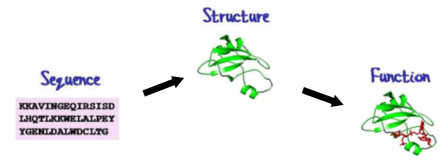
Algorithmic robotics and Motion Planing

Spring 2011

Dynamic Maintenance of Kinematic Chains

Itay Lotan

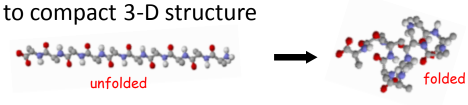
Central Dogma of Molecular Biology



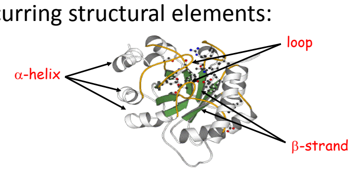
Understanding the 3-D structure of proteins is essential to understanding life

Protein Folding

- Protein chain spontaneously collapses (folds) to compact 3-D structure

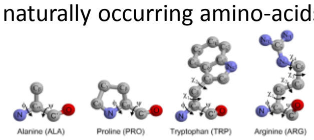


- Recurring structural elements:

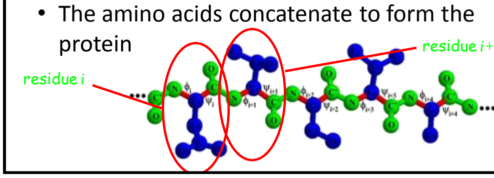


Chain of Amino-Acids

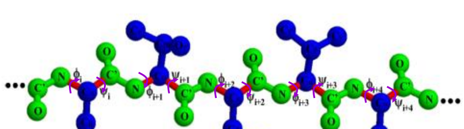
- 20 naturally occurring amino-acids



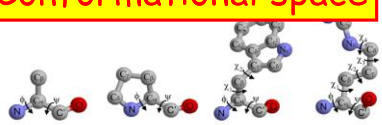
- The amino acids concatenate to form the protein



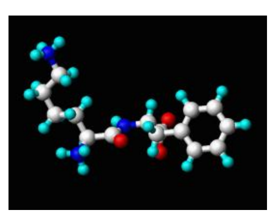
ϕ-ψ Kinematic Linkage Model

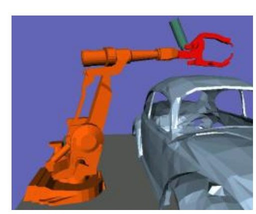


Conformational space



Molecule ≈ Robot



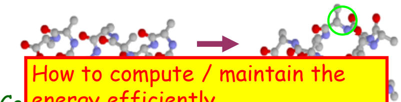


Monte Carlo Simulation (MCS)

- Popular method for sampling the conformation space of proteins
- Used for
  - estimating thermodynamic quantities
  - searching for low-energy conformations and the folded structure

MCS: How It Works

1. Perturb current conformation at random



How to compute / maintain the energy efficiently

2. Co
3. Accept with probability:

$$P(accept) = \min\left\{1, e^{-\Delta E / k_B T}\right\}$$

[E - energy, k_B - Boltzmann's constant, T - temperature]

Requires >>10^9 steps to sample adequately

Energy Function

- E = Σ bonded terms
  + Σ non-bonded terms
  + Σ solvation terms
- Bonded terms
  - *Bond-stretching, bending, rotating*
  - *O(n)*
- Non-bonded terms
  - E.g., Van der Waals and electrostatic
  - Depend on distances between pairs of atoms
  - $O(n^2)$ → Expensive to compute
- Solvation terms
  - May require computing molecular surface

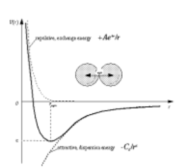
Energy Function

- E = Σ bonded terms
  + Σ non-bonded terms
  + Σ solvation terms
- Bonded terms
  - *Bond-stretching, bending, rotating*
  - *O(n)*
- Non-bonded terms
  - E.g., Van der Waals and electrostatic
  - Depend on distances between pairs of atoms
  - $O(n^2)$ → Expensive to compute
- Solvation terms
  - May require computing molecular surface

Non-Bonded Terms

- Energy terms go to 0 when distance increases
  - Cutoff distance (6 - 12Å)
- vdW forces prevent atoms from bunching up
  - Only *O(n)* interacting pairs

    [Halperin & Overmars, '98]



Challenge: find interacting pairs without enumerating all atom pairs?

Finding Pairs

	Update	Detection
Brute Force	$\Theta(n)$	$\Theta(n^2)$
Grid	$\Theta(n)$	$\Theta(n)$
Spatially-adapted hierarchy	$O(n \log n)$	$\Theta(n)$
Chain-aligned hierarchy	$O(n)$	$\Theta(n^{4/3})$
ChainTree	$O(k \log n/k)$	$\Theta(n^{4/3})$

Finding Pairs

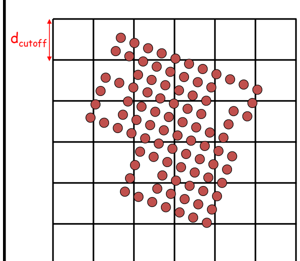
	Update	Detection
Brute Force	$\Theta(n)$	$\Theta(n^2)$
Grid	$\Theta(n)$	$\Theta(n)$
Spatially-adapted hierarchy	$O(n \log n)$	$\Theta(n)$
Chain-aligned hierarchy	$O(n)$	$\Theta(n^{4/3})$
ChainTree	$O(k \log n/k)$	$\Theta(n^{4/3})$

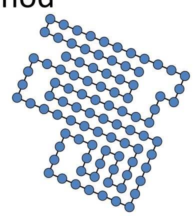
Finding Pairs

	Update	Detection
Brute Force	$\Theta(n)$	$\Theta(n^2)$
Grid	$\Theta(n)$	$\Theta(n)$
Spatially-adapted hierarchy	$O(n \log n)$	$\Theta(n)$
Chain-aligned hierarchy	$O(n)$	$\Theta(n^{4/3})$
ChainTree	$O(k \log n/k)$	$\Theta(n^{4/3})$

PARTITION THE SPACE

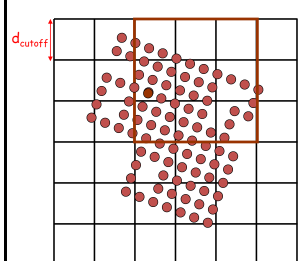
Grid Method





- Subdivide 3-space into cubic cells
- Compute cell that contains each atom center
- Represent grid as hash table

Grid Method



- $\Theta(n)$ time to build grid
- O(1) time to find interacting pairs for each atom
- $\Theta(n)$ to find all interacting pairs

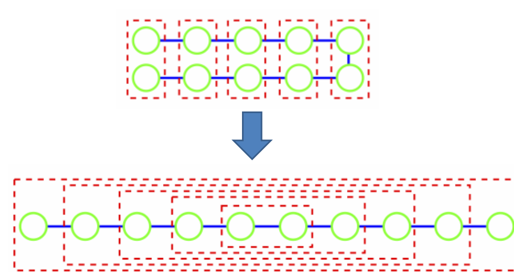
Asymptotically optimal in worst-case

Finding Pairs

	Update	Detection
Brute Force	$\Theta(n)$	$\Theta(n^2)$
Grid	$\Theta(n)$	$\Theta(n)$
Spatially-adapted hierarchy	$O(n \log n)$	$\Theta(n)$
Chain-aligned hierarchy	$O(n)$	$\Theta(n^{4/3})$
ChainTree	$O(k \log n/k)$	$\Theta(n^{4/3})$

PARTITION THE OBJECT

Finding Pairs



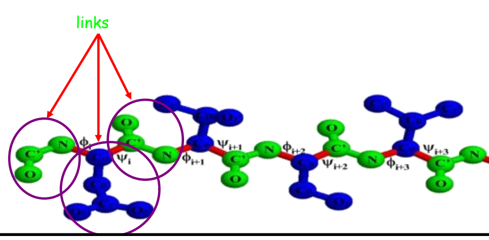
Finding Pairs

	Update	Detection
Brute Force	$\Theta(n)$	$\Theta(n^2)$
Grid	$\Theta(n)$	$\Theta(n)$
Spatially-adapted hierarchy	$O(n \log n)$	$\Theta(n)$
Chain-aligned hierarchy	$O(n)$	$\Theta(n^{4/3})$
ChainTree	$O(k \log n/k)$	$\Theta(n^{4/3})$

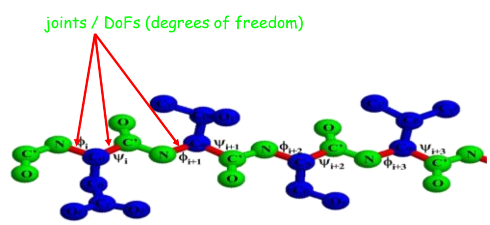
Finding Pairs

	Update	Detection
Brute Force	$\Theta(n)$	$\Theta(n^2)$
Grid	$\Theta(n)$	$\Theta(n)$
Spatially-adapted hierarchy	$O(n \log n)$	$\Theta(n)$
Chain-aligned hierarchy	$O(n)$	$\Theta(n^{4/3})$
ChainTree	$O(k \log n/k)$	$\Theta(n^{4/3})$

Protein as Kinematic Chain

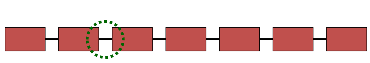


Protein as Kinematic Chain



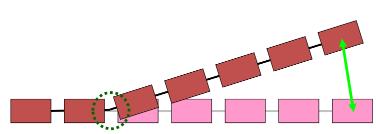
Properties of kinematic chains

- Small changes ⇒ large effects



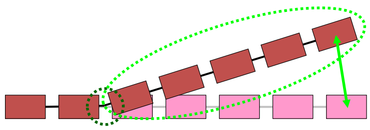
Properties of kinematic chains

- Small changes ⇒ large effects



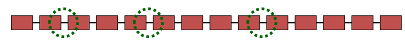
Properties of kinematic chains

- Small changes ⇒ large effects
- Local changes ⇒ global effects



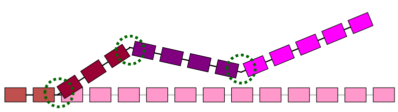
Properties of kinematic chains

- Small changes ⇒ large effects
- Local changes ⇒ global effects
- Few DoF changes ⇒ long rigid sub-chains



Properties of kinematic chains

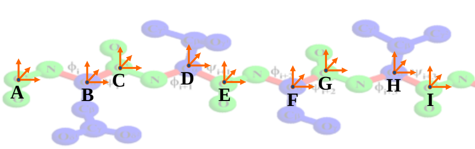
- Small changes ⇒ large effects
- Local changes ⇒ global effects
- Few DoF changes ⇒ long rigid sub-chains



ChainTree: A tale of two hierarchies

- **Transform hierarchy:** approximates kinematics of protein backbone at successive resolutions
- **Bounding volume hierarchy:** approximates geometry of protein at successive resolutions

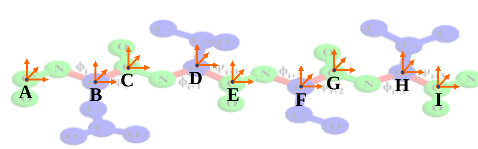
Hierarchy of Transforms



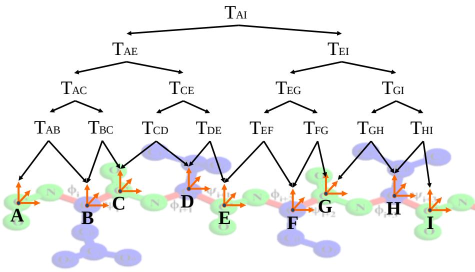
Hierarchy of Transforms

Rigid Body Transform = **T**ranslation + **R**otation

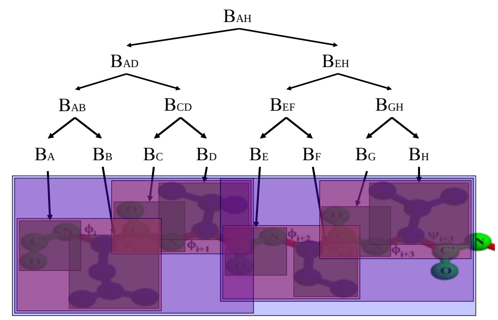
$$T = \begin{bmatrix} t_0 \\ t_1 \\ t_2 \end{bmatrix} \qquad R = \begin{bmatrix} r_{0,0} & r_{0,1} & r_{0,2} \\ r_{1,0} & r_{1,1} & r_{1,2} \\ r_{2,0} & r_{2,1} & r_{2,2} \end{bmatrix}$$



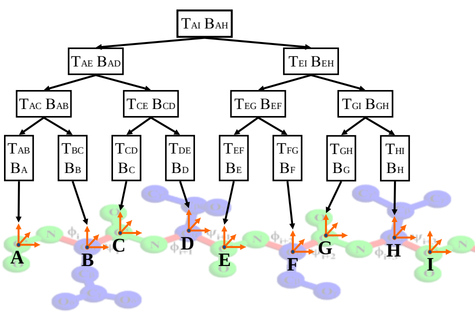
Hierarchy of Transforms



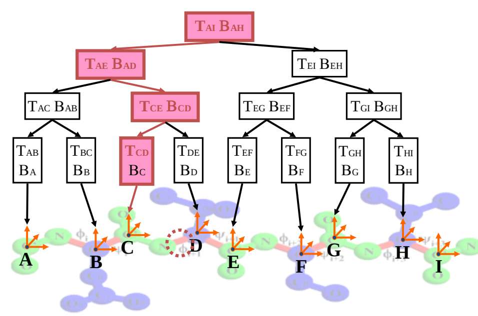
Hierarchy of Bounding Volumes (BVs)



The ChainTree



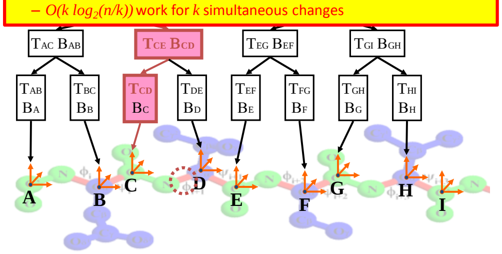
Updating the ChainTree



Updating the ChainTree

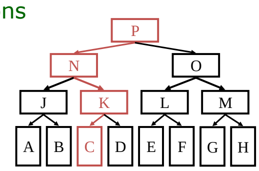
Update path to root:

- Recompute transforms that "shortcut" the DoF change
- Recompute BVs that contain the DoF change
- $O(k \log_2(n/k))$ work for k simultaneous changes



Detecting Interactions

Recursively search ChainTree for interactions



Pruning rules:

1. Prune search when distance between bounding volumes is more than cutoff distance
2. Do not search inside rigid sub-chains (no marked node between the tested nodes)

Computational complexity

- Updating:

$$O\left(k \log n/k\right)$$

- Searching:

$$\Theta\left(n^{4/3}\right)$$ worst case bound

Much faster in practice

Proof

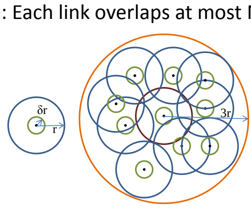
- Regularize the chain:
  - Replace all links by their bounding spheres
  - Make all sphere equal in size by growing smaller spheres
  - Further grow all links equally until each pair of consecutive links is in contact

No effect on complexity of the problem

Proof

Assumption: Distance between link centers > $\delta \cdot r$

Proposition: Each link overlaps at most M other links



At level i, replace r with $2^i \cdot r$

$M_i = 27 \cdot (2^i)^2 / q \quad 0 < q < 1$

Proof

M_i is bounded by number of BVs at level i

$$M_i \leq n/2^i$$

This bound is reached when $i > 1/3 \log n$

$$T = \sum_{i=0}^{\frac{1}{3}\log n} \left(\frac{27 \cdot 2^{2i}}{q}\right)\left(\frac{n}{2^i}\right) + \sum_{\frac{1}{3}\log n}^{\log n} \left(\frac{n}{2^i}\right)^2$$
$$= \frac{27n}{q}\left(2n^{1/3} - 1\right) + \frac{4}{3}\left(n^{4/3} - 1\right)$$
$$= O\left(n^{4/3}\right)$$

Proof

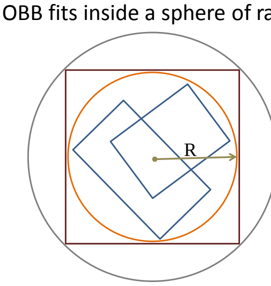
This bound holds as long as each BV at level i can be enclosed by sphere of radius $c \cdot 2^i r$

We still need to show this is true for the ChainTree BVs

We will prove the existence of c by induction

Proof

Lemma: given two OBBs inside sphere of radius R, their OBB fits inside a sphere of radius $\sqrt{3} \cdot R$



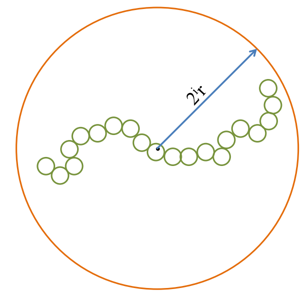
Proof

Base: for levels 0 through 4 choose an appropriate $c = c_1$

Induction step:

- Each BV at level i-5 fits inside a sphere of radius $c \cdot 2^{i-5} r$
- A sphere of radius $2^i r$ is enough to bound all links inside a BV at level i

Proof



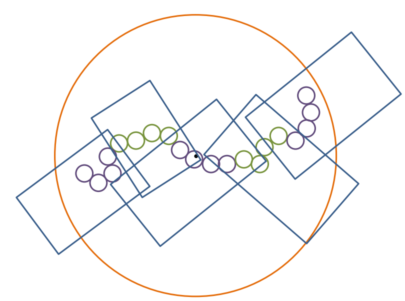
Proof

Base: for levels 0 through 4 choose an appropriate $c=c_1$

Induction step:

- Each BV at level i-5 fits inside sphere of radius $c \cdot 2^{i-5} r$
- A sphere of radius $2^i r$ is enough to bound all links inside a BV at level i
- No point in any of the 32 BVs at level i-5 is further than $2^i r + 2 \cdot c \cdot 2^{i-5} r$ from the center of this sphere

Proof



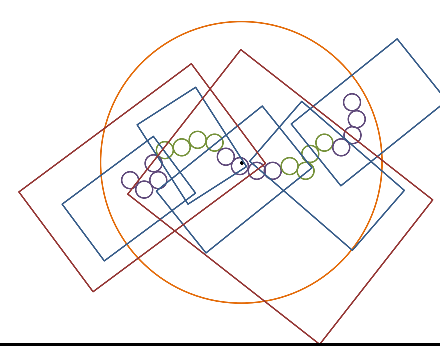
Proof

Base: for levels 0 through 4 choose an appropriate $c=c_1$

Induction step:

- Each BV at level i-5 fits inside sphere of radius $c \cdot 2^{i-5} r$
- A sphere of radius $2^i r$ is enough to bound all links inside a BV at level i
- No point in any of the 32 BVs at level i-5 is further than $2^i r + 2 \cdot c \cdot 2^{i-5} r$ from the center of this sphere
- No point in any of the 16 BVs at level i-4 is further than $\sqrt{3}(2^i r + 2 \cdot c \cdot 2^{i-5} r)$ from the center of this sphere

Proof



Proof

$$\left(2^i r + 2c \cdot 2^{i-5} r\right) \cdot \sqrt{3}^5 \le c \cdot 2^i r$$

$$\left(\frac{1}{\sqrt{3}^5} - \frac{1}{16}\right)^{-1} \le c$$

$$c = \max\left\{\left(\frac{1}{\sqrt{3}^5} - \frac{1}{16}\right)^{-1}, c_1\right\}$$

Proof

- We have shown $O(n^{4/3})$
- Still need to prove $\Omega(n^{4/3})$
- Construct a chain configuration where bound is achieved

Proof

Each ***link*** is a sphere of radius r

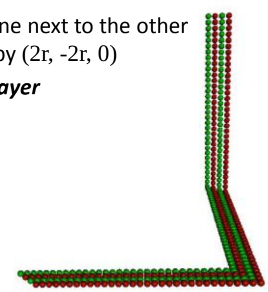
- Place d ***links*** along x-axis (starting at origin)
- Place d ***links*** parallel to y-axis
- Place d ***links*** parallel to z-axis

This constitutes a ***unit***

Proof

- Place d/8 ***units*** one next to the other each translated by (2r, -2r, 0)

This constitutes a ***layer***

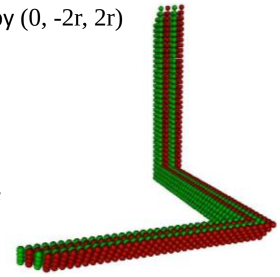


Proof

- Place d/8 ***layers*** one on top of the other translated by (0, -2r, 2r)

This is the ***chain***

$$n = d/8 \ \textit{layers} = d^2/64 \ \textit{units} = 3d^3/64 \ \textit{links} = O(d^3)$$



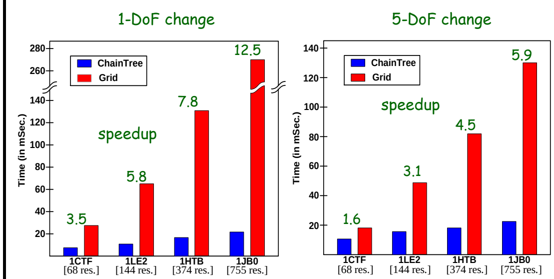
Proof

- The point $\left(2(d-1)r, (d-1)r, \tfrac{1}{4}(d-1)r\right)$ is contained inside convex hull of each ***unit***
- There is a level in BVH where each ***unit*** has its own BV
- At this level all BV pairs intersect
- We have $\frac{1}{2}\left(d^2/64\right)\left(d^2/64 - 1\right)$ overlaps: $\Omega(n^{4/3})$

Experimental Setup

- Energy function:
  - Van der Waals
  - Electrostatic
  - Attraction between folded-state contacts
  - Cutoff at 12Å
- 300,000 steps MCS with Grid and ChainTree
- **Steps are the same with both methods**
- Early rejection for large vdW terms

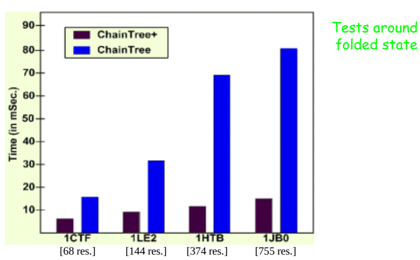
Test



Two-Pass ChainTree

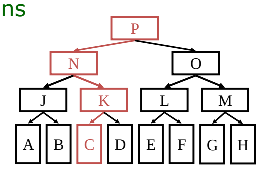
(ChainTree+)

1st pass: small cutoff distance to detect steric clashes
2nd pass: normal cutoff distance



Computing the Energy

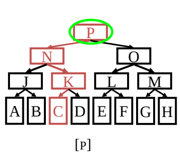
Recursively search ChainTree for interactions



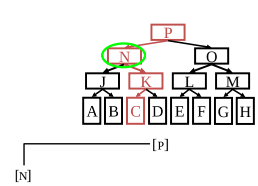
Pruning rules:

1. Prune search when distance between bounding volumes is more than cutoff distance
2. Do not search inside rigid sub-chains (no marked node between the tested nodes)

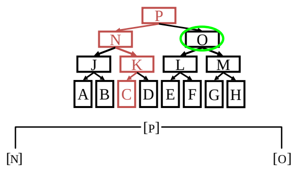
Computing the energy



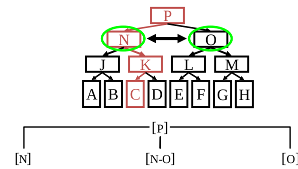
Computing the energy



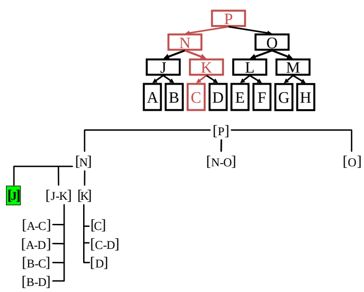
Computing the energy



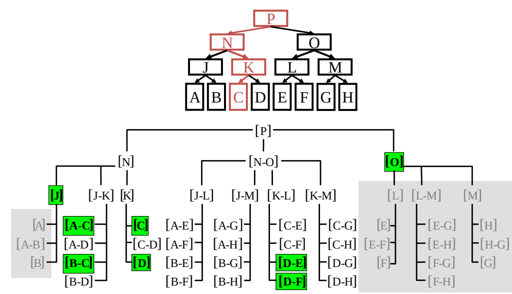
Computing the energy



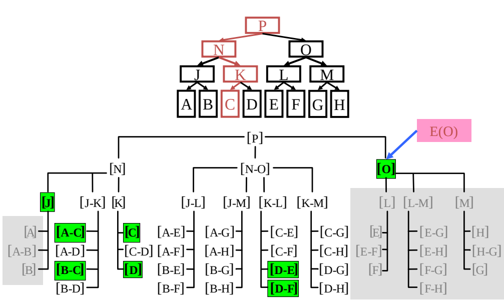
Computing the energy



Computing the Energy

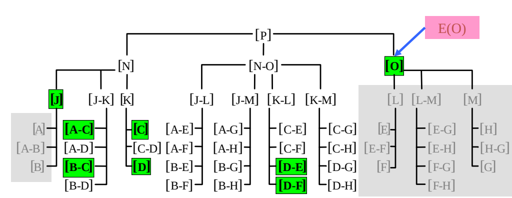


Computing the Energy

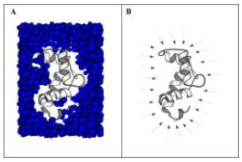


Computing the Energy

The Energy Tree



Extension: Interaction with Solvent



- Implicit solvent model: solvent as continuous medium, interface is solvent-accessible surface

E. Eyal, D. Halperin. Dynamic Maintenance of Molecular Surfaces under Conformational Changes. Proc. 21st ACM Symposium on Computational Geometry (SoCG05) 2005