Constraint Programming and Biology: The protein Folding Problem

Agostino Dovier

Dept. Math and Computer Science, Univ. of Udine, Italy

ACP Summer School in Constraint Programming
Wrocław, September 2012

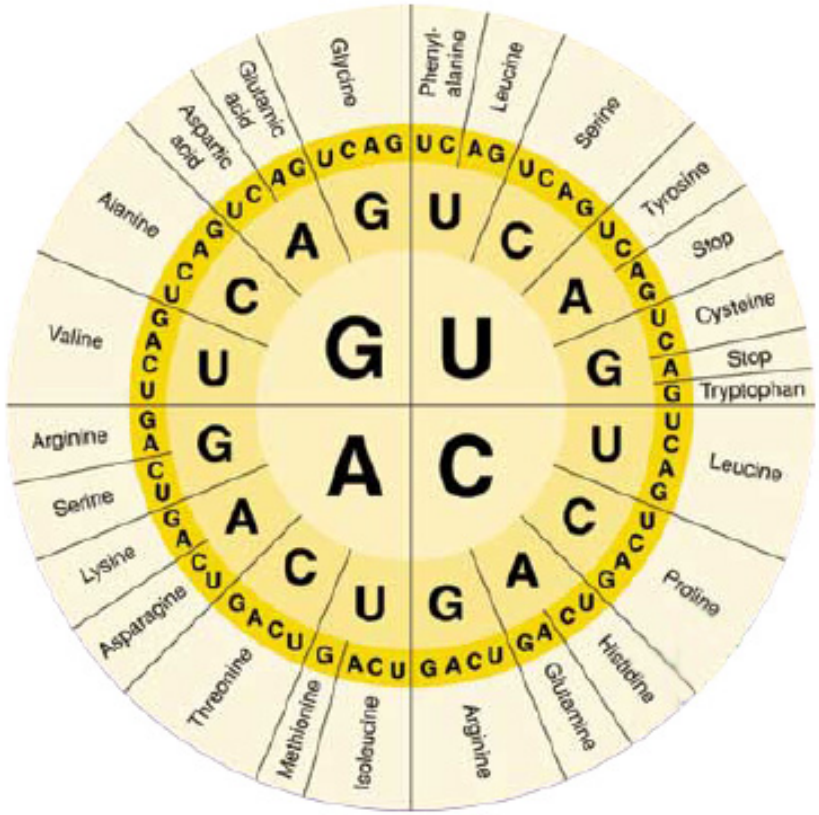
Introduction

- For the sake of completeness we spend some words on the translation stage
- The central dogma is: DNA $\mapsto$ (transcription) RNA $\mapsto$ (translation) protein.
- We have already seen some DNA problems (Haplotype inference/Phylogenetic trees) and one RNA problem (secondary structure prediction).
- Now we focus on the last stage: RNA $\mapsto$ protein, and in particular on the problem of predicting the 3D form of the protein.

Introduction

RNA $\mapsto$ protein

- RNA is a sequence of nucleotides A, C, G, U.
- Triplets of nucleotides identify a new object (monomer) called amino-acid.
- Although in principle we can have $4^3 = 64$ different amino-acids, just 20 (+ 1 rare) of them are generated.



Introduction

Proteins

- A protein is a sequence of amino acids.
- This sequence is also called *primary structure*
- Typical length is 50–500.
- Amino acids are of 20 types, identified by a letter: **Ala**nine (A), **Cys**teine (C), **Asp**artic Acid (D), **Glu**tamic Acid (E), **Phe**nylalanine (F), **Gly**cine (G), **His**tidine (H), **Isole**ucine (I), **Lys**ine (K), **Leu**cine (L), **Met**hionine (M), **As**paragi**n**e (N), **Pro**line (P), **Gl**utami**n**e (Q), **Arg**inine (R), **Ser**ine (S), **Thr**eonine (T), **Val**ine (V), **Tryp**tophan (W), **Tyr**osine (Y).

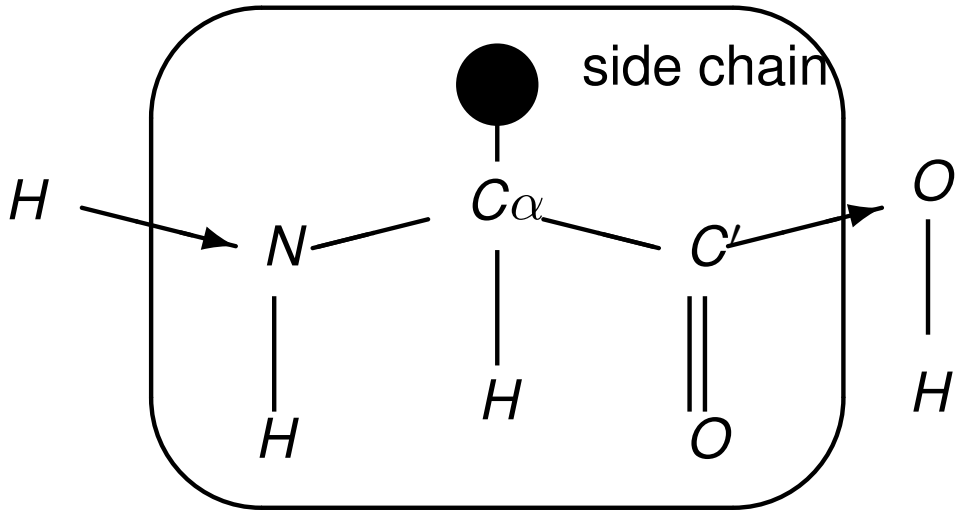
Introduction

Proteins

- A protein is a sequence of amino acids.
- This sequence is also called *primary structure*
- Typical length is 50–500.
- Amino acids are of 20 types, identified by a letter: **Ala**nine (A), **Cys**teine (C), **Asp**artic Acid (D), **Glu**tamic Acid (E), **Phe**nylalanine (F), **Gly**cine (G), **His**tidine (H), **Isole**ucine (I), **Lys**ine (K), **Leu**cine (L), **Met**hionine (M), **As**paragi**n**e (N), **Pro**line (P), **Gl**utami**n**e (Q), **Arg**inine (R), **Ser**ine (S), **Thr**eonine (T), **Val**ine (V), **Tryp**tophan (W), **Tyr**osine (Y).
- Mathematically: The primary structure of amino acids is a string of $(\{A, \ldots, Z\} \setminus \{B, J, O, U, X, Z\})^*$.

Introduction

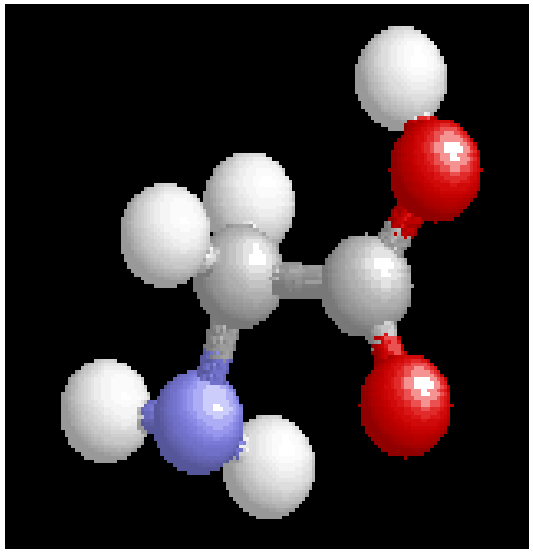
Aminoacid structure

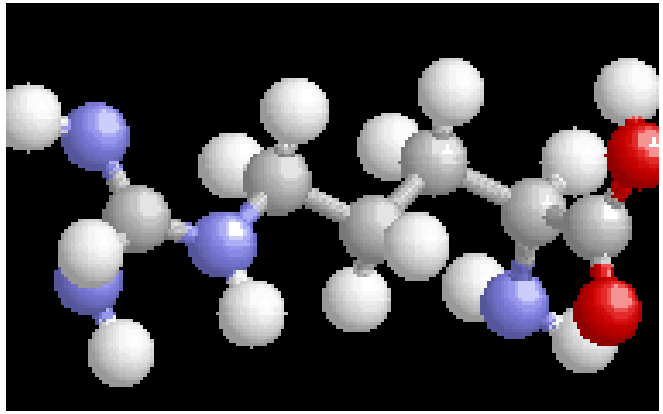


- The *backbone* is the same for all aminoacids
- The *side chain* characterizes an aminoacids.
- A side chain contains from 1 (Glycine) up to 18 (Tryptophan) atoms.

Amino acids

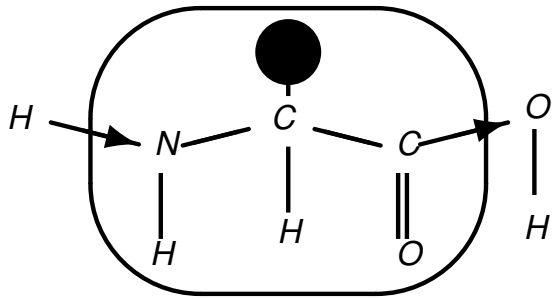
Glycine e Arginine





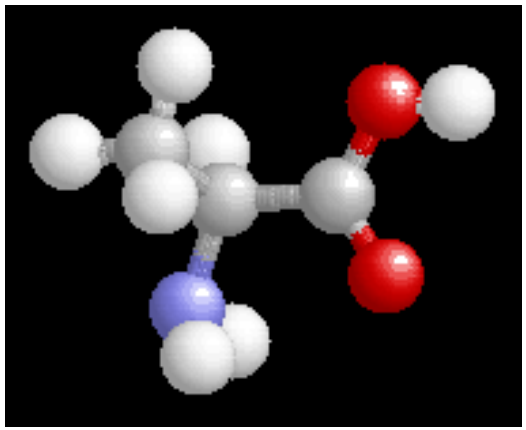
$C_2H_5NO_2 \rightarrow 10$ atoms $\quad C_6H_{14}N_4O_2 \rightarrow 26$ atoms

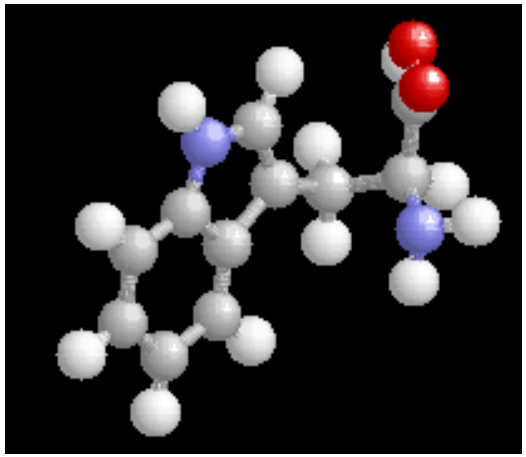
White $=$ H
Blue $=$ N
Red $=$ O
Grey $=$ C



Amino acids

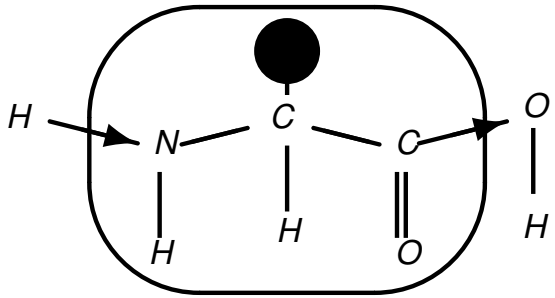
Alanine e Tryptophan





$C_3H_7NO_2 \rightarrow$ 13 atoms $\quad C_{11}H_{12}N_2O_2 \rightarrow$ 27 atoms

White $=$ H
Blue $=$ N
Red $=$ O
Grey $=$ C



Amino acids

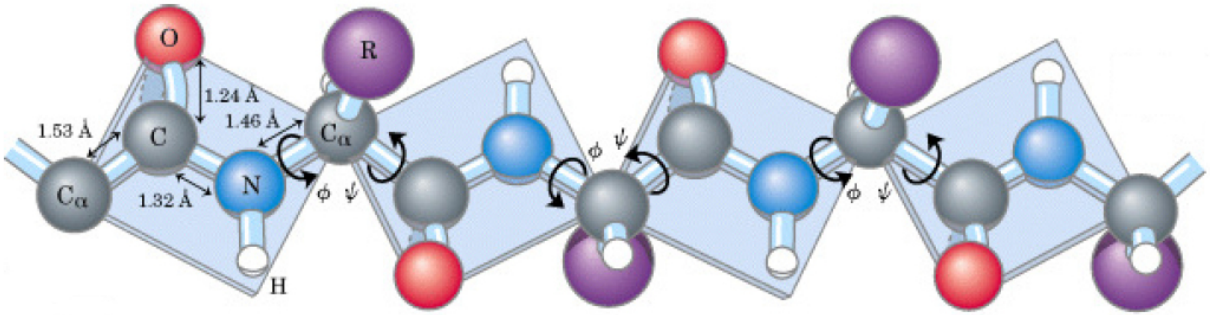
Chemical composition (for completeness)

Name	Chemical	Side Chain
A	$C_3H_7NO_2$	4
C	$C_3H_7NO_2S$	4
D	$C_4H_7NO_4$	16
E	$C_5H_9NO_4$	10
F	$C_9H_{11}NO_2$	14
G	$C_2H_5NO_2$	1
H	$C_6H_9N_3O_2$	11
I	$C_6H_{13}NO_2$	13
K	$C_6H_{14}N_2O_2$	15
L	$C_6H_{13}NO_2$	13

Name	Chemical	Side Chain
M	$C_5H_{11}NO_2S$	11
N	$C_4H_8N_2O_3$	8
P	$C_5H_9NO_2$	$8(*)$
Q	$C_5H_{10}N_2O_3$	11
R	$C_6H_{14}N_4O_2$	17
S	$C_3H_7NO_3$	5
T	$C_4H_9NO_3$	9
Y	$C_9H_{11}NO_3$	15
V	$C_5H_{11}NO_2$	10
W	$C_{11}H_{12}N_2O_2$	18

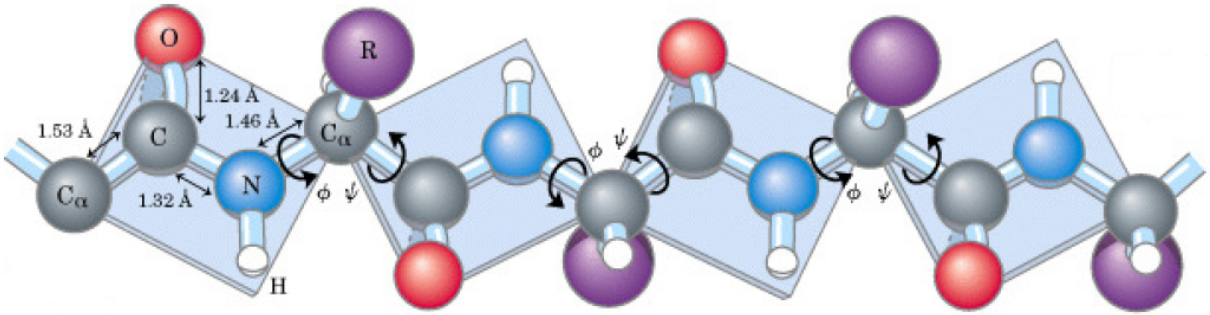
Proteins

- Amino acids combine sequentially to form a flexible chain



Proteins

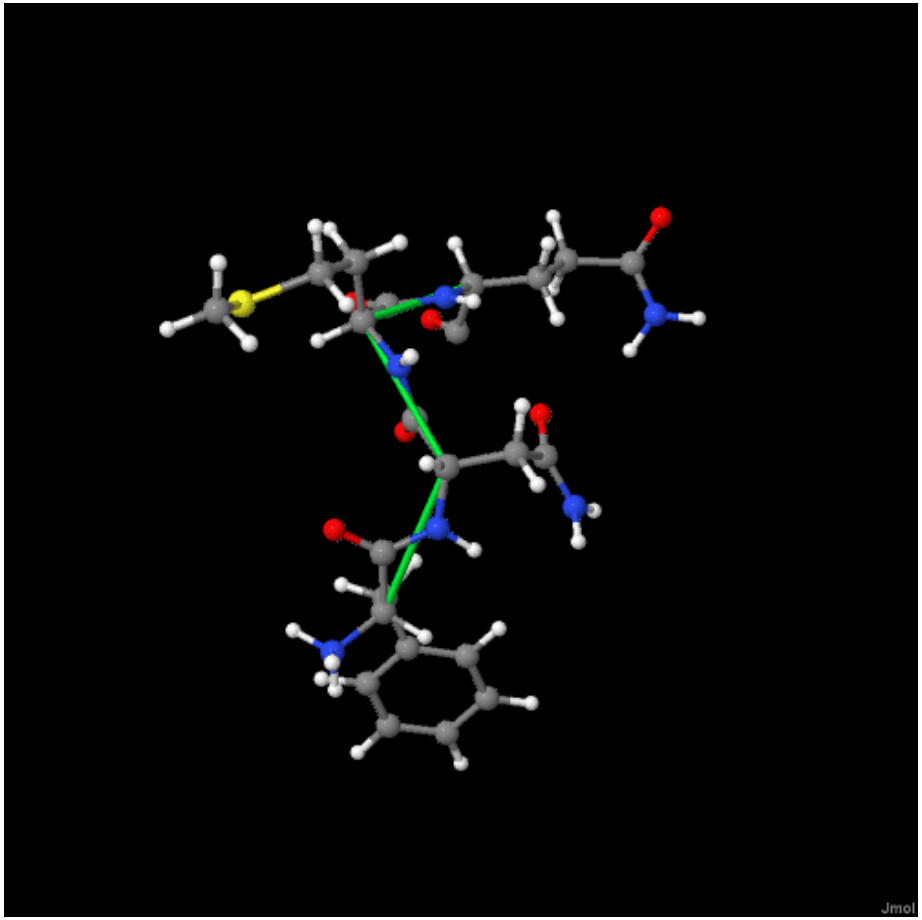
- Amino acids combine sequentially to form a flexible chain



- There are several degrees of freedom (black arrows)
- The purple dots represent the *side chains*, that is a sub molecule that depends on the amino acid type

Proteins

Simplified structure

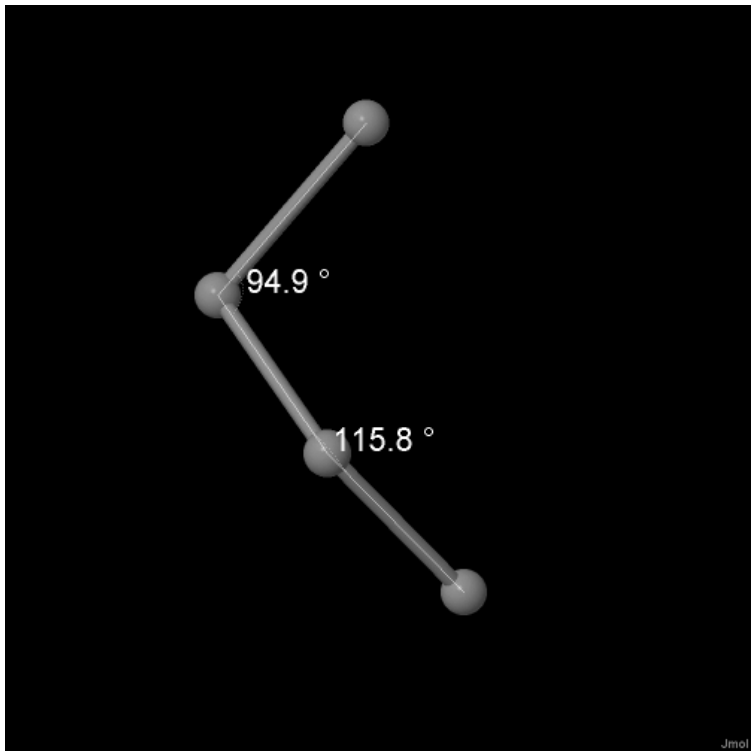


- The *backbone* (green) links consecutive amino acids
- Each amino acid can be identified by its carbon alpha (C_α)
- The distance between consecutive $C\alpha$ points is constant (3.81Å).

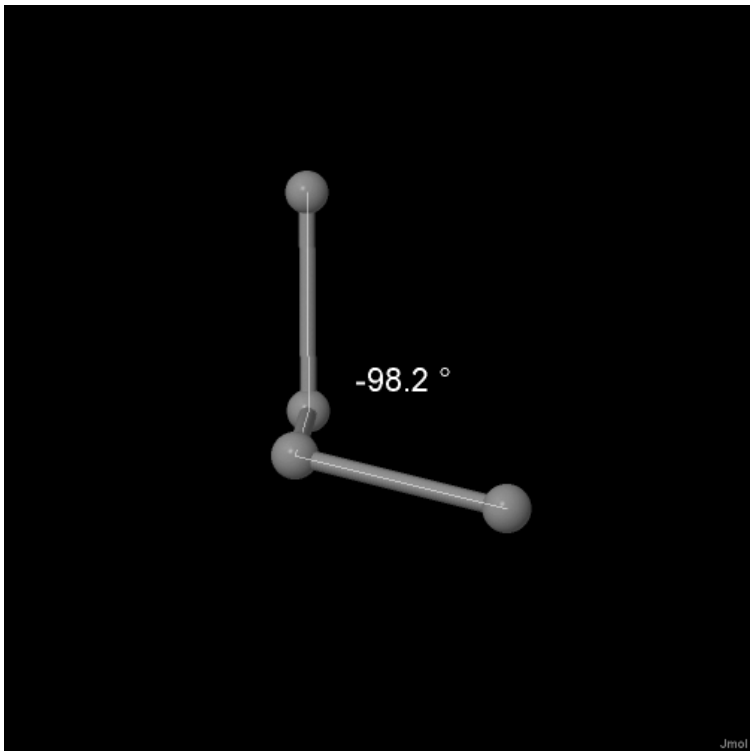
Proteins

Backbone

- The distance between $C\alpha$ points is constant (3.81Å).
- Let us consider a backbone made of 4 amino acids.



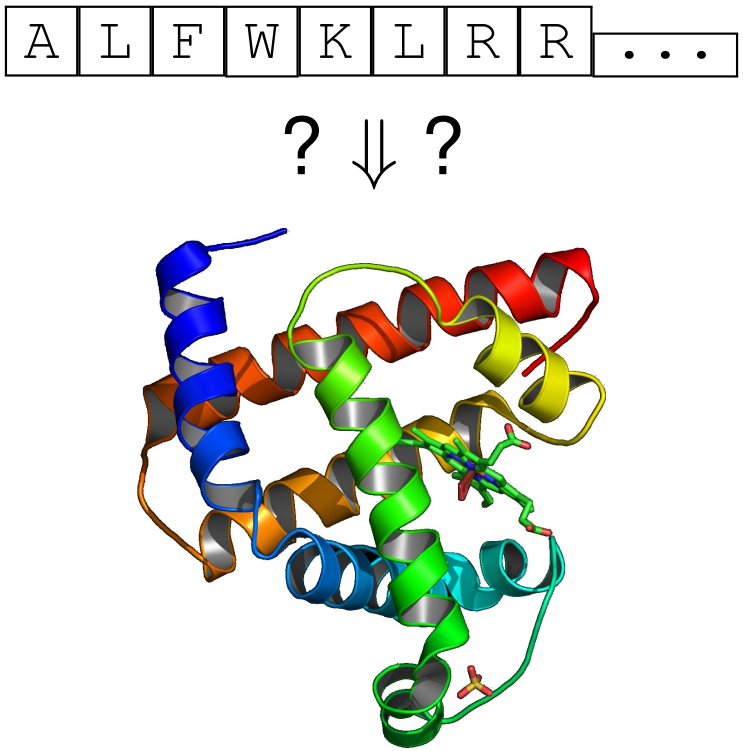
There are two bend angles



And one torsional angle

The structure prediction problem

- Given the primary structure of a protein (its amino acid sequence)
- For each amino acid, output its position in the space (tertiary structure of a protein)



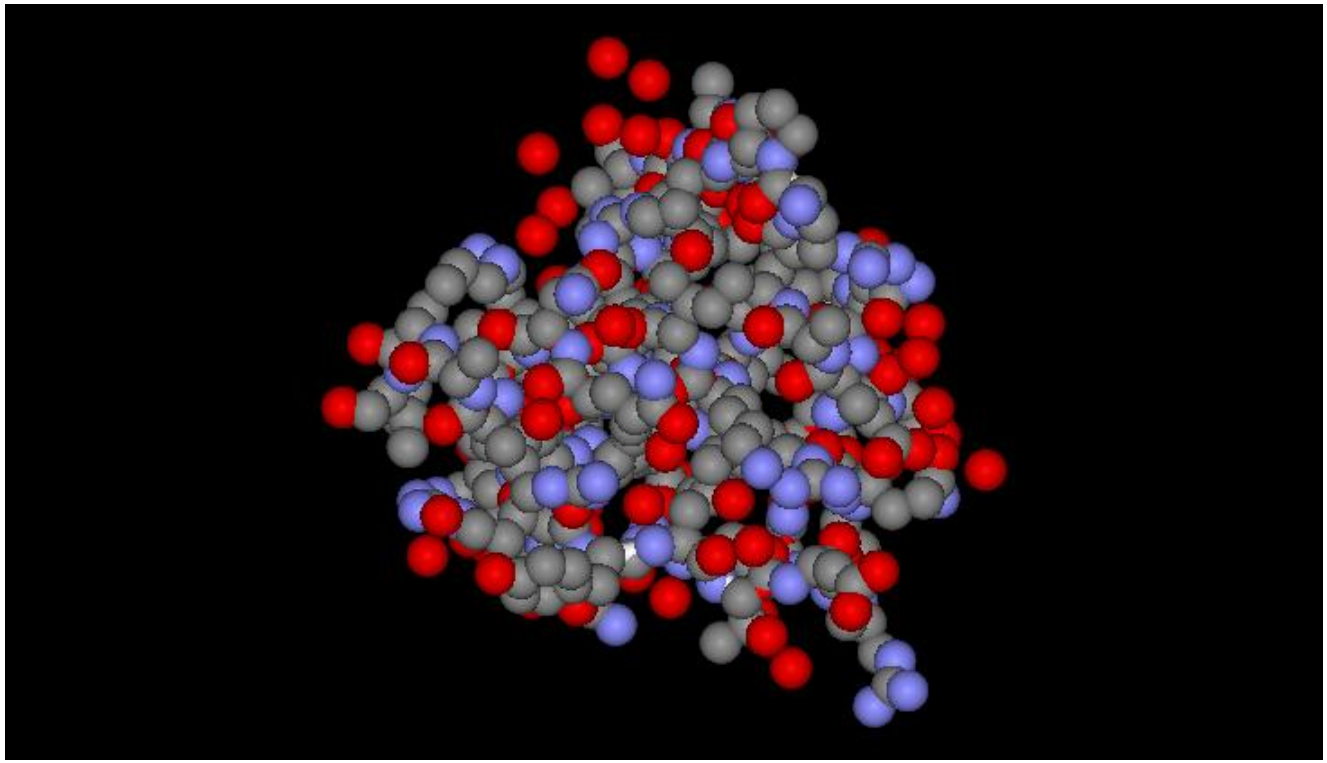
The structure prediction problem

... and this is the hard part:

- In nature a protein has a unique/stable 3D conformation
- A cost function (that mimics physics laws) can be used to score each conformation
- Searching for the optimal score produces the best candidate is difficult (NP-complete even in extremely simplified modelings)
- Abstractions are either spatial or energetic.

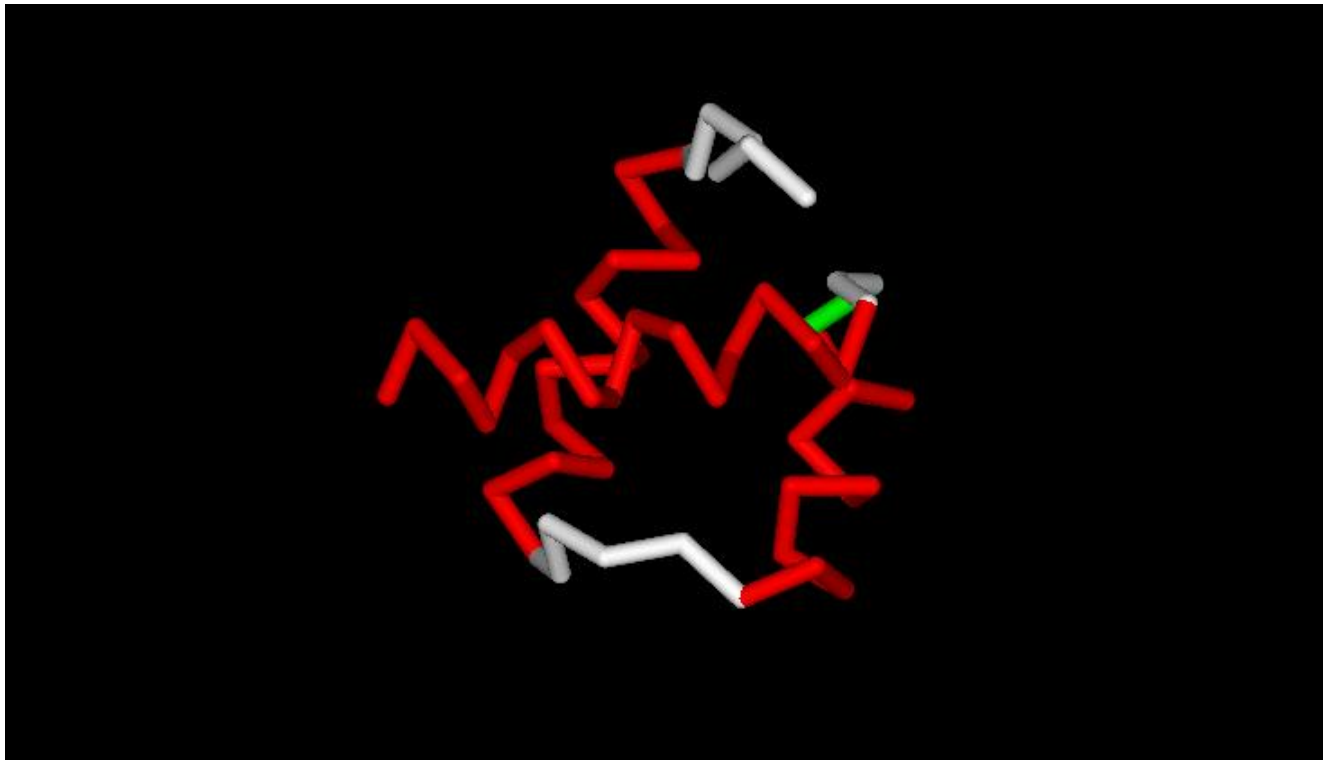
The structure prediction problem

Primary: rprtafsseqlarlkrefnenrylterrrqqlsselglneaqikiwfqnkraki



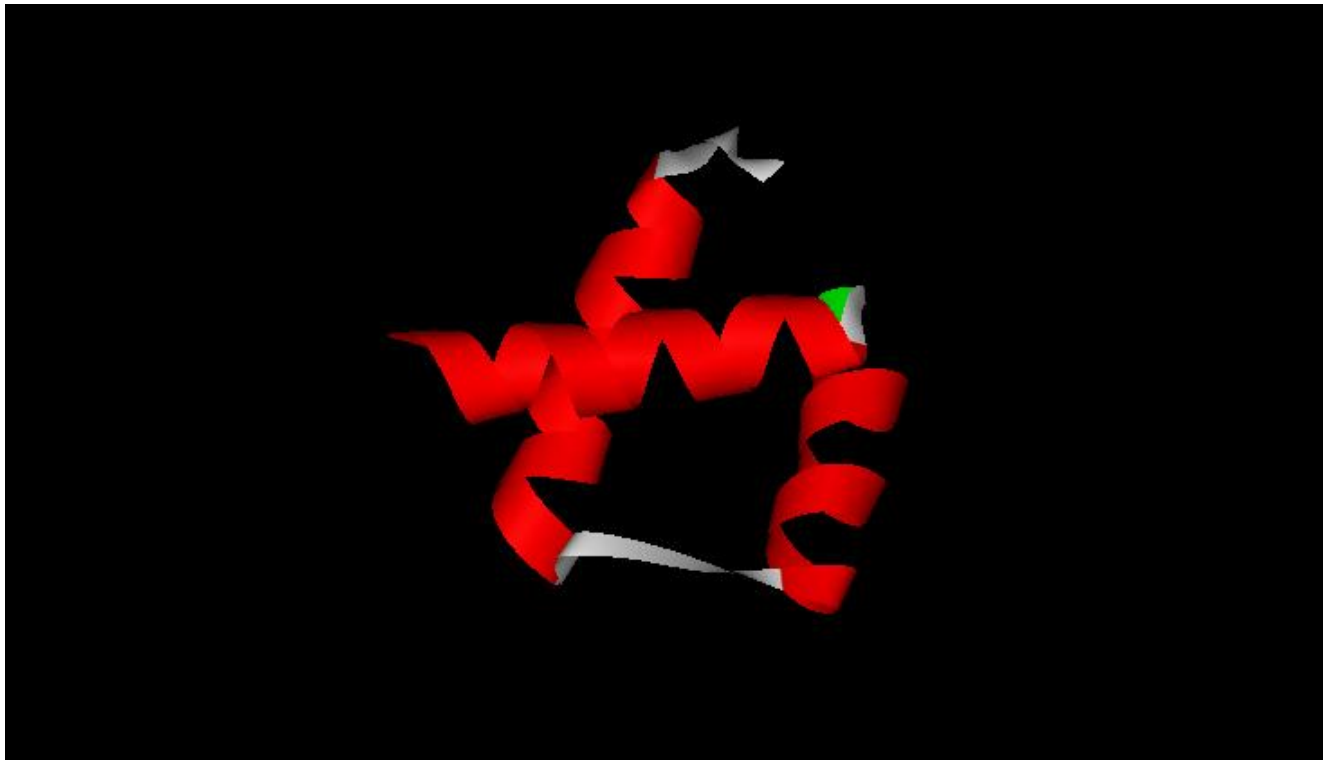
The structure prediction problem

Primary: rprtafsseqlarlkrefnenrylterrrqqlsselglneaqikiwfqnkraki



The structure prediction problem

Primary: rprtafsseqlarlkrefnenrylterrrqqlsselglneaqikiwfqnkraki



The protein folding problem

- Let $\mathcal{D}$ be a set of admissible points for the amino acids (for which part of the aminoacid? E.g., the $C\alpha$ atom—First simplification)
- Let c, d two fixed distances. For two points $p, q \in \mathcal{D}$, we say that next(p, q) if and only if $|p - q| = d$ ($d = 3.8$Å)
- A function $\omega : \{1, \ldots, n\} \longrightarrow \mathcal{D}$ is said a *folding* if
  - for $i, j \in \{1, \ldots, n\}$ if $i \neq j$ then $|\omega(i) - \omega(j)| \geq d$ (or simply $\omega(i) \neq \omega(j)$ in lattice)
  - for $i \in \{1, \ldots, n-1\}$ it holds that next$(\omega(i), \omega(i+1))$

The protein folding problem

- For two points $p, q \in \mathcal{D}$, we define the Boolean function contact as follows: contact$(p, q) = 1$ if and only if $|p - q| \leq c$ (Second simplification: contact energy)
- Let Pot be a function from pairs of amino acids to integer numbers. The free energy of a folding $E(\omega, \vec{s})$ is computed as follows:

$$E(\omega, \vec{s}) = \sum_{\substack{1 \leq i < n \\ i+2 \leq j \leq n}} \text{contact}(\omega(i), \omega(j))\text{Pot}(s_i, s_j)$$

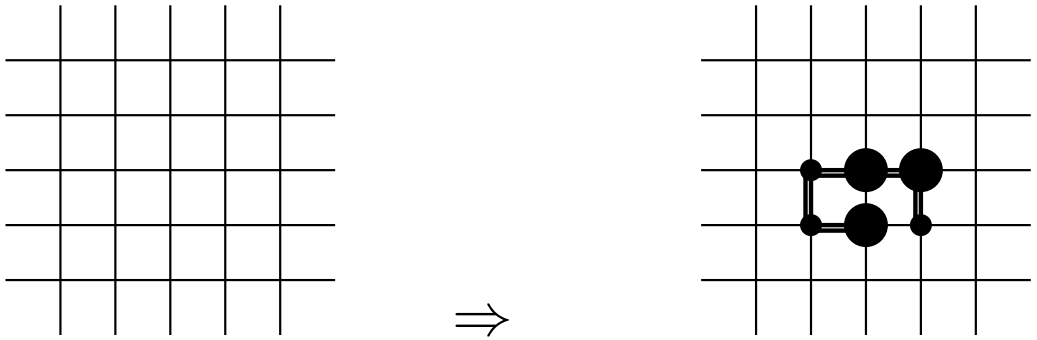
- The protein structure prediction problem (PSP) is the problem of determining the folding(s) ω with minimal energy.

Lattice models

- The fixed distance between $C\alpha$s (and some statistics on the most frequent angles) prompts for using discrete lattice models.

Lattice models

- The fixed distance between $C\alpha$s (and some statistics on the most frequent angles) prompts for using discrete lattice models.
- Just to play with, let us start with $\mathbb{N}^2$



- A folding is just the writing (e.g. by pencil) of a line of length $n-1$ on the paper following the N^2 grid. Different circle size for different aminoacids. Difficult for 20 kind of amino acids, but ...

Simplified (contact) energy

The HP Energy model (Dill)

- Aminoacids are split into two families: Hydrophobics (H) and Polar (P)
- Two aminoacids of kind “H” in contact at distance *d* (lattice unit) contribute with an energy of -1, otherwise the contribution is 0.

Simplified (contact) energy

The HP Energy model (Dill)

- Aminoacids are split into two families: Hydrophobics (H) and Polar (P)
- Two aminoacids of kind “H” in contact at distance *d* (lattice unit) contribute with an energy of -1, otherwise the contribution is 0.
- This formulation of the problem has a nice property: you can teach it to a children without speaking of proteins and so on: *Write a folding using paper and pencil that maximizes the contacts between “H” aminoacids (large circles)*

Simplified (contact) energy

The HP Energy model (Dill)

- Aminoacids are split into two families: Hydrophobics (H) and Polar (P)
- Two aminoacids of kind "H" in contact at distance d (lattice unit) contribute with an energy of -1, otherwise the contribution is 0.
- This formulation of the problem has a nice property: you can teach it to a children without speaking of proteins and so on: *Write a folding using paper and pencil that maximizes the contacts between "H" aminoacids (large circles)*
- With $\mathbb{N}^2$ and HP, establishing whether there is a folding with energy $< k$ is NP-complete (Crescenzi, Goldman, Papadimitriou, Piccolboni, Yannakakis. On the Complexity of Protein Folding. Journal of Computational Biology 5(3): 423-466 (1998))

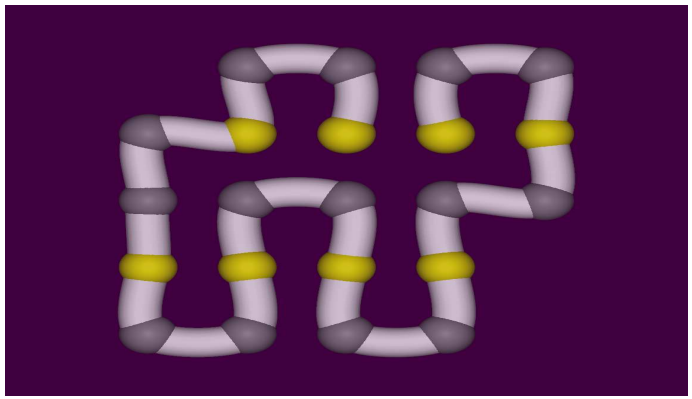
Simplified (contact) energy

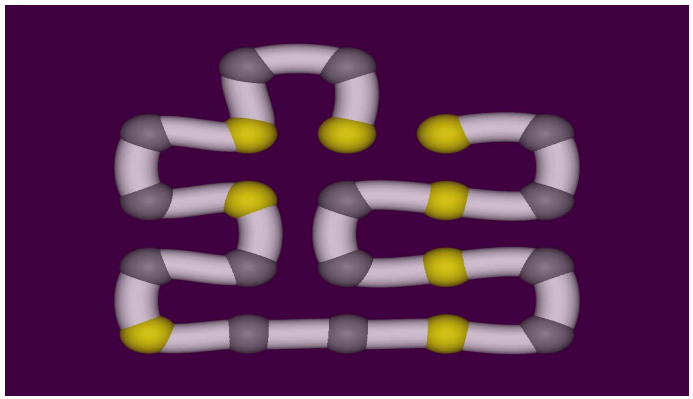
The HP Energy model (Dill)

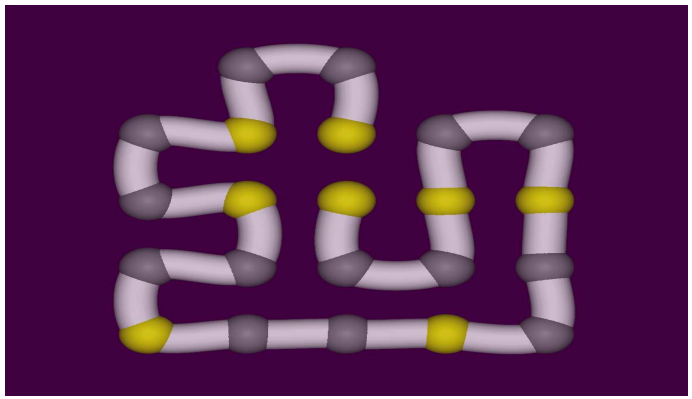
- Aminoacids are split into two families: Hydrophobics (H) and Polar (P)
- Two aminoacids of kind "H" in contact at distance d (lattice unit) contribute with an energy of -1, otherwise the contribution is 0.
- This formulation of the problem has a nice property: you can teach it to a children without speaking of proteins and so on: *Write a folding using paper and pencil that maximizes the contacts between "H" aminoacids (large circles)*
- With $\mathbb{N}^2$ and HP, establishing whether there is a folding with energy $< k$ is NP-complete (Crescenzi, Goldman, Papadimitriou, Piccolboni, Yannakakis. On the Complexity of Protein Folding. Journal of Computational Biology 5(3): 423-466 (1998))
- Combinatorial exercise. Assume to fix the first point: what is the size of the search space?
  (hint: it is the number of self-avoiding-walks)

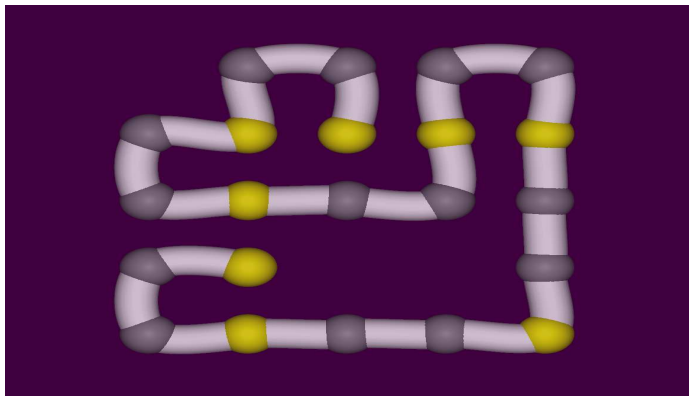
Example of PF HP N^2

Yellow: H, Grey: P. All foldings have energy -6









HP on $\mathbb{N}^2$: FD encoding

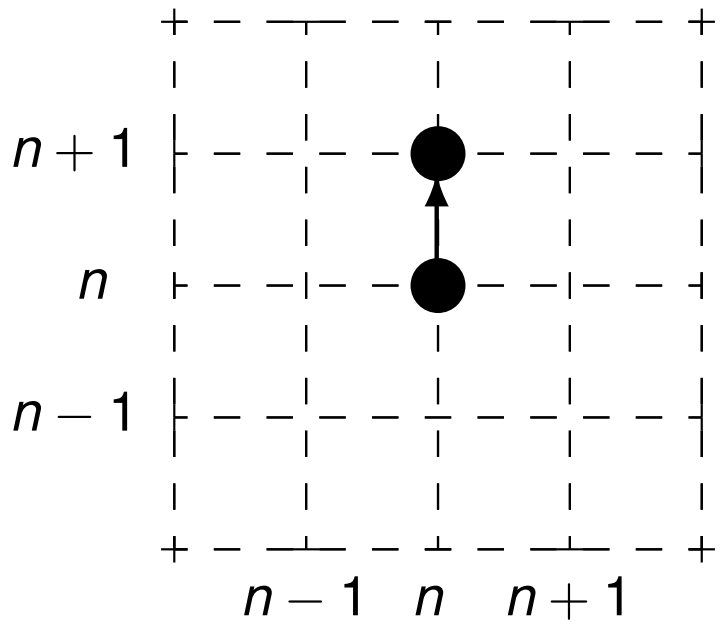
- Primary $= [a_1, \ldots, a_n] = [h/p, p/p, h/p, ...]$
- Tertiary$_x = [X_1, \ldots, X_n]$, Tertiary$_y = [Y_1, \ldots, Y_n]$

HP on $\mathbb{N}^2$: FD encoding

- Primary $= [a_1, \ldots, a_n] = [h/p, p/p, h/p, ...]$
- Tertiary$_x = [X_1, \ldots, X_n]$, Tertiary$_y = [Y_1, \ldots, Y_n]$
- W.l.o.g., let $X_1 = X_2 = Y_1 = n$, $Y_2 = n + 1$.

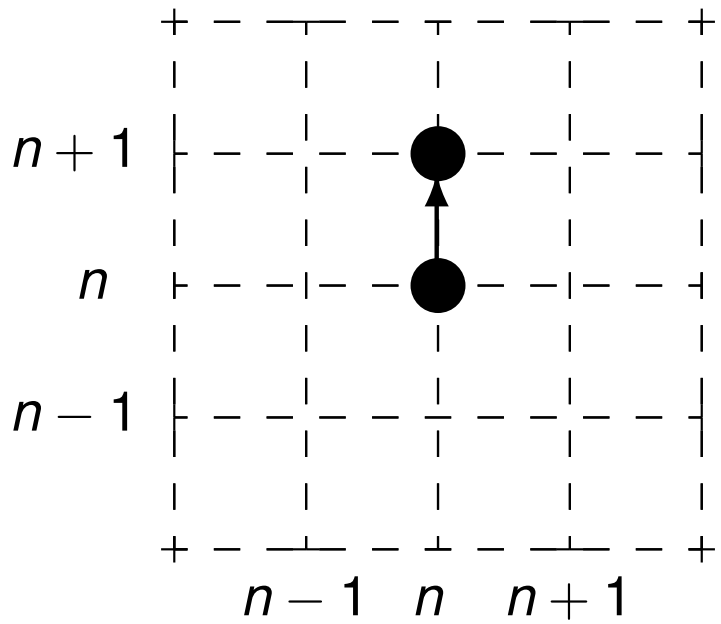
HP on $\mathbb{N}^2$: FD encoding

- Primary $= [a_1, \ldots, a_n] = [h/p, p/p, h/p, ...]$
- Tertiary$_x = [X_1, \ldots, X_n]$, Tertiary$_y = [Y_1, \ldots, Y_n]$
- W.l.o.g., let $X_1 = X_2 = Y_1 = n$, $Y_2 = n + 1$.
- Namely, we start with



HP on $\mathbb{N}^2$: FD encoding

- Primary $= [a_1, \ldots, a_n] = [h/p, p/p, h/p, ...]$
- Tertiary$_x = [X_1, \ldots, X_n]$, Tertiary$_y = [Y_1, \ldots, Y_n]$
- W.l.o.g., let $X_1 = X_2 = Y_1 = n$, $Y_2 = n + 1$.
- Namely, we start with



- $\mathrm{dom}(X_1) = \cdots = \mathrm{dom}(X_n) = \mathrm{dom}(Y_1) = \cdots = \mathrm{dom}(Y_n) = 1..2n$

HP on $\mathbb{N}^2$: FD encoding

- $\text{Tertiary}_x = [X_1, \ldots, X_n], \text{Tertiary}_y = [Y_1, \ldots, Y_n]$
- next: for $i = 1, \ldots, n-1$: $|X_i - X_{i+1}| + |Y_i - Y_{i+1}| = 1$
- saw: for $i = 1, \ldots, n-1$, for $j = i+1, \ldots, n$: $|X_i - X_i| + |Y_i - Y_j| \geq 1$

HP on $\mathbb{N}^2$: FD encoding

- $\text{Tertiary}_x = [X_1, \ldots, X_n], \text{Tertiary}_y = [Y_1, \ldots, Y_n]$
- next: for $i = 1, \ldots, n-1$: $|X_i - X_{i+1}| + |Y_i - Y_{i+1}| = 1$
- saw: for $i = 1, \ldots, n-1$, for $j = i+1, \ldots, n$: $|X_i - X_i| + |Y_i - Y_j| \geq 1$
- We want to express that $(X_i, Y_i) \neq (X_j, Y_j)$. Can we use alldifferent?

HP on $\mathbb{N}^2$: FD encoding

- $\text{Tertiary}_x = [X_1, \ldots, X_n], \text{Tertiary}_y = [Y_1, \ldots, Y_n]$
- next: for $i = 1, \ldots, n-1$: $|X_i - X_{i+1}| + |Y_i - Y_{i+1}| = 1$
- saw: for $i = 1, \ldots, n-1$, for $j = i+1, \ldots, n$: $|X_i - X_i| + |Y_i - Y_j| \geq 1$
- We want to express that $(X_i, Y_i) \neq (X_j, Y_j)$. Can we use alldifferent?
- Let $[P_1, \ldots, P_n]$ be a list and M a "big" integer (100 is ok for us).
- for $i = 1, \ldots, n-1$: $P_i = X_i + MY_i$.

HP on $\mathbb{N}^2$: FD encoding

- $\text{Tertiary}_x = [X_1, \ldots, X_n], \text{Tertiary}_y = [Y_1, \ldots, Y_n]$
- next: for $i = 1, \ldots, n-1$: $|X_i - X_{i+1}| + |Y_i - Y_{i+1}| = 1$
- saw: for $i = 1, \ldots, n-1$, for $j = i+1, \ldots, n$: $|X_i - X_i| + |Y_i - Y_j| \geq 1$
- We want to express that $(X_i, Y_i) \neq (X_j, Y_j)$. Can we use alldifferent?
- Let $[P_1, \ldots, P_n]$ be a list and M a "big" integer (100 is ok for us).
- for $i = 1, \ldots, n-1$: $P_i = X_i + MY_i$.
- We can now post: alldifferent($[P_1, \ldots, P_n]$).

HP on $\mathbb{N}^2$: FD encoding

- Primary $= [a_1, \ldots, a_n] = [h, p, p, h, p, p, h, ...]$
- Tertiary$_x = [X_1, \ldots, X_n]$, Tertiary$_y = [Y_1, \ldots, Y_n]$

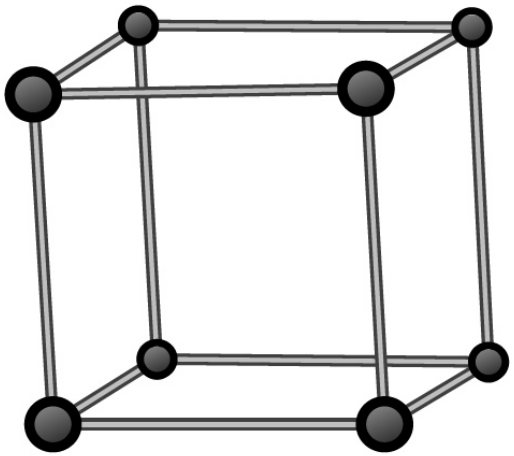
HP on $\mathbb{N}^2$: FD encoding

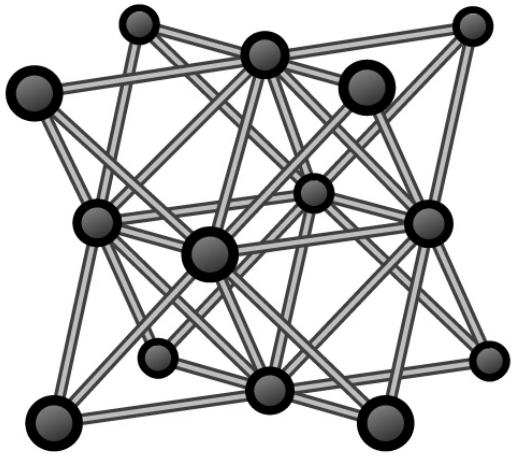
- Primary $= [a_1, \ldots, a_n] = [h, p, p, h, p, p, h, ...]$
- Tertiary$_x = [X_1, \ldots, X_n]$, Tertiary$_y = [Y_1, \ldots, Y_n]$
- energy: for $i = 1, \ldots, n-2$, for $j = i+2, \ldots, n$: $c_{i,j} \in \{0, -1\}$

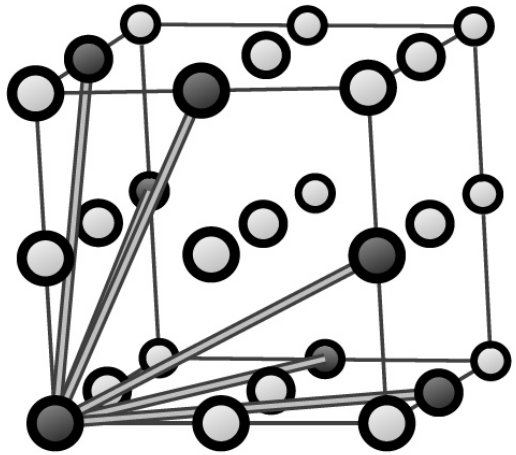
$$c_{i,j} = -1 \leftrightarrow (|X_i - X_i| + |Y_i - Y_j)| = 1) \wedge (a_i = a_j = h)$$

- Energy $= \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} c_{i,j}$
- See the code `pfhp.pl`

Lattice models: Cube, FCC, Chess Knight



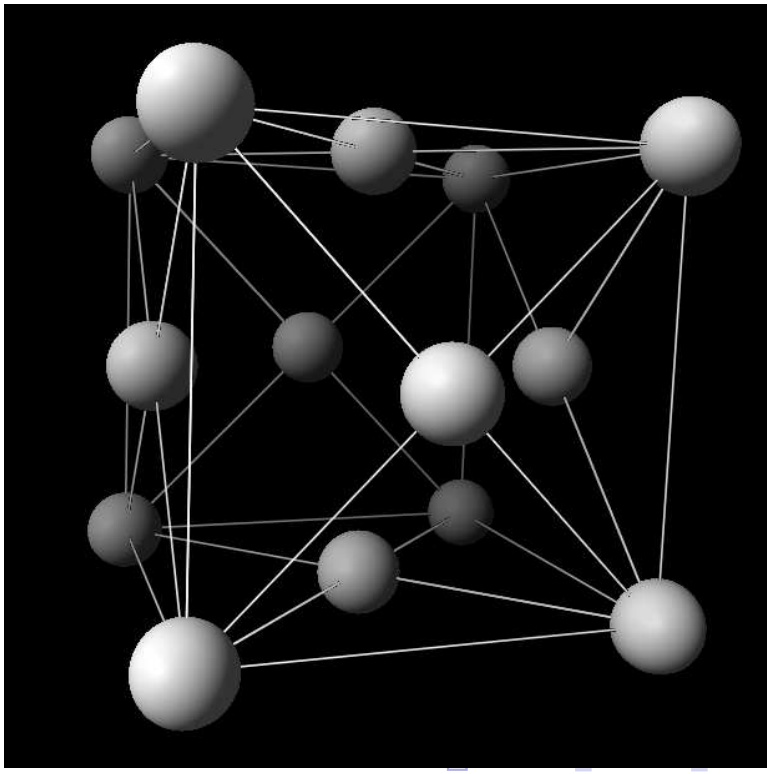




The FCC lattice

- The Face Centered Cube lattice models the discrete space in which the protein can fold.
- It is proved to allow realistic conformations.
- The cube has size 2.

- Two points are *connected* (next) iff $|x_i - x_j|^2 + |y_i - y_j|^2 + |z_i - z_j|^2 = 2$,
- Each point has 12 neighbors (but 60° and 180° can be removed).



The protein folding problem

HP on FCC

- Backofen and Will fold HP-proteins up to length 200 on FCC using constraint programming
- Clever propagation, an idea of stratification and some geometrical results on the lattice.
- Drawbacks: It is only an abstraction. The solutions obtained are far from reality. For instance, helices and sheets are never obtained.
- Problems:
  - Energy function too simple.
  - Contact too strict.

The protein folding problem

A more realistic Energy function

- A 20×20 *potential matrix* Pot storing the contribution for each pair of aminoacids is used.
- Values are either positive or negative.
- The notion of *contact* (easy) on lattice models is slightly extended:
- if distance $(a_i, a_j) < k$ then $\text{Pot}(a_i, a_j)$ else $\frac{\text{Pot}(a_i, a_j)}{\text{distance}^2}$

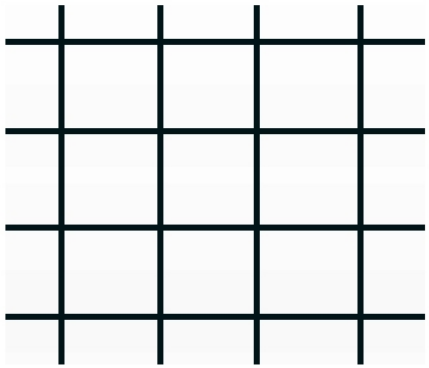
The protein folding problem

A more realistic Energy function

- A 20×20 *potential matrix* Pot storing the contribution for each pair of aminoacids is used.
- Values are either positive or negative.
- The notion of *contact* (easy) on lattice models is slightly extended:
- if distance $(a_i, a_j) < k$ then $\text{Pot}(a_i, a_j)$ else $\frac{\text{Pot}(a_i, a_j)}{\text{distance}^2}$
- COLA (COnstraint solving on LAttices) can predict on FCC proteins of length 100–120 in reasonable time

Global constraints

contiguous



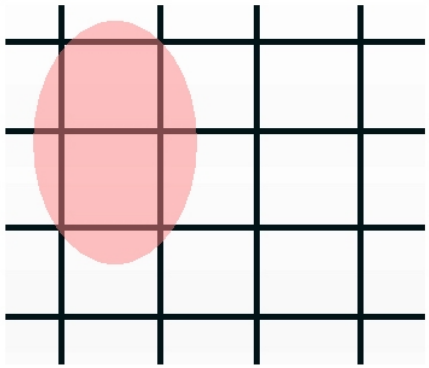
- Let $X_1, \ldots, X_n$ be variables with domains $D_1, \ldots, D_n$:

$$\begin{aligned}\texttt{contiguous}(X_1, \ldots, X_n) = (D_1 \times \cdots \times D_n) \setminus \\ \{(a_1, \ldots, a_n) \in (D_1 \times \cdots \times D_n) : \\ \exists i.\ (1 \leq i < n \wedge (a_i, a_{i+1}) \notin E)\}\end{aligned}$$

  where E is the set of lattice edges.
- CON (consistency chcking) and GAC (generalized arc consistency filtering) are polynomial

Global constraints

contiguous



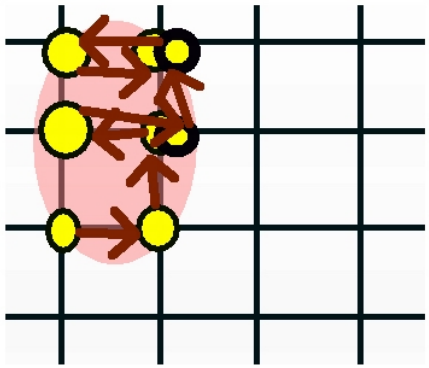
- Let $X_1, \ldots, X_n$ be variables with domains $D_1, \ldots, D_n$:

$$\begin{aligned}\texttt{contiguous}(X_1, \ldots, X_n) = (D_1 \times \cdots \times D_n) \setminus \\ \{(a_1, \ldots, a_n) \in (D_1 \times \cdots \times D_n) : \\ \exists i.\ (1 \leq i < n \wedge (a_i, a_{i+1}) \notin E)\}\end{aligned}$$

  where E is the set of lattice edges.
- CON (consistency chcking) and GAC (generalized arc consistency filtering) are polynomial

Global constraints

contiguous



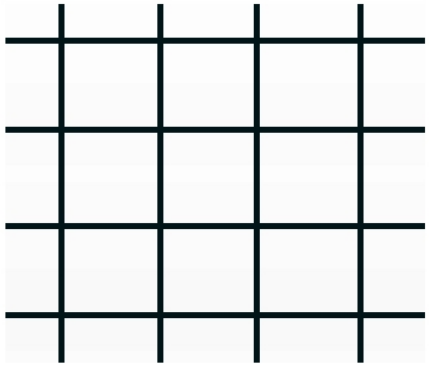
- Let $X_1, \ldots, X_n$ be variables with domains $D_1, \ldots, D_n$:

$$\begin{aligned}\texttt{contiguous}(X_1, \ldots, X_n) = (D_1 \times \cdots \times D_n) \setminus \\ \{(a_1, \ldots, a_n) \in (D_1 \times \cdots \times D_n) : \\ \exists i.\ (1 \leq i < n \wedge (a_i, a_{i+1}) \notin E)\}\end{aligned}$$

  where E is the set of lattice edges.
- CON (consistency chcking) and GAC (generalized arc consistency filtering) are polynomial

Global constraints

alldifferent



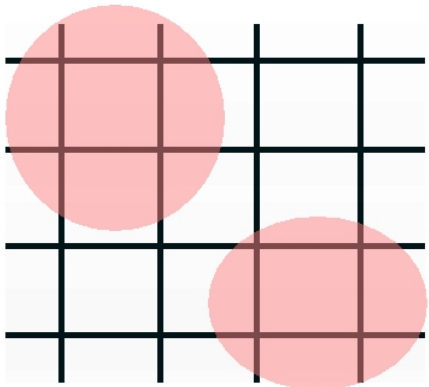
- Let $X_1, \ldots, X_n$ be variables with domains $D_1, \ldots, D_n$:

$$\texttt{alldifferent}(X_1, \ldots, X_n) = (D_1 \times \cdots \times D_n) \setminus \{(a_1, \ldots, a_n) \in (D_1 \times \cdots \times D_n) : \exists i, j.\ (1 \leq i < j \leq n \wedge a_i = a_j)\}$$

- CON and GAC are polynomial

Global constraints

alldifferent



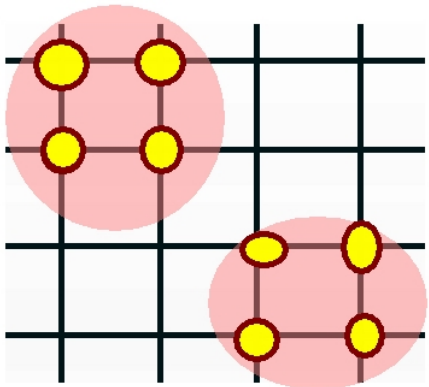
- Let $X_1, \ldots, X_n$ be variables with domains $D_1, \ldots, D_n$:

$$\texttt{alldifferent}(X_1, \ldots, X_n) = (D_1 \times \cdots \times D_n) \setminus \{(a_1, \ldots, a_n) \in (D_1 \times \cdots \times D_n) : \exists i, j.\ (1 \leq i < j \leq n \wedge a_i = a_j)\}$$

- CON and GAC are polynomial

Global constraints

alldifferent



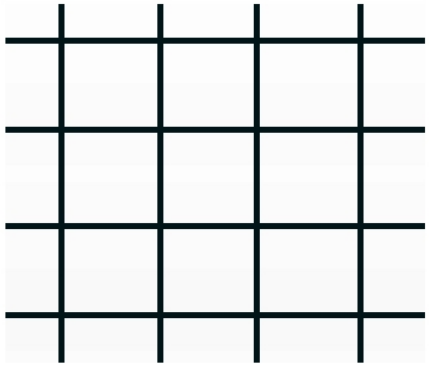
- Let $X_1, \ldots, X_n$ be variables with domains $D_1, \ldots, D_n$:

$$\texttt{alldifferent}(X_1, \ldots, X_n) = (D_1 \times \cdots \times D_n) \setminus \{(a_1, \ldots, a_n) \in (D_1 \times \cdots \times D_n) : \exists i, j.\ (1 \leq i < j \leq n \wedge a_i = a_j)\}$$

- CON and GAC are polynomial

Global constraints

self avoiding walk



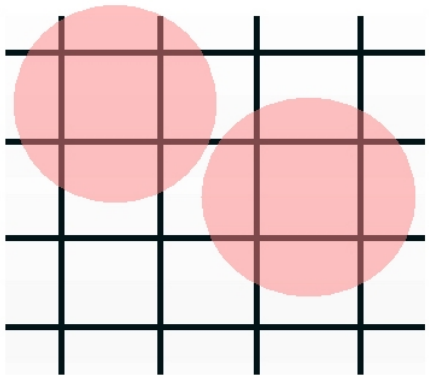
- Given n variables $X_1, \ldots, X_n$, with domains $D_1, \ldots, D_n$, the global constraint `saw` is the following:

$$\texttt{saw}(X_1, \ldots, X_n) = \texttt{alldifferent}(X_1, \ldots, X_n) \cap \texttt{contiguous}(X_1, \ldots, X_n)$$

- CON (and GAC) are NP-complete (Dal Palù, Dovier, Pontelli. IJDMB 4(1), 2010)

Global constraints

self avoiding walk



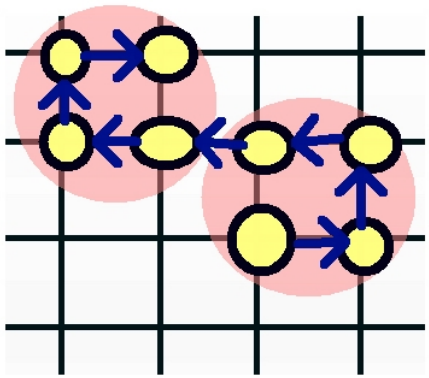
- Given n variables $X_1, \ldots, X_n$, with domains $D_1, \ldots, D_n$, the global constraint `saw` is the following:

$$\begin{aligned}\texttt{saw}(X_1, \ldots, X_n) = \\ \texttt{alldifferent}(X_1, \ldots, X_n) \cap \\ \texttt{contiguous}(X_1, \ldots, X_n)\end{aligned}$$

- CON (and GAC) are NP-complete (Dal Palù, Dovier, Pontelli. IJDMB 4(1), 2010)

Global constraints

self avoiding walk



- Given n variables $X_1, \ldots, X_n$, with domains $D_1, \ldots, D_n$, the global constraint `saw` is the following:

$$\texttt{saw}(X_1, \ldots, X_n) = \texttt{alldifferent}(X_1, \ldots, X_n) \cap \texttt{contiguous}(X_1, \ldots, X_n)$$

- CON (and GAC) are NP-complete (Dal Palù, Dovier, Pontelli. IJDMB 4(1), 2010)

Global constraints

Other constraints

- all distant (NP)
- chain (NP)
- rigid block (P)
- Density maps (NP)
- These constraints are built-in in the tool COLA, in order to predict proteins in the FCC lattice with a 20×20 contact energy function.

Conclusions (of this part)

- Constraint Programming is a viable technique for studying the Protein Structure prediction on Lattice Domains.

Conclusions (of this part)

- Constraint Programming is a viable technique for studying the Protein Structure prediction on Lattice Domains.
- Small number of angles allowed by a lattice models: large errors are unavoidable for long proteins
- Search space is definitely too large for proteins larger than 150 amino acids
- Difficult to reuse known information from deposited proteins (state-of-the-art methods are largely built upon this idea).
- The $C\alpha$-$C\alpha$ model has limits. The same holds for contact energy potentials.

Conclusions (of this part)

- Constraint Programming is a viable technique for studying the Protein Structure prediction on Lattice Domains.
- Small number of angles allowed by a lattice models: large errors are unavoidable for long proteins
- Search space is definitely too large for proteins larger than 150 amino acids
- Difficult to reuse known information from deposited proteins (state-of-the-art methods are largely built upon this idea).
- The $C\alpha$-$C\alpha$ model has limits. The same holds for contact energy potentials.
- We'll see how exiting from lattices while using a discrete reasoner.

Some References

- R. Backofen and S. Will, A constraint-based approach to fast and exact structure prediction in 3-dimensional protein models, Constraints 11(1):5-30, 2006.
- A. Dal Palù, A. Dovier and F. Fogolari. Constraint logic programming approach to protein structure prediction, BMC Bioinformatics 5(186), 2004.
- A. Dal Palù, A. Dovier and E. Pontelli, A constraint solver for discrete lattices, its parallelization, and application to protein structure prediction, Software Practice and Experience 37(13):1405-1449, 2007. (COLA)
- A. Dal Palù, A. Dovier and E. Pontelli. Computing approximate solutions of the protein structure determination problem using global constraints on discrete crystal lattices, Int'l Journal of Data Mining and Bioinformatics 4(1):1–20, 2010. Also in WCB 06 and WCB 07
- P. Barahona and L. Krippahl, Constraint programming in structural bioinformatics, Constraints 13(1-2):3-20, 2008.
- A. Dovier. Recent constraint/logic programming based advances in the solution of the protein folding problem. Intelligenza Artificiale 5(1):113-117, 2011.
- Approximated results with local search and/or LNS by Hoos et al. and by Van Hentenryck et al.