Interactive Protein Manipulation

Oliver Kreylos[a], Bernd Hamann[a], Nelson L. Max[a],

Silvia N. Crivelli[b], and E. Wes Bethel[b]

{kreylos,hamann,max}@cs.ucdavis.edu; {sncrivelli,ewbethel}@lbl.gov.

[a] Center for Image Processing and Integrated Computing,

Department of Computer Science,

University of California, Davis;

[b] Lawrence Berkeley National Laboratory,
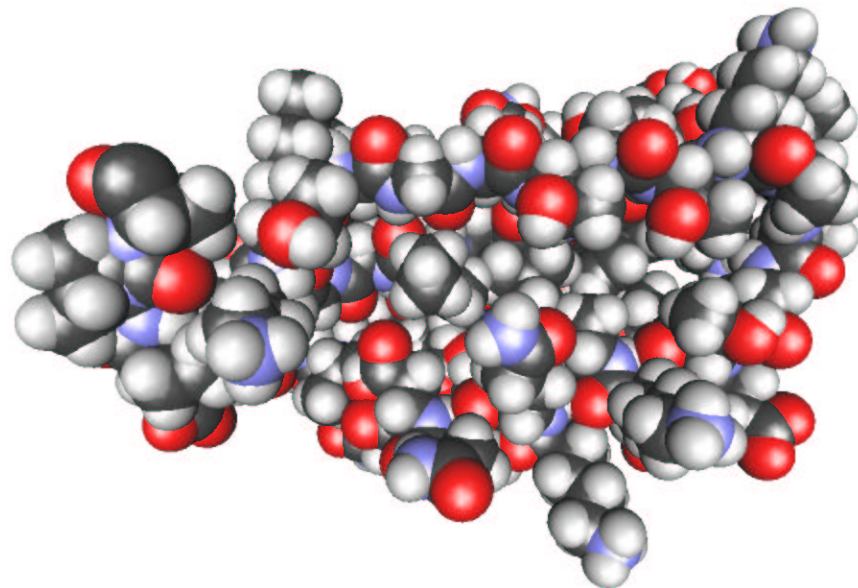
Berkeley, California

Outline

- The problem
- Our approach
  - Creating structures "from scratch"
  - Manipulating structures
- Live demo

The Problem

Protein folding: The problem of translating

$\cdots$ GCCAATGATAACGGAGTGGACGGGGTA $\cdots$

into

Why We Want To Solve It

- The genotype–phenotype knowledge gap:
  - Gene defines chemical structure
  - Chemical structure defines 3D structure
  - 3D structure defines protein function
- Traditional techniques too slow and tedious
  - X-ray crystallography
  - NMR
- Quick and reliable structure prediction opens many research and application areas
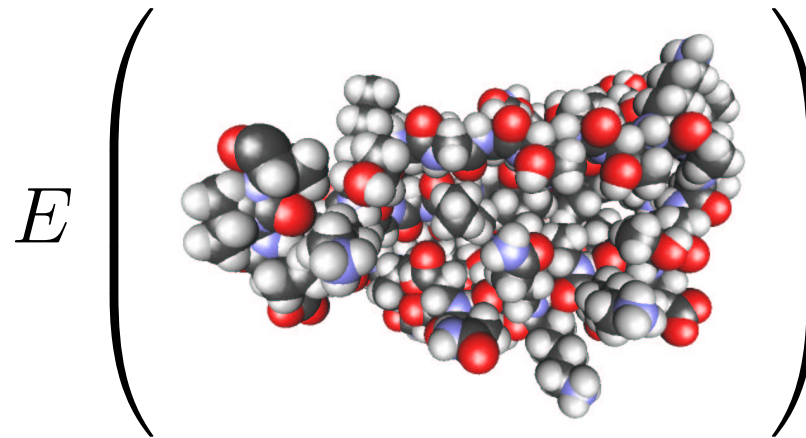
Protein Folding by Optimization

Approach: Find global minimum of

$$E \left(\quad \right)$$

Protein Folding by Optimization

Approach: Find global minimum of

$$E \left(\right)$$
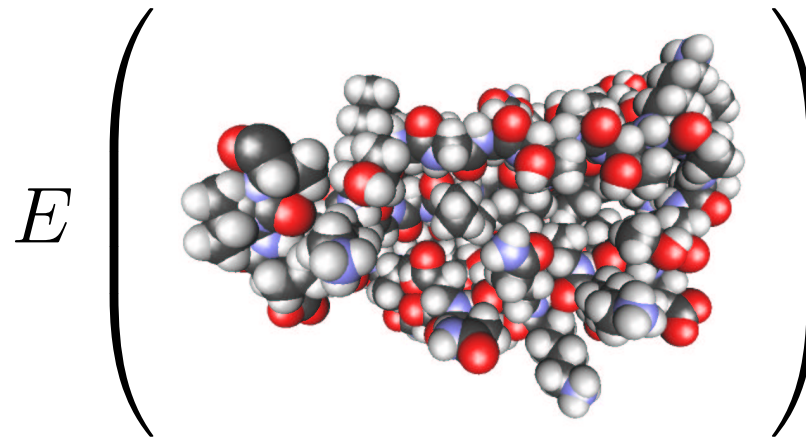
$\Rightarrow$ Difficult global optimization problem

- Many degrees of freedom
- Many local minima

Global Optimization Algorithm

- Create set S of initial configurations

- Repeat until minimum found:
  - Pick configuration C from S
  - Modify C into C'
  - If $E(C') < E(C)$, put C' into S

- Report best configuration(s) from S

Global Optimization Algorithm

- Create set S of initial configurations
- Repeat until minimum found:
  - Pick configuration C from S
  - Modify C into C'
  - If $E(C') < E(C)$, put C' into S
- Report best configuration(s) from S

Problem: Creating good initial configurations

Creating Initial Configurations

Creating a 3D protein configuration from a gene takes three steps:

1. Decode gene

2. Predict secondary structures

3. Concatenate amino acid residues

Decode Gene

- Each triplet of base pairs (codon) in a gene encodes for one amino acid in a protein

- 64 possible codons encode for 20 amino acid types (and end-of-gene marker)

- Encoding is known

$\cdots$ GCC AAT GAT AAC GGA GTG GAC GGG GTA $\cdots$
A N D N G V D G V

Predict Secondary Structures

- Three types of substructures:
  - α-helix
  - β-strand
  - Coil (loop, turn) region
- Methods of substructure type prediction:
  - Hidden Markov models
  - Neural networks

$$\ldots \quad \begin{array}{cccccccc} A & N & D & N & G & V & D & G & V \\ H & H & H & C & C & C & E & E & E \end{array} \quad \ldots$$

Concatenate Amino Acid Residues

- Assemble protein by concatenating amino acid residue templates

- Modify templates during assembly to create secondary structures

Purely geometric approach $\Rightarrow$ fast

Concatenate Amino Acid Residues

- Assemble protein by concatenating amino acid residue templates

- Modify templates during assembly to create secondary structures
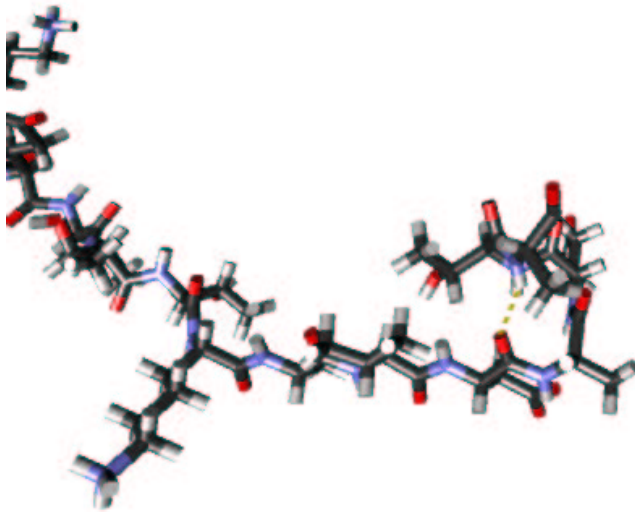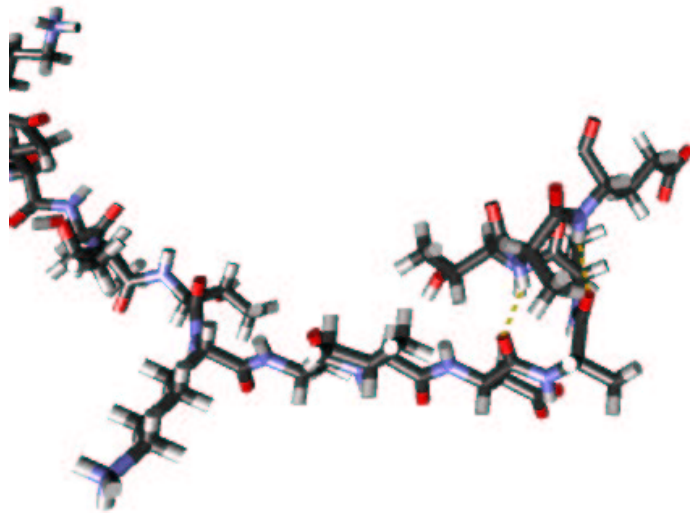
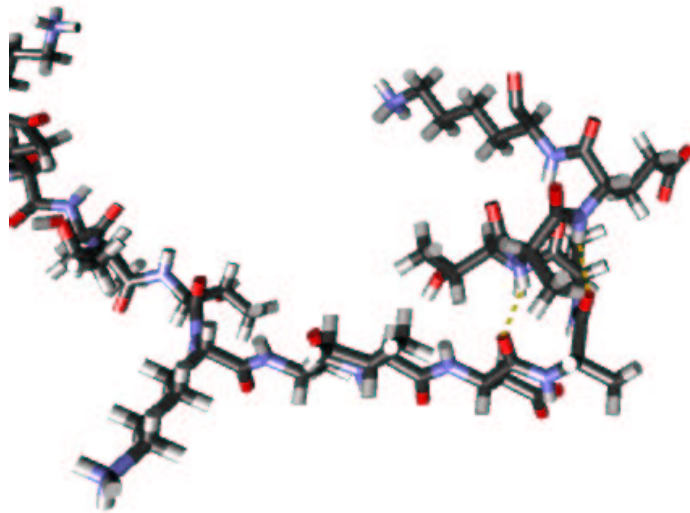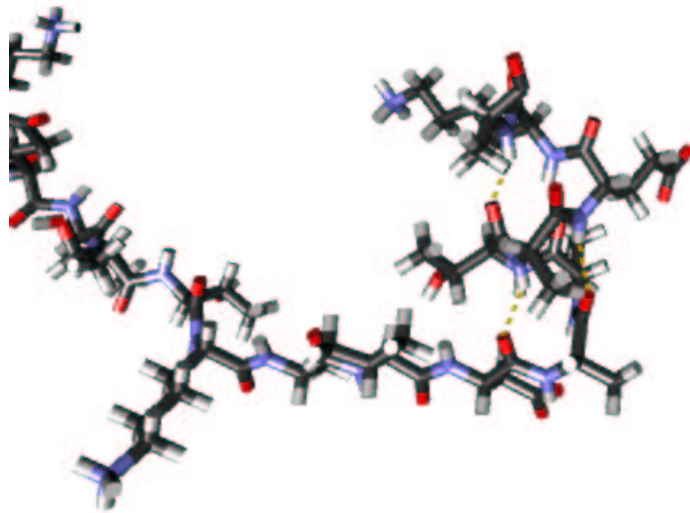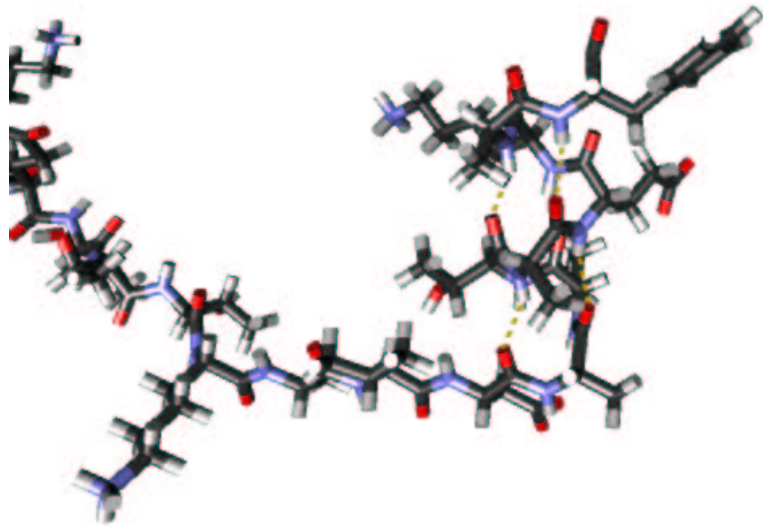Purely geometric approach $\Rightarrow$ fast

Concatenate Amino Acid Residues

Concatenate Amino Acid Residues

Concatenate Amino Acid Residues

Concatenate Amino Acid Residues

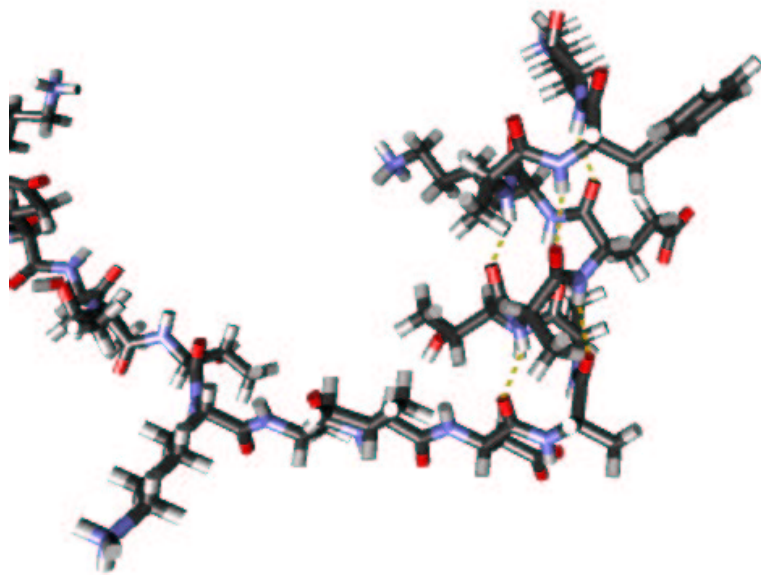# Concatenate Amino Acid Residues
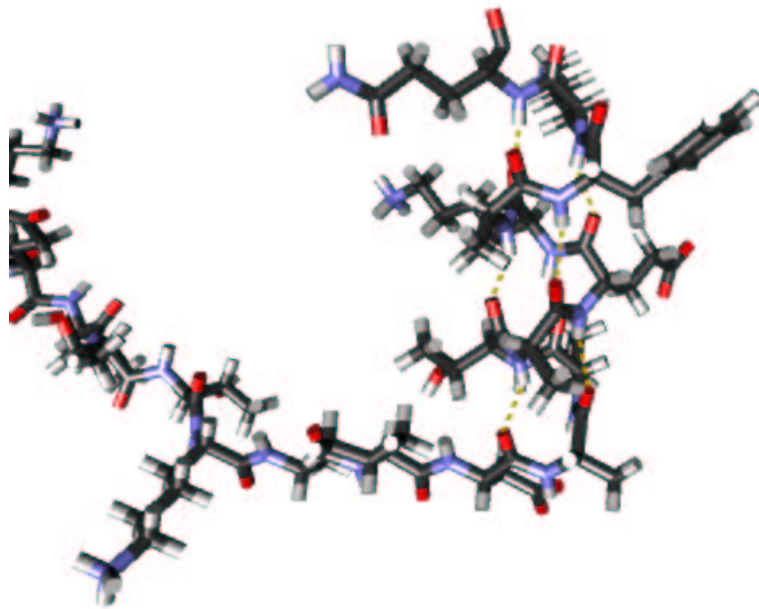
Concatenate Amino Acid Residues

Concatenate Amino Acid Residues

Concatenate Amino Acid Residues

Concatenate Amino Acid Residues

Concatenate Amino Acid Residues

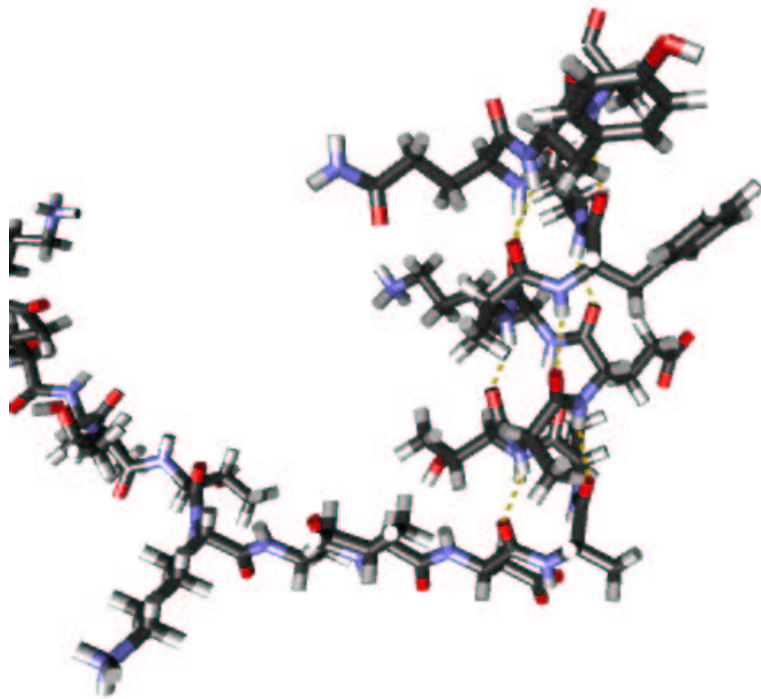# Concatenate Amino Acid Residues

Concatenate Amino Acid Residues

Concatenate Amino Acid Residues
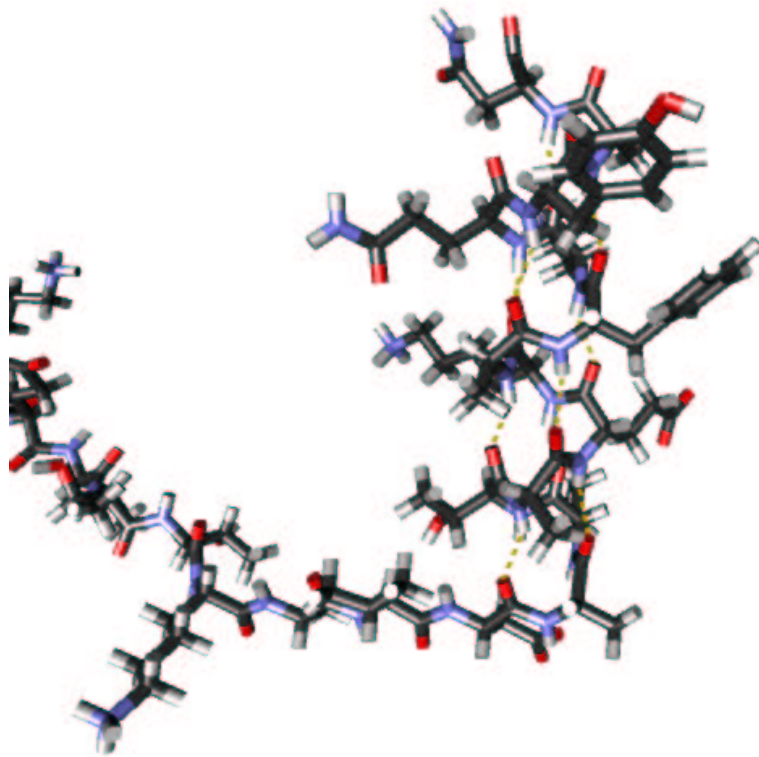
Concatenate Amino Acid Residues

Concatenate Amino Acid Residues

Concatenate Amino Acid Residues
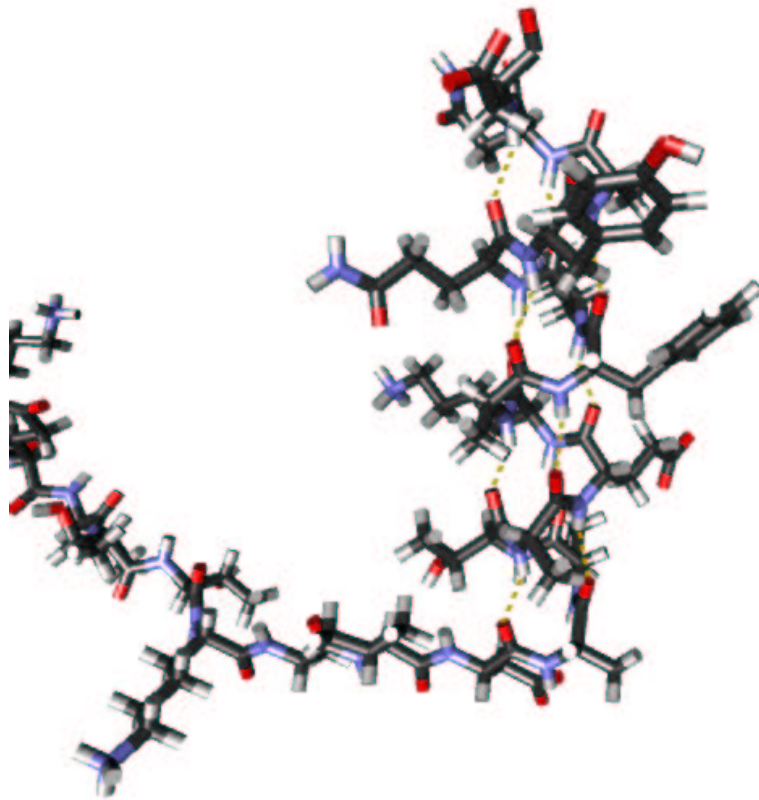
Concatenate Amino Acid Residues

Concatenate Amino Acid Residues

Concatenate Amino Acid Residues

Concatenate Amino Acid Residues
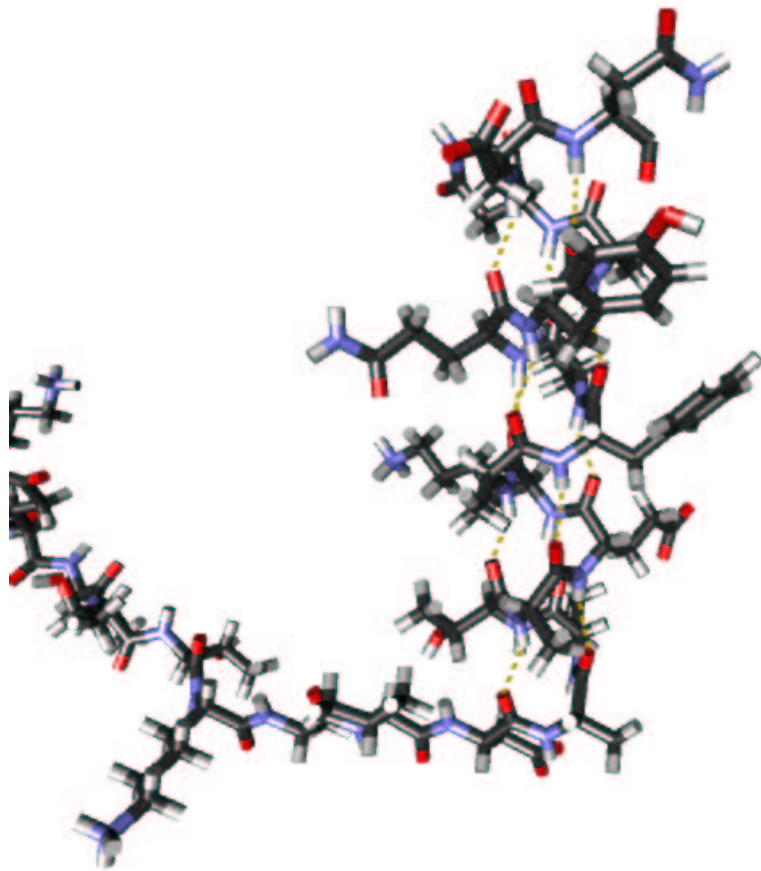
Concatenate Amino Acid Residues

Concatenate Amino Acid Residues

Concatenate Amino Acid Residues
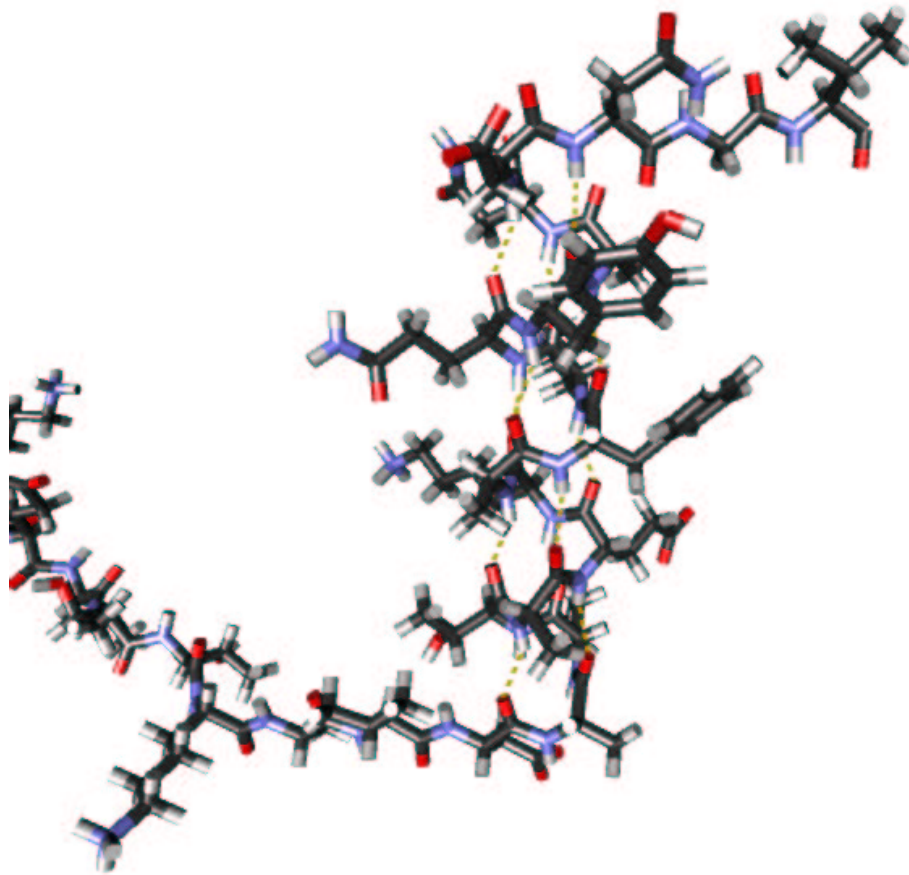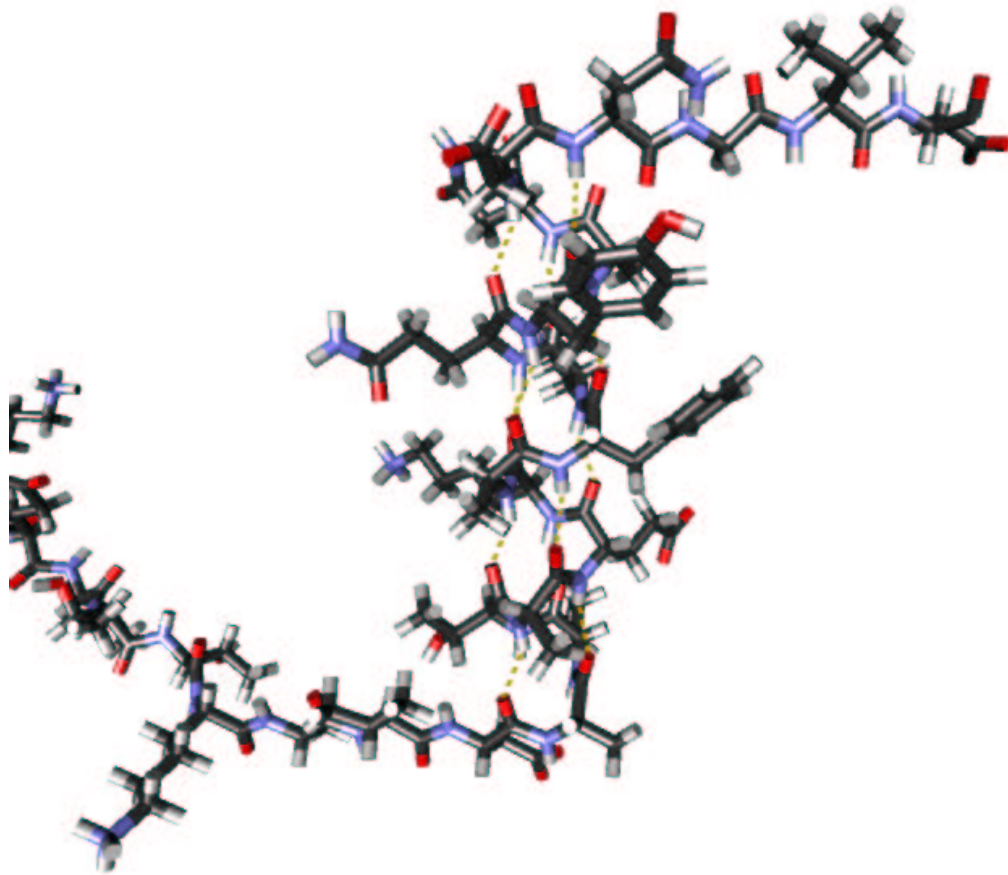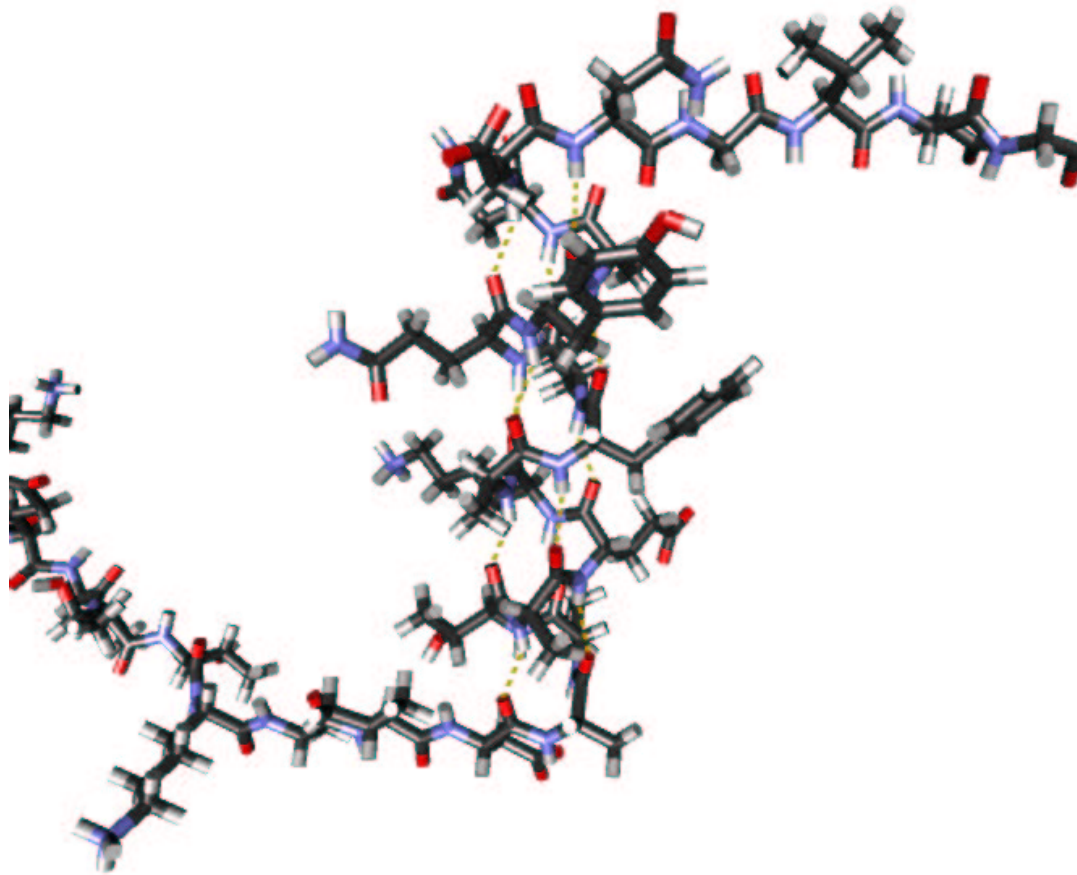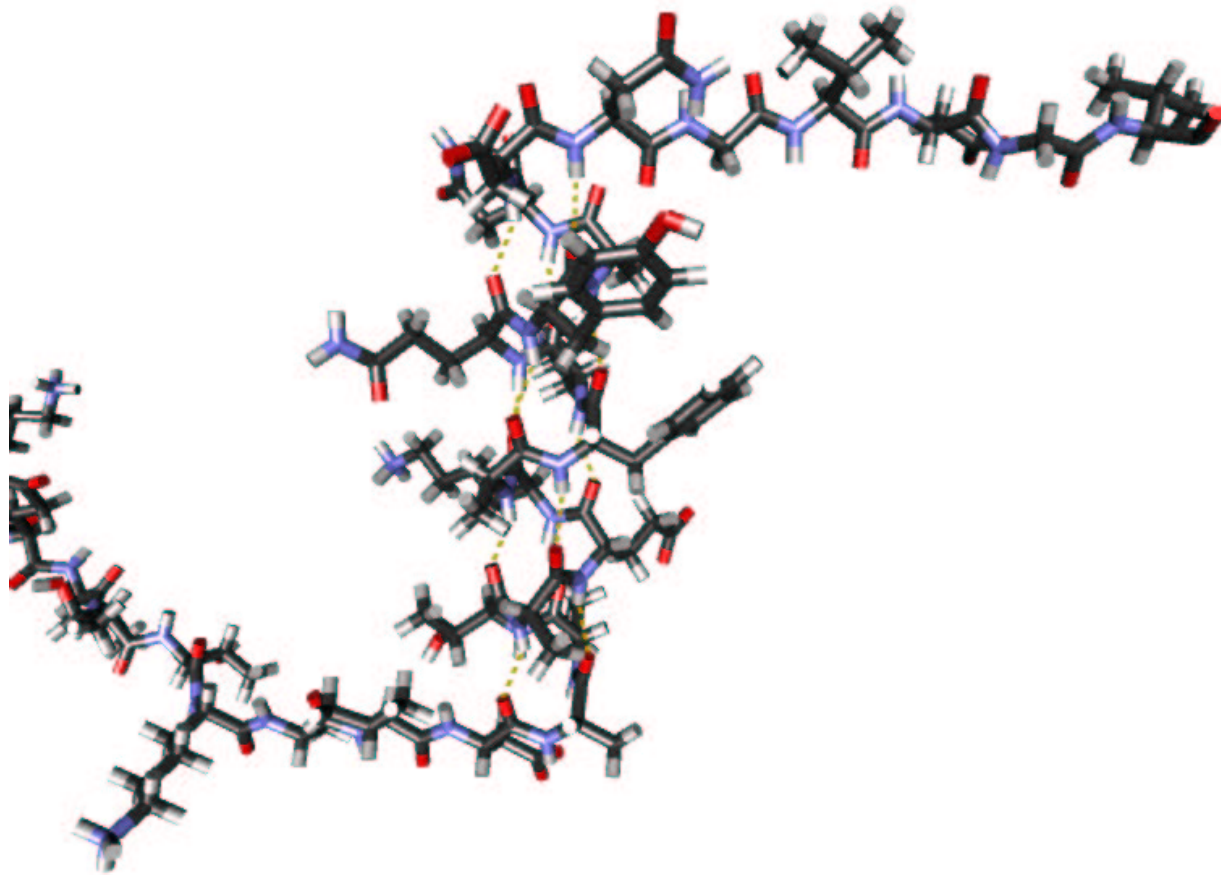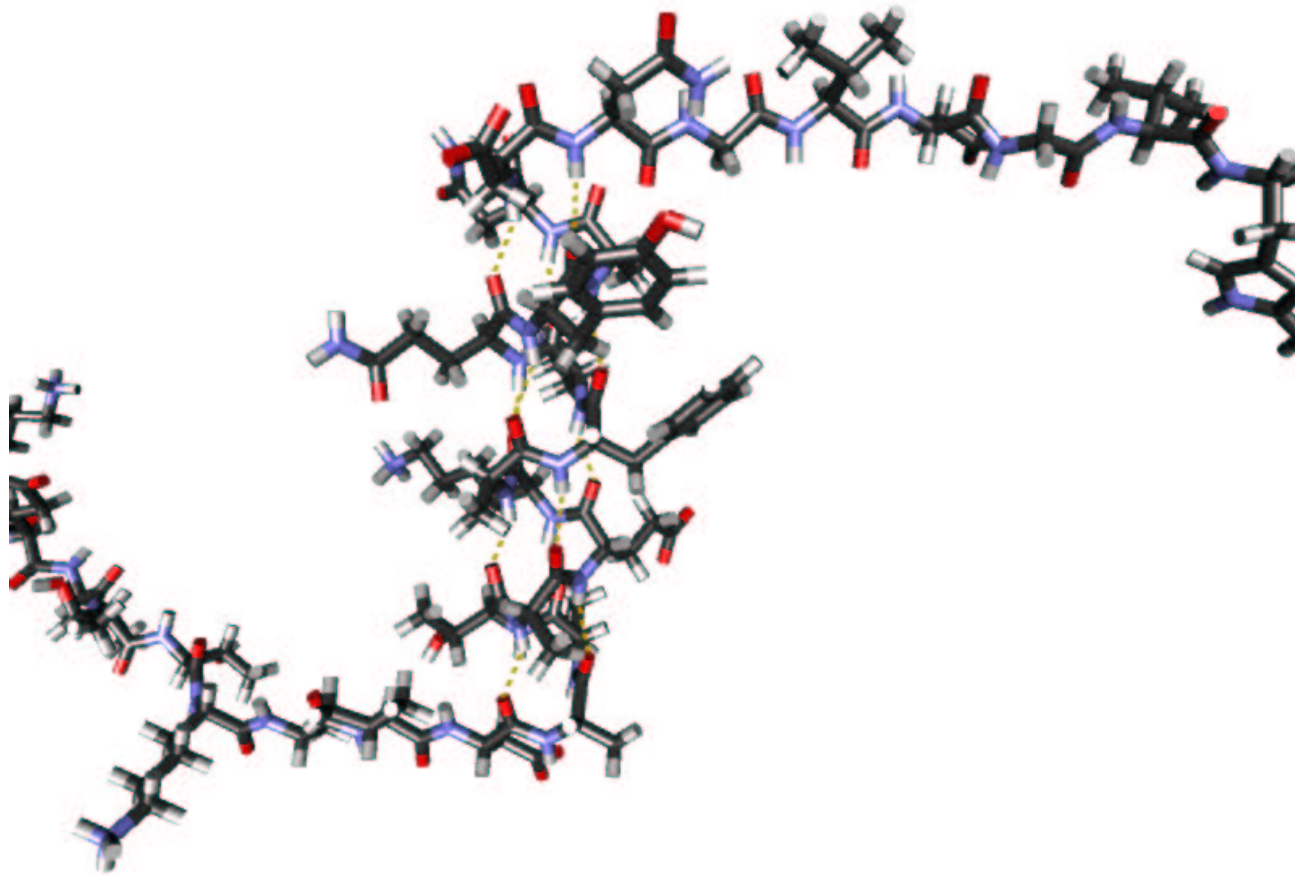
Concatenate Amino Acid Residues

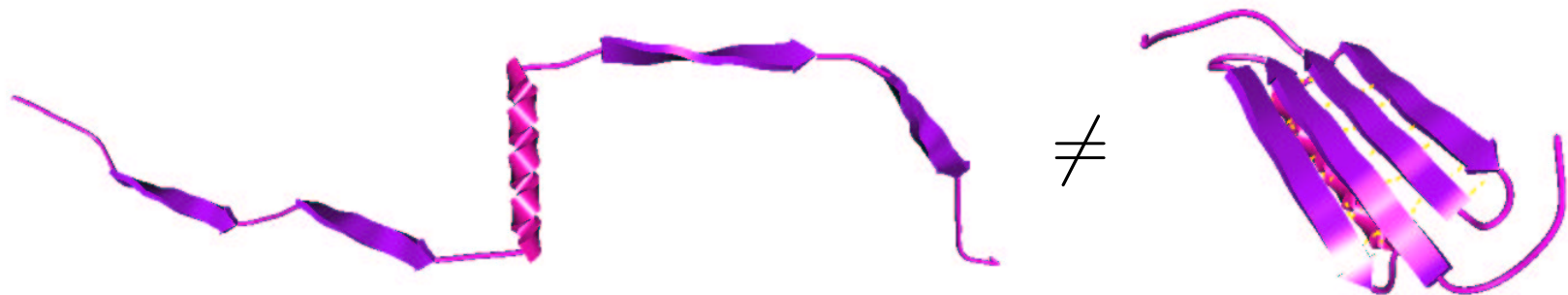# Concatenate Amino Acid Residues

Concatenate Amino Acid Residues

Creating Better Configurations
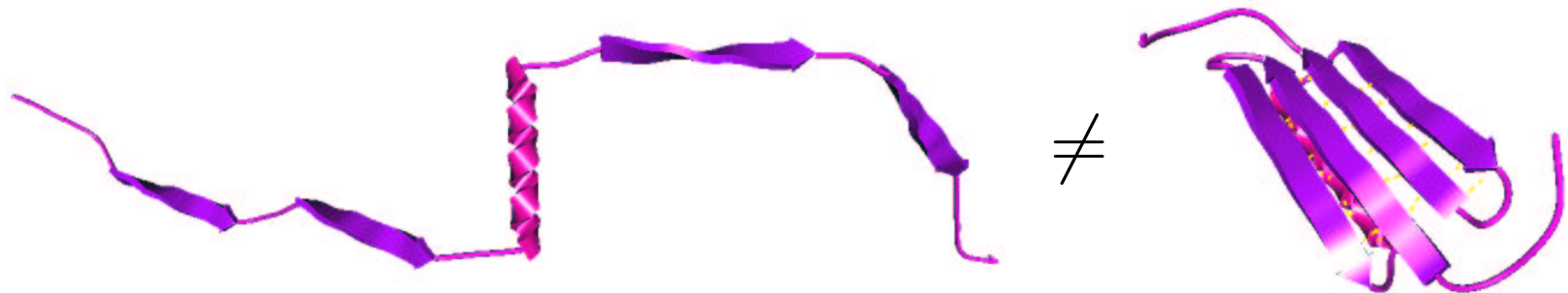
Problem: *Pre-configurations* are not good enough for subsequent global optimization

Creating Better Configurations

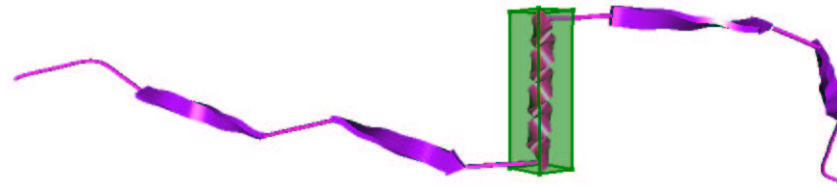Problem: *Pre-configurations* are not good enough for subsequent global optimization

Solution: Use human intuition and interactive manipulation to create sets of better configurations

Manipulating Proteins

- Idea: Let user assemble proteins by dragging substructures

Manipulating Proteins

- Idea: Let user assemble proteins by dragging substructures
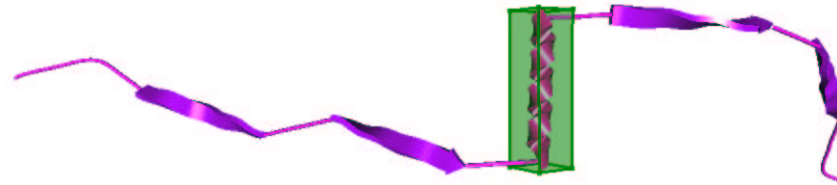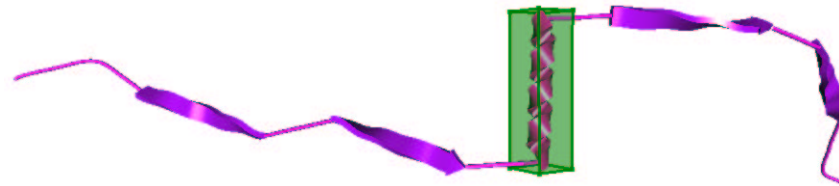

- Problem: When only a part of a protein is moved, bonds will break

Manipulating Proteins

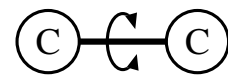- Idea: Let user assemble proteins by dragging substructures

- Problem: When only a part of a protein is moved, bonds will break

- Solution: Use inherent flexibility of protein to preserve chemical structure

Flexibility of Proteins

- Each single covalent bond along the amino acid chain can rotate around its bond axis

- Protein's overall shape is determined by vector of *dihedral angles* (ϕ_i) and (ψ_i)
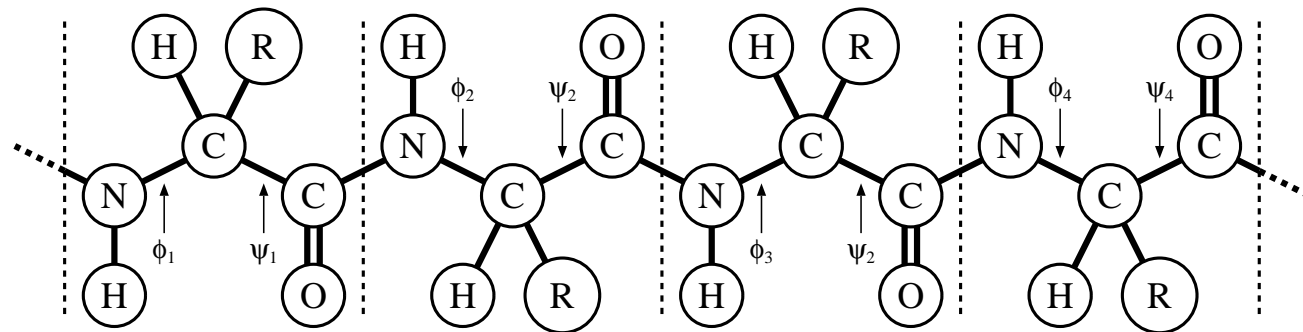
Inverse Kinematics

- Use coil regions between selected structures as movement buffers
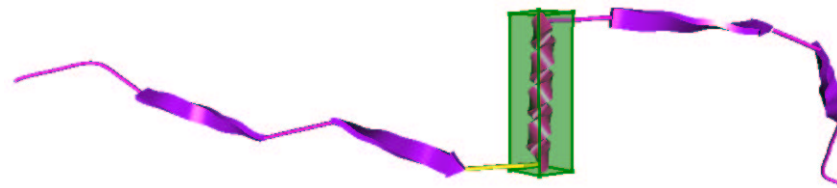
- Use Inverse Kinematics (IK) to translate structure movements into dihedral angle changes

Example Manipulation
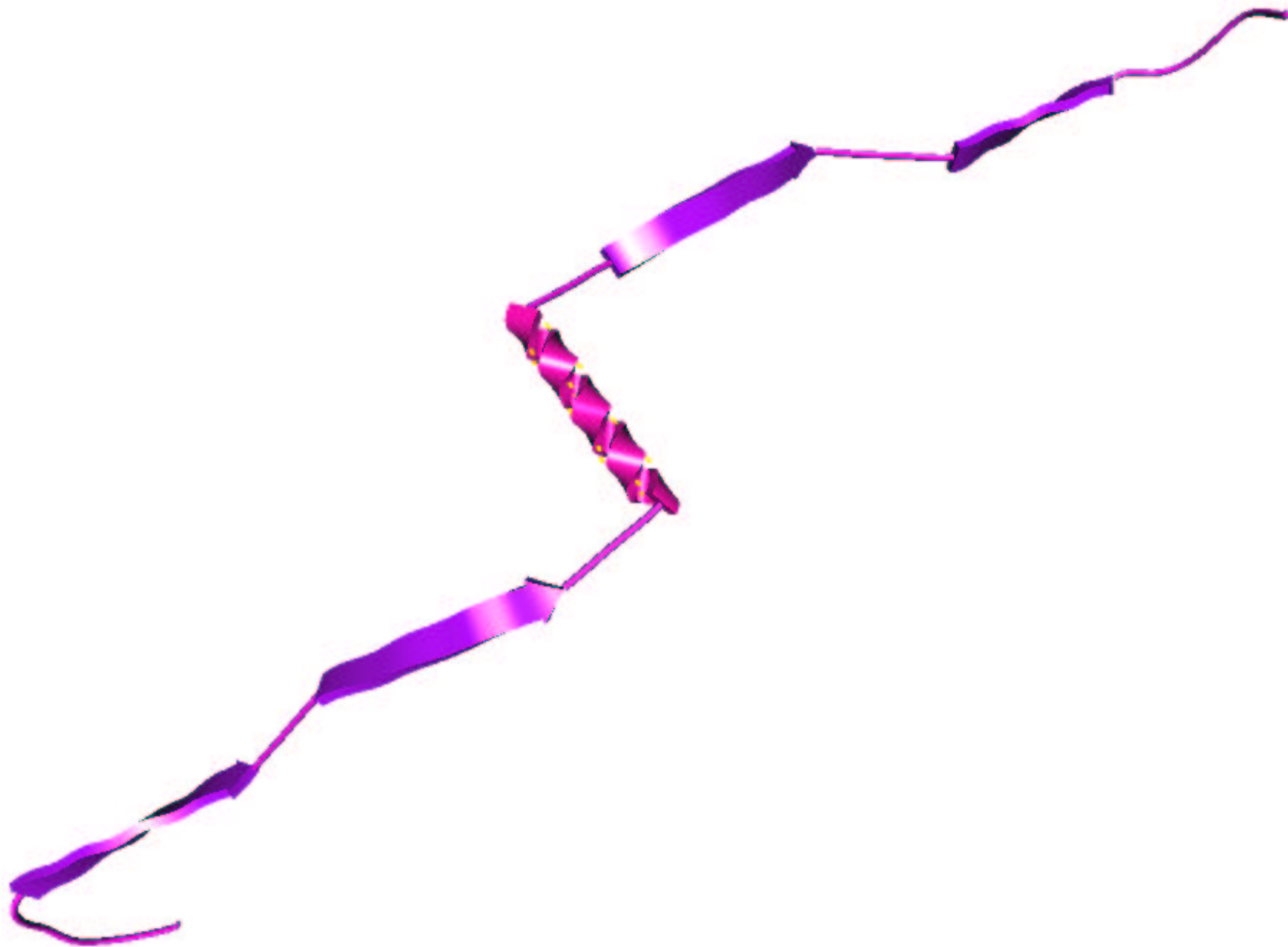
Example Manipulation
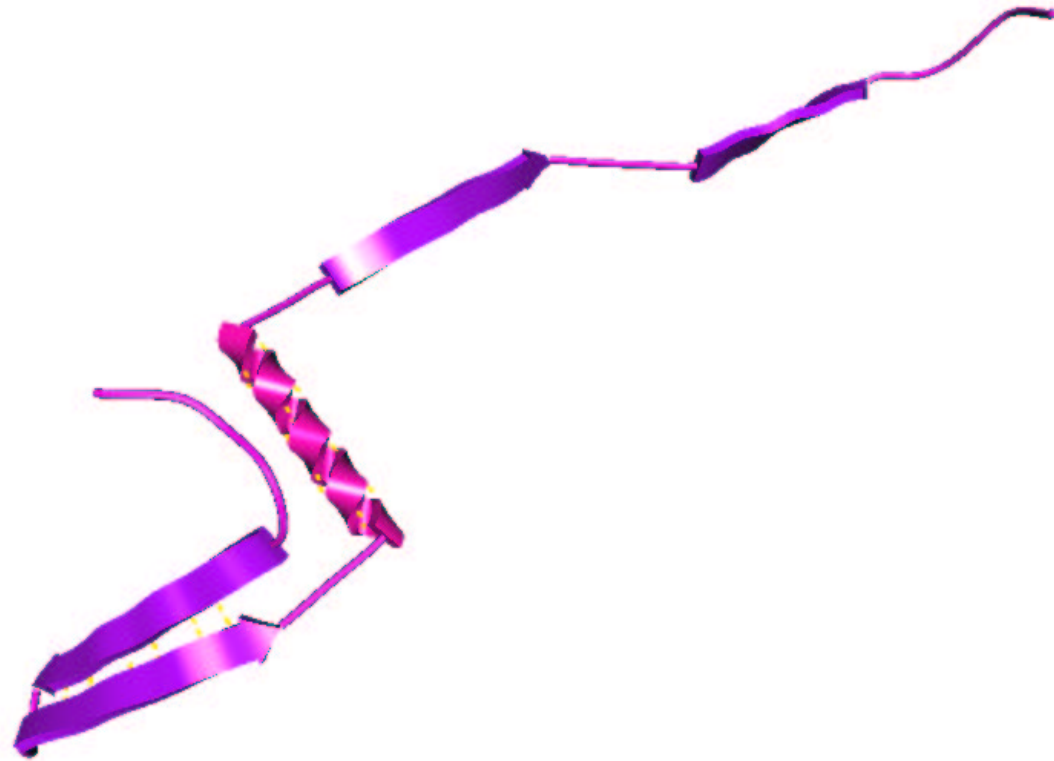
Example Manipulation

Example Manipulation

Example Manipulation

Set of Initial Configurations
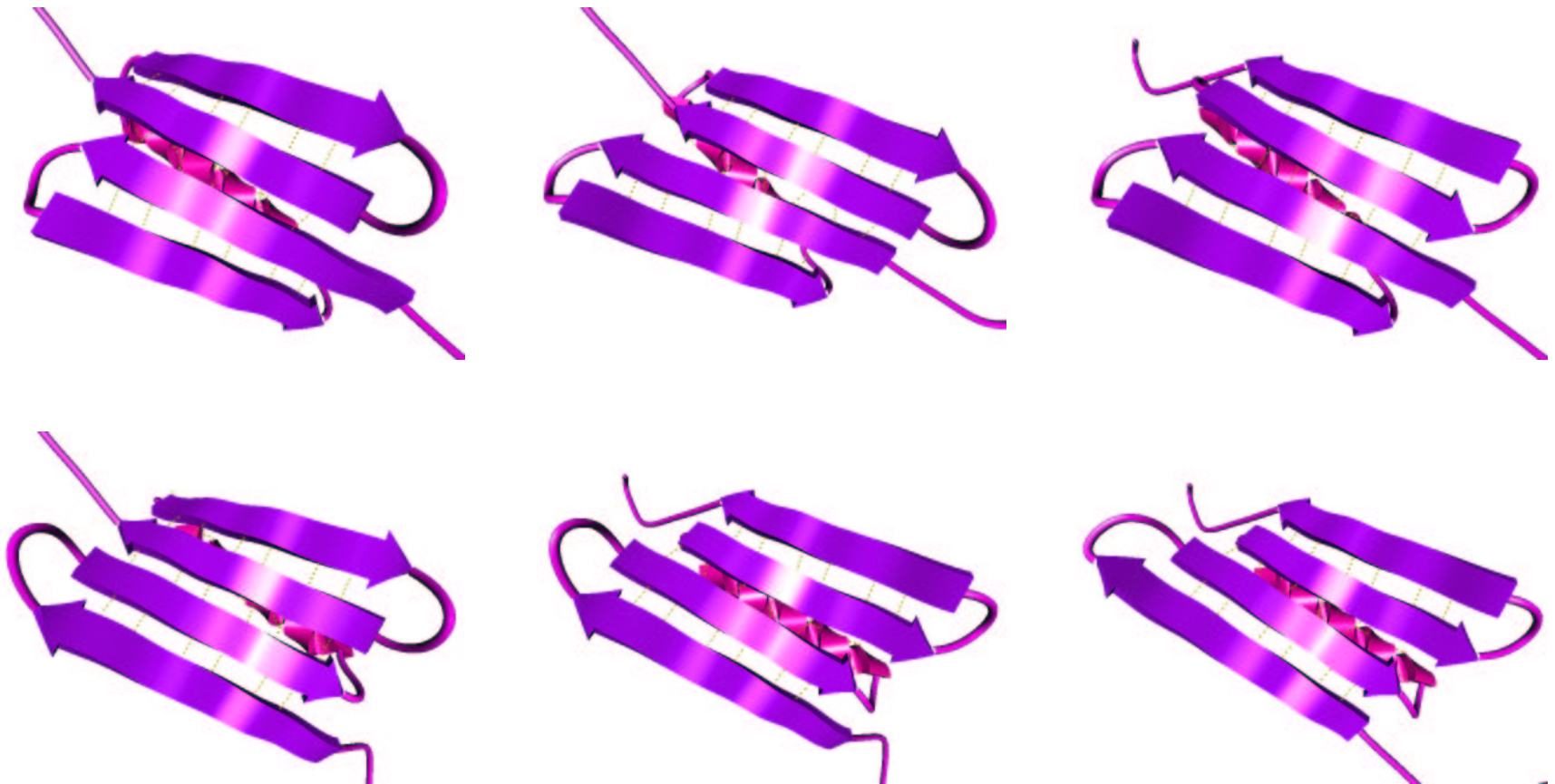
ProtoShop I

ProtoShop is a tool combining

- standard molecular visualization techniques,

- interactive manipulation (using IK),

- interaction-specific visual guides, and

- real-time visual feedback

to allow creating good initial configurations for subsequent protein folding.

ProtoShop II

ProtoShop...

- was used extensively during CASP5 competition
  - Created 20–30 confgurations each for over a dozen large proteins
- has been released to research community: `http://proteinshop.lbl.gov`
- is being evaluated/used by several dozen research groups world-wide

Lessons Learned

- Protein folding needs human intuition

- ProtoShop offers direct method to translate intuition into 3D protein configurations

- More and better configurations can be created in less time

- Improved overall performance

Current and Future Work

- Better user interface for manipulation
  - 6-DOF input devices
  - Virtual reality environment

- Steering of global optimization process itself
  - Monitoring of configuration set
  - On-the-fly manipulation of configurations
  - Pruning of optimization tree

- Visualization of internal energy

- Integration of local energy optimization

Acknowledgments

- Collaborators:
    - University of California, Berkeley
    - University of Colorado, Boulder
    - Lawrence Berkeley National Laboratory
    - Center for Image Processing and Integrated Computing, University of California, Davis

- Funding agencies:
    - DOE, NSF, NPACI, NIH, NIMH, . . .