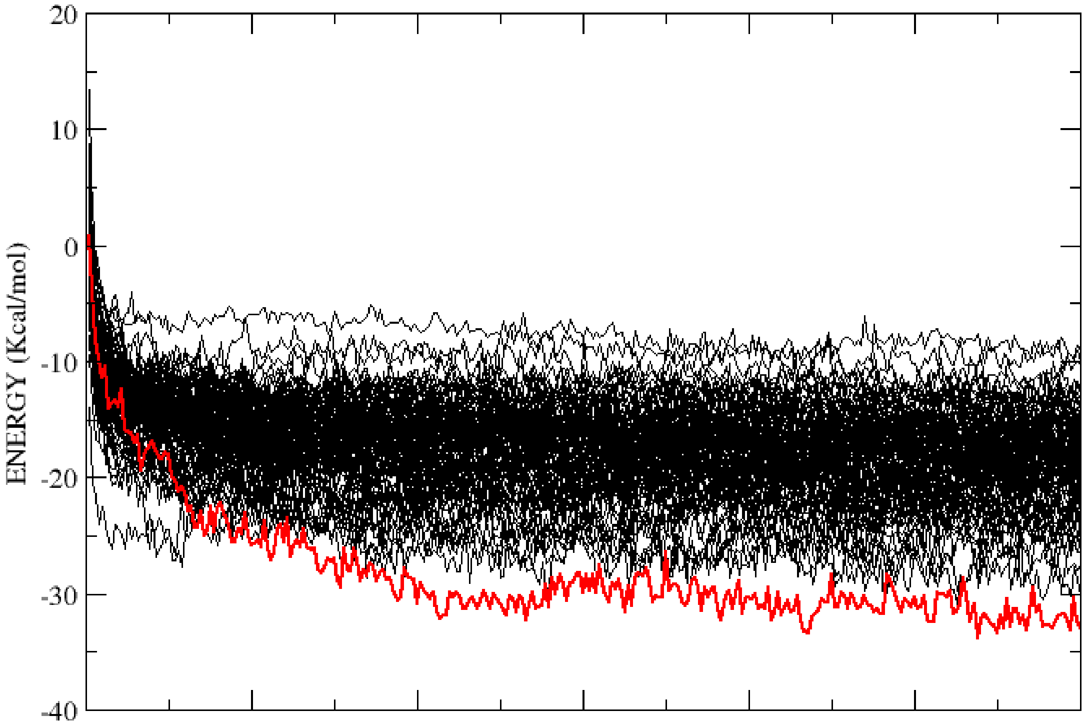
Interface: residues with some atom of the other protein within 8 A

Layer surrounding the interface: residues with some atom of the other protein within 12 A. These atoms are frozen, but interact with the atoms at the interface. They reconstruct the environment of the interface in the full atomistic model.

Parameters given to the setup program to define the width of the interface.

ENERGY OF THE DOCKING CONFIGURATIONS

We run molecular dynamics with the DMD method for each docking configuration.
The **binding energy** is obtained from the DMD interaction potential $V = V_{VdW} + V_{Solv} + V_{Coul}$



The energy varies along the simulation, and reaches a stationary average value after 2 ns of trajectory. We compute the average of each potential energy term over the last 0.5 ns of simulation, and use these averages to score each conformation.

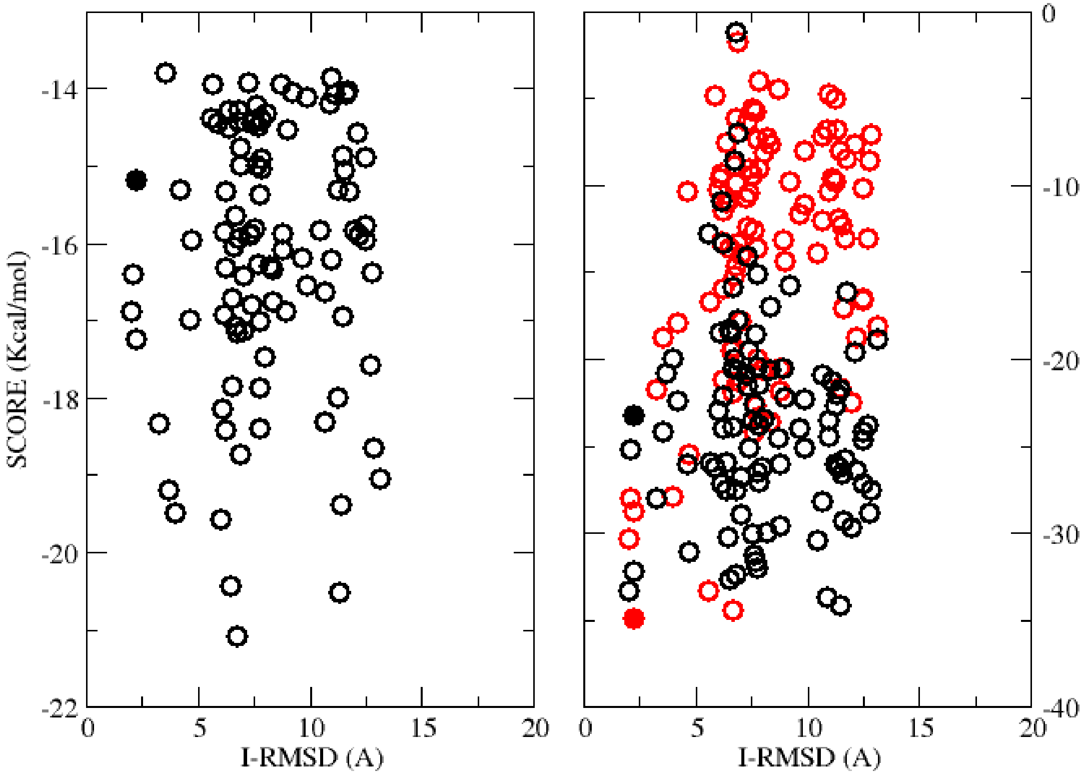
Computing time: 0.5 h for 1000 particles

The efficiency of a docking method is tested with a **benchmark** of known complexes.

We have chosen the Weng's benchark 4.0 (*Proteins* **78**, 3311 (2010))
We have studied 61 complexes of this benchmark for which pyDock gives a near native solution (interface RMSD < 4 A respect to experimental complex) between the first 100 top-ranked configurations. Ranking made with pyDock

Rigid docking

DMD refinement



Plots of score vs RMSD for each docking pose, obtained in rigid doking (left) with DMD (right)

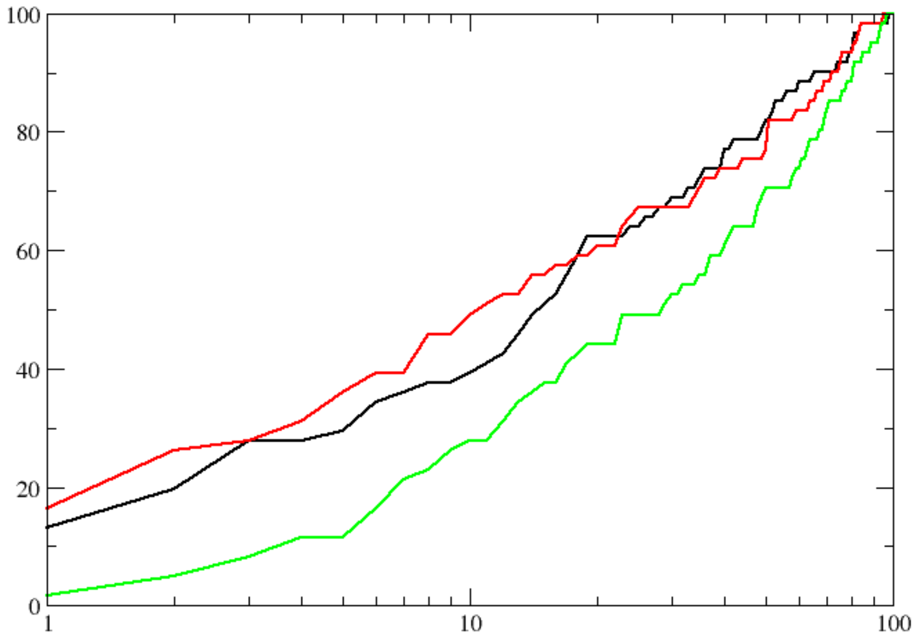
In the right panel, binding energies **before** and **after** DMD relaxation

Ideal result: **funnel-like** distribution, near native conformations with the best score (lower energy)

PERFORMANCE OF THE DMD REFINEMENT

The performance of the refinement method over the whole benchmark is evaluated with the **success rate:**

For how many complexes of the benchmark is found a near **native solution between the N top ranked** docking poses?



pyDock (rigid structures)

DMD potential (relaxed structures)

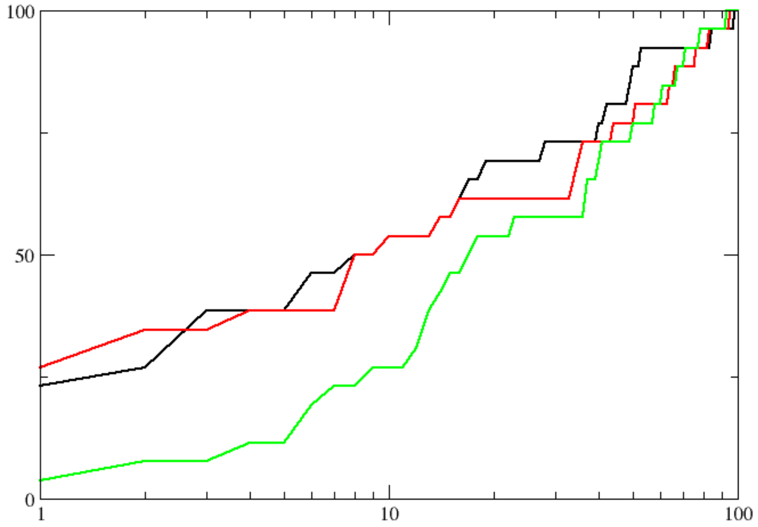
DMD potential (rigid structures)

The flexibility included via DMD has an effect equivalent to the optimization of the scoring function (improvement from green to red curve).

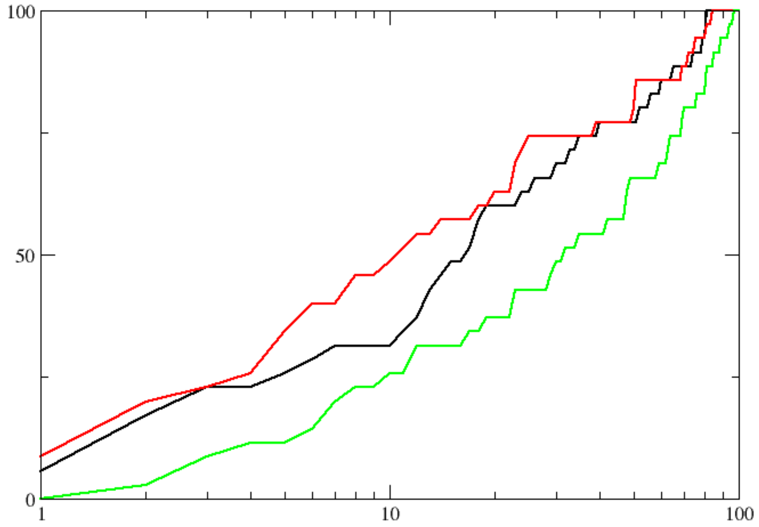
Emperador et al, *J. Chem. Theory Comput.* **9**, 1222 (2013)

DEPENDENCE WITH THE EXPERIMENTAL DEFORMATION UPON BINDING

DEFORMATION < 1 A



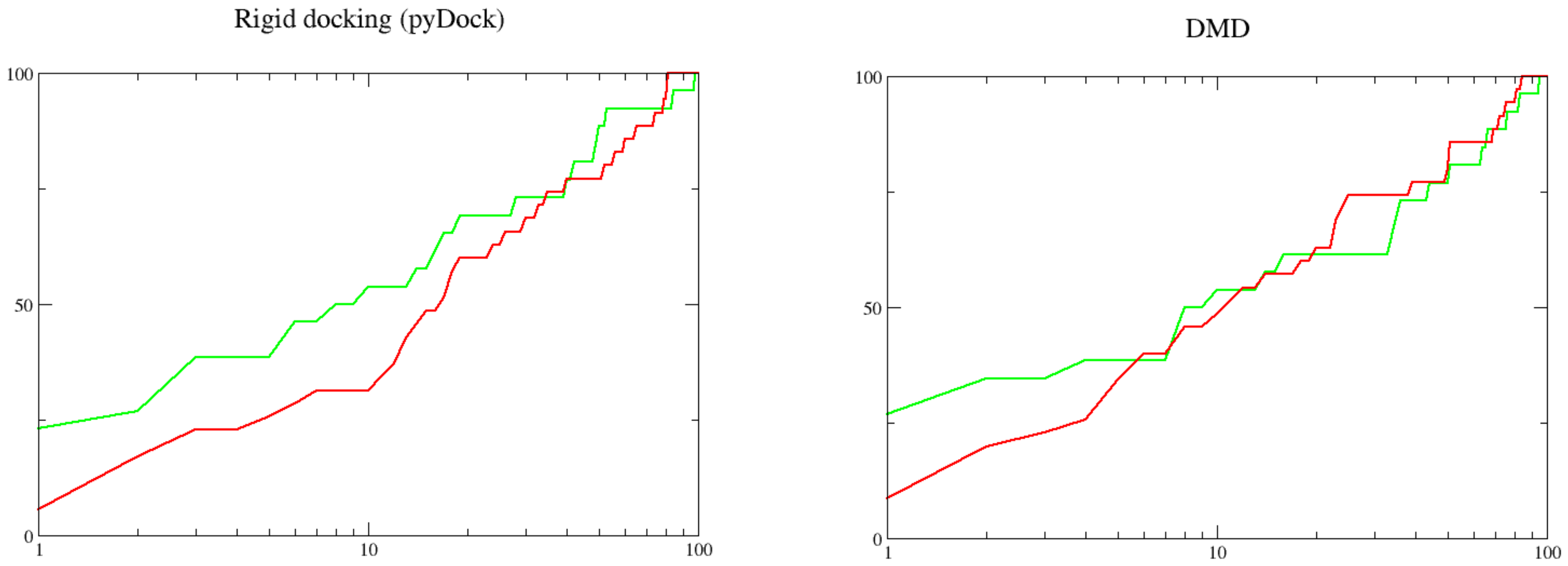
DEFORMATION > 1 A



Complexes with *low deformation upon binding*: rigid docking scored with pyDock and DMD show the same performance because deformation does not affect seriously the interaction energy

Complexes with *high deformation upon binding:* backbone movements are relevant and modify the binding energy between receptor and ligand, improving the success rates

RIGID DOCKING vs FLEXIBLE DOCKING



At TOP10, the performance of the DMD method (right panel) is virtually the same for 'rigid' and 'flexible' complexes

Unlike rigid docking based methods, the performance of flexible docking is not seriously affected by conformational changes upon binding. This is due to the **amplitude of the structural changes** during the DMD trajectory.