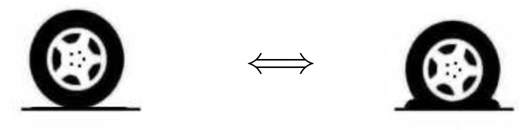
Optimal structure prediction in 3D HP models and applications

Martin Mann

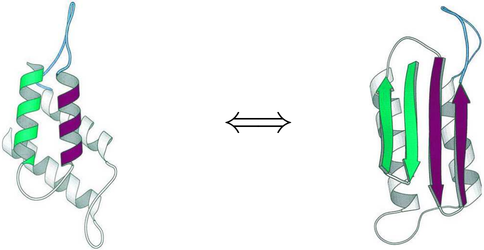
Bioinformatics Group
Albert-Ludwigs-University Freiburg

Why to tackle structure prediction?



$\Longleftrightarrow$

$$\boxed{\textbf{Structure} = \textbf{Function}}$$



$\Longleftrightarrow$

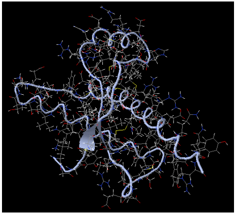
One of the central questions

What is the functional fold?

```
>1HJM:A|PDBID|CHAIN|SEQUENCE
LGGYMLGSAMSRPIIHFGSDYEDRYYRENM
HRYPNQVYYRPMDEYSNQNNFVHDCVNITI
KQHTVTTTTKGENFTETDVKMMERVVEQMC
ITQYERESQAYYQR
```

⇔

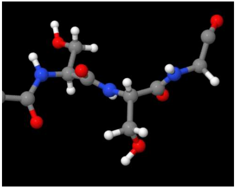
?



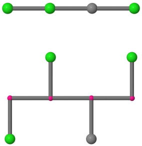
- **What determines the structure?**
- **What drives folding?**
- **What distinguishes protein and random sequences?**

Lattice proteins

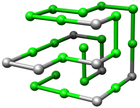
A common abstraction of proteins



monomers



structure space

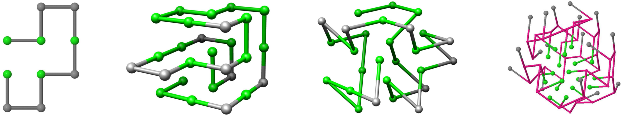


complex interactions

energy function

contact based pairwise potentials

Lattice proteins



Energy function and lattice
determine
level of abstraction and comp. complexity.

central assumption

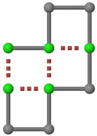
Optimal structure ⇔ Functional fold

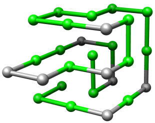
The HP model

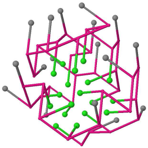
- Introduced by Lau and Dill (1989) in 2D-square
- Simplest lattice model
- Focus on hydrophobic forces
  ⇒ **H**ydrophobic or **P**olar monomers
- Energy = negated sum of HH-contacts
- Structures = Self-Avoiding Walks

⇒ Optimal structure prediction is NP-complete

(Berger&Leighton,1998)

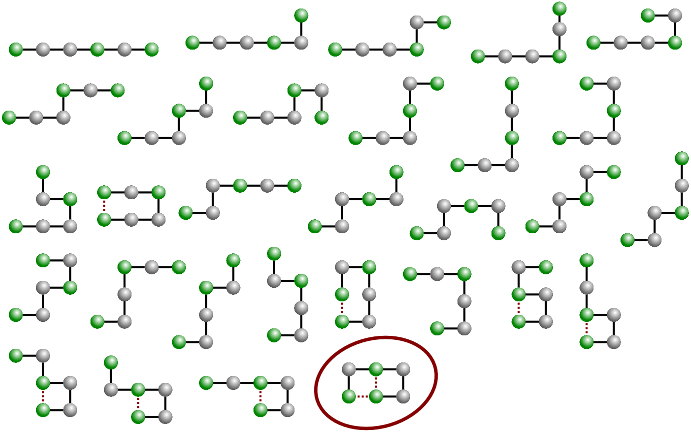






Structures in the HP-Model

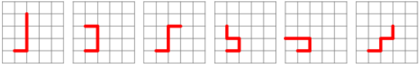
Sequence HPPHPH



Optimal structure prediction

How to do for lattice proteins?

Discrete structure space:



. . .

Exhaustive structure enumeration
⇒ yields optimal structures
⇒ restricted to short lengths

NP-complete problem
⇒ let's try Constraint Programming !

3D-cubic lattice

n	# SAW(n)
0	1
1	6
2	30
3	150
4	726
5	3534
6	16926
7	81390
8	387966
9	1853886
10	8809878
11	41934150
12	198842742
13	943974510
14	4468911678
15	21175146054
16	100121875974
17	473730252102
18	2237723684094
19	10576033219614
20	49917327838734
21	235710090502158
22	1111781983442406
23	5245988215191414
	$> 10^{15}$

Constraint Programming?

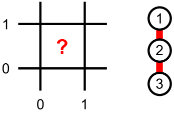
Constraint programming …

- … is a programming technique
- … describes what rather than how
- … i.e. it is declarative and generic
- … combines logic reasoning with search
- … performs “intelligent” enumeration
- … is for slaying NP-hard dragons

Constraint Programming? An Example . . .

SAW enumeration

The problem definition - “What is a solution?”



Constraint Satisfaction Problem (CSP)

- $①,②,③ \in \{\binom{0}{0}, \binom{0}{1}, \binom{1}{0}, \binom{1}{1}\}$
- $\mathit{neigh}(①,②)$, $\mathit{neigh}(②,③)$
- $① \neq ②$, $① \neq ③$, $② \neq ③$

Automatically done by constraint solvers:

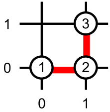
Solution is computed by guessing/search and reasoning

1. guess $① = \binom{0}{0} \rightarrow \{②,③\} \neq \binom{0}{0} \quad \wedge \quad ② \in \{\binom{0}{1}, \binom{1}{0}\}$
2. guess $② = \binom{1}{0} \rightarrow ③ = \binom{1}{1} \rightarrow$ solution found

Constraint Programming? An Example . . .

SAW enumeration

The problem definition - “What is a solution?”



Constraint Satisfaction Problem (CSP)

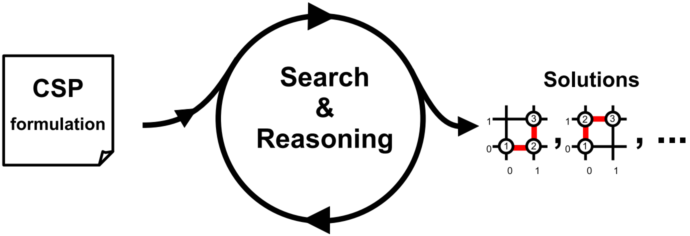
- ①,②,③ $\in \left\{\binom{0}{0}, \binom{0}{1}, \binom{1}{0}, \binom{1}{1}\right\}$
- *neigh*(①,②), *neigh*(②,③)
- ①$\neq$②, ①$\neq$③, ②$\neq$③

Automatically done by constraint solvers:

Solution is computed by guessing/search and reasoning

1. guess ① $= \binom{0}{0} \rightarrow$ {②,③} $\neq \binom{0}{0} \quad \wedge \quad$ ② $\in \left\{\binom{0}{1}, \binom{1}{0}\right\}$
2. guess ② $= \binom{1}{0} \rightarrow$ ③$= \binom{1}{1} \rightarrow$ solution found

Constraint Programming Workflow



CSP formulation

Search & Reasoning

Solutions

Constraint Satisfaction Problem (CSP)

Definition

A Constraint Satisfaction Problem (CSP) consists of

- *variables* $\mathcal{X} = \{X_1, \dots, X_n\}$,
- the domain D that associates finite domains $D_1 = D(X_1), \dots, D_n = D(X_n)$ to $\mathcal{X}$.
- a set of constraints C.

A solution is an assignment of variables to values of their domains that satisfies the constraints.

BUT:

Optimal structure prediction

$\Updownarrow$

Optimization Problem

Constraint Optimization

Definition

A Constraint Optimization Problem (COP) is a CSP together with an objective function f on solutions.
A solution of the COP is a solution of the CSP that maximizes/minimizes f.

Solving by Branch & Bound Search

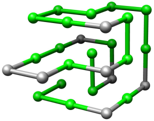
Idea of B&B:

- Search & Reasoning as for solving the CSP
- Whenever a solution s is found, add constraint "next solutions must be better than $f(s)$".

A First Constraint Model

Formulation of the COP

$S \in \{H,P\}^n$ HHPHPHPHP. . . $\Rightarrow$ Optimal Structures



A COP for optimal structure prediction

Variables : one X_i for each S_i

Domains : $D(X_i) = \mathbb{Z}^3$

Constraints : $\forall_{1 \leq i < n} : neigh(X_i, X_{i+1})$ (chain)

$\forall_{i \neq j} : X_i \neq X_j$ (self-avoidance)

Objective : “maximize number of HH-contacts”

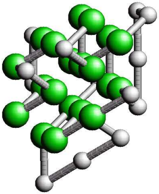
Awful performance due to poor contact bounds

From a partial solution no good estimation of final HH-contacts

Anything else we can use?

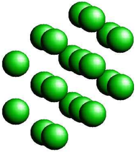
The H-core observation

Optimal structures show (nearly) optimal H-monomer packings



Optimal
Structure

$\Longleftrightarrow$



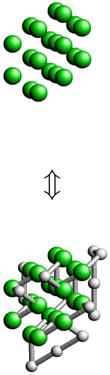
Optimal
H-core

Let's utilize H-cores for a new Constraint Approach!

What is an H-core?

<u>H-cores in HP-models</u>

- Corresponds to hydrophobic core in globular protein structures
- Set of H-monomer lattice positions
- Sequence independent
- Optimal = maximal # HH-contacts

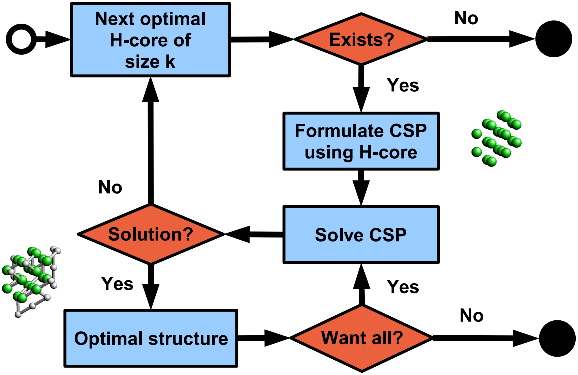


Central observation

If a structure contains an optimal H-core no better structure can be found.

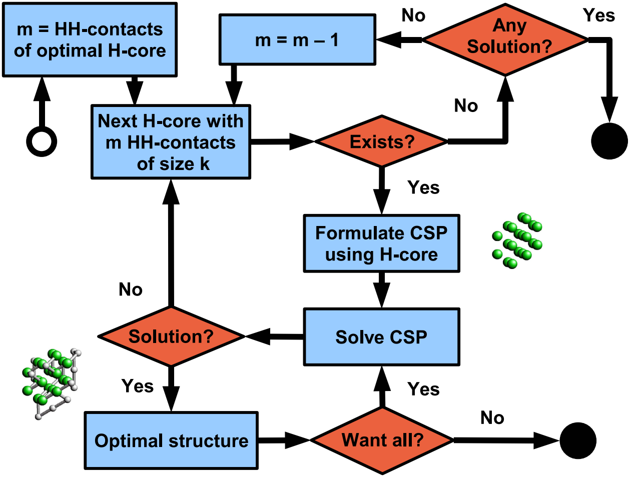
Idea of the new Constraint Approach

Optimal structure prediction solving CSPs



Idea of the new Constraint Approach

Optimal structure prediction solving CSPs



m = HH-contacts of optimal H-core

m = m – 1

Any Solution?

Next H-core with m HH-contacts of size k

Exists?

Formulate CSP using H-core

Solve CSP

Solution?

Optimal structure

Want all?

The CPSP Approach

Constraint-based Protein Structure Prediction solving CSPs

Advantages of the new approach

- Solving CSPs more efficient than for COPs
- Ensures optimality via H-core order
  (decreasing HH-contact number)
- Allows for calculation of all optimal structures
  (even suboptimal)

$\Longrightarrow$ CPSP-approach by Backofen and Will (2006)

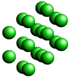
Questions to answer

- What CSP is formulated?
- How to obtain (optimal) H-cores?

The CPSP Approach

The Constraint Satisfaction Problem (simple version)

Input : ① Sequence $S \in \{H, P\}^n$, ② H-core h



The CPSP CSP

Variables : one X_i for each S_i

Domains : $D(X_i) = \begin{cases} h & \text{if } S_i = \text{H} \\ \mathbb{Z}^3 \setminus h & \text{else } (S_i = \text{P}) \end{cases}$

Constraints : $\forall_{1 \leq i < n} : neigh(X_i, X_{i+1})$ (chain)
$\forall_{i \neq j} : X_i \neq X_j$ (self-avoidance)

$\Longrightarrow$ defines all SAWs with H-monomers (X_i) in H-core h

$\Longrightarrow$ if h is optimal → each solution is an optimal structure

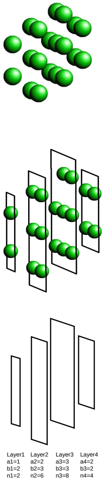
How to obtain (optimal) H-cores

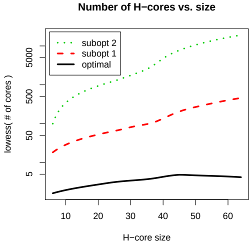
The workhorses of the CPSP approach

- Optimal H-core calculation very hard
- For 3D cubic & FCC solved via DP & CP by (Backofen & Will, 2001)

BUT : Sequence independent !

- only size of importance
- create precalculated H-core DB !





3D cubic

The CPSP Approach

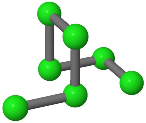
A first summary

- Utilize observation:
  Optimal structures show (nearly) optimal H-cores
- Precalculate (sub)optimal H-core database
  (sequence independent)
- For a given sequence S screen through appropriate H-cores in decreasing # contacts order (Backofen&Will,2006)
  - for each formulate & solve a CSP until solution found
  - order ensures: first solution = optimal
- Combine with symmetry breaking, distance measure, . . .
- Easy to extend (lattice, side chain, . . .)

CPSP-Tools

Tools for high-throughput studies in 3D HP models

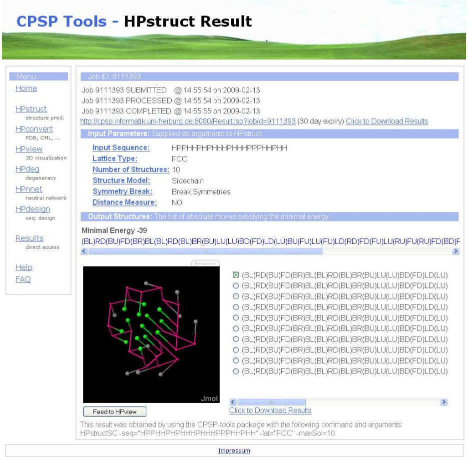
- C++ implementation of CPSP approach and other CPSP based methods (Mann et al., 2008)
- For high-throughput usage
- Extremly fast (length < 100 within 1s)



- Available at
  `www.bioinf.uni-freiburg.de/sw/cpsp`
- NEW: CPSP-web-tools for online usage
  `cpsp.informatik.uni-freiburg.de`

CPSP-web-tools

Online usage of CPSP-tools



CPSP Tools - HPstruct Result

Job ID: 9111393

Job 9111393 SUBMITTED @ 14:55:54 on 2009-02-13
Job 9111393 PROCESSED @ 14:55:54 on 2009-02-13
Job 9111393 COMPLETED @ 14:55:55 on 2009-02-13
http://cpsp.informatik.uni-freiburg.de:8080/Result.jsp?jobid=9111393 (30 day expiry) Click to Download Results

Input Parameters: Supplied as arguments to HPstruct

Input Sequence:	HPPHHPHPHHHPHHHPPPHHPHH
Lattice Type:	FCC
Number of Structures:	10
Structure Model:	Sidechain
Symmetry Break:	Break Symmetries
Distance Measure:	NO

Output Structures: The list of absolute moves satisfying the minimal energy

Minimal Energy -39

(BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)BU(FU)LU(FU)LD(RD)FD(FU)LU(RU)FU(RU)FD(BD)F

● (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)

Click to Download Results

This result was obtained by using the CPSP-tools package with the following command and arguments:

```
HPstructSC -seq="HPPHHPHPHHHPHHHPPPHHPHH" -lat="FCC" -maxSol=10
```

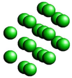
`http://cpsp.informatik.uni-freiburg.de`

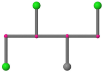
The CPSP Extension to Side Chain Models

Optimal structure prediction

Idea : Only change CSP and keep CPSP approach

Input : ① Sequence $S \in \{H, P\}^n$, ② H-core h

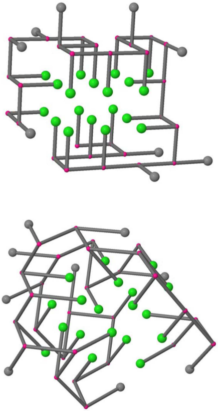


		
Variables	one X_i for each S_i	X_i^b and X_i^s for each S_i
Domains	$D(X_i)\begin{cases} h & \text{if H} \\ \mathbb{Z}^3 \setminus h & \text{if P} \end{cases}$	$D(X_i^s)\begin{cases} h & \text{if H} \\ \mathbb{Z}^3 \setminus h & \text{if P} \end{cases}$ $D(X^b) = \mathbb{Z}^3 \setminus h$
Constraints	$\forall : \mathit{neigh}(X_i, X_{i+1})$ $\forall : X_i \neq X_j$	$\forall : \mathit{neigh}(X_i^b, X_{i+1}^b)$ $\forall : X_i^{b,s} \neq X_j^{b,s}$ $\forall : \mathit{neigh}(X_i^b, X_i^s)$

The CPSP Extension to Side Chain Models

First Results

- Slower but still fast
  (length 60 within seconds)
- Immense # of optimal structures
  (best so far in 3D cubic HP^{28} $= 48$)
  (none so far in 3D fcc HP^{28} $< 10^3$)
- HP energy model for complex lattice protein models insufficient!

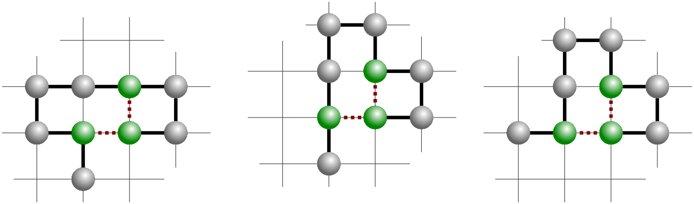


CPSP Applications

- Optimal structure/energy prediction
- Degeneracy calculation
- Definition of proteinlike sequences
- Sequence design & Evolutionary studies
- . . . ?

Degeneracy of the HP model

Sequence HPPHPPPHP



. . . ?

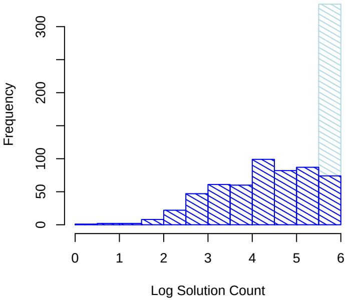
- There can be many ...
- HP-model is degenerated
- Number of optimal structures = degeneracy
- Important for thermodynamic stability

Degeneracy via CPSP approach

The `HPdeg` tool

- CPSP allows for calculation of all optimal structures $\Longrightarrow$ degeneracy
- Even faster via new CP techniques (Will&Mann, 2006)
- CPSP also answers: "Is $deg(S) \leq d_{max}$?"

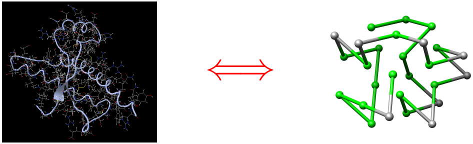
- Degeneracy is immense in 3D cubic HP model
- Even higher in 3D FCC or side chain models



CPSP Applications

- Optimal structure/energy prediction
- Degeneracy calculation
- Definition of proteinlike sequences
- Sequence design & Evolutionary studies
- . . . ?

The Abstraction Problem



The Problem

- No useful mapping of functional protein AA sequences to HP sequences due to abstractions
- Random sequence will not show proteinlike behavior

$\Longrightarrow$ Definition of “proteinlike sequence” needed!

What are Proteinlike Features?

What we want

- Thermodynamically stable structure (low degeneracy)
- Optimal structure = target of folding (= functional)
- Smooth energy landscape → fast folder
- Consider sequential assembly of proteins (low co-translational energy barriers)

Results for 3D cubic HP^{27} (Mann et al.,2008)

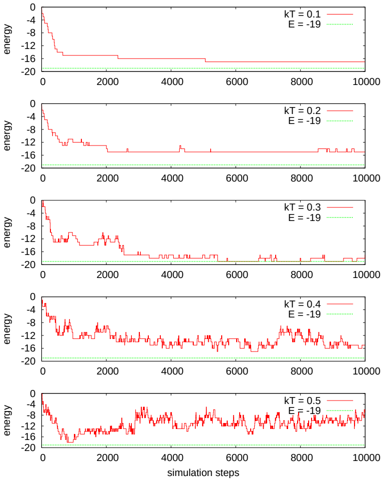
- Only a few sequences are classified as proteinlike
- Used folding temperature kT is of high importance (artificial energy model)

Determining the Folding Temperature kT^f

- Via folding simulations
- Metropolis criterion: $e^{-\Delta G/kT}$ if $\Delta G > 0$

- Screen kT for kT^f
- kT^f is model specific

- Example:
  - 3D cubic HP^{27} (deg=1)
  - Pull-moves (Lesh,2003)
  - $kT^f \approx 0.3$



Filter for Proteinlike Sequences

The Workflow (Mann et al.,2008)

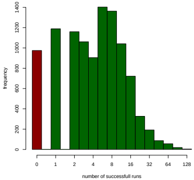
Input : sequence set S_0

① Filter all sequences $S_1 = \{s \mid s \in S_0 \wedge deg(s) \leq deg_{max}\}$
$\Rightarrow$ stable optimal structures

② Determine kT^f for random sample of S_1

$\forall_{s \in S_1}$: run m folding simulations

- calculate successive run ratio
- set threshold for good folder
- $\Rightarrow$ derive $S_2 \subseteq S^1$



③ Filter S_2 via chain-growth folding simulation
$\Rightarrow S_3 = \{s \mid s \in S_2 \wedge E^{ctf}_{barrier} \leq E^{ctf}_{max}\}$ $\Rightarrow$ low barriers

Output : proteinlike sequences S_3

Proteinlike Sequences in HP^{27} 3D cubic

Results:

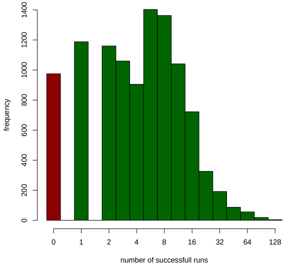
S_0	$\subset HP^{27}$	non-exhaustive
S_1	10,500	degeneracy $= 1$ ($\approx$0.01%)
S_2	2447	good folder
S_3	605	proteinlike

Utilities:

- CPSP-tools (Mann et al.,2008)
  `cpsp.informatik.uni-freiburg.de`
- LatPack-tools (Mann et al.,2008)
  `www.bioinf.uni-freiburg.de/Software/`

Sequence data:

- freely available at
  `www.bioinf.uni-freiburg.de/Data/`



1000 short runs with 4000 steps

good folder: hit rate $> 1\%$ ($\frac{10}{1000}$)

CPSP Applications

- Optimal structure/energy prediction
- Degeneracy calculation
- Definition of proteinlike sequences
- Sequence design & Evolutionary studies
- . . . ?

Thanks to . . .

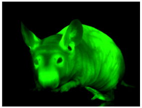
Rolf Backofen
Sebastian Will

Daniel Maticzka (ELL, LatPack)
Cameron Smith (CPSP-web-tools)
Mohamad Rabbath (CPSP side chain)

Rhodri Saunders (Oxford)
Guido Tack (Saarbrücken)

Thanks for your attention !

Contact:
`mmann[@]informatik.uni-freiburg.de`



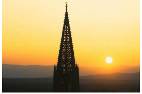
CPSP-tools

`http://cpsp.informatik.uni-freiburg.de`

Energy Landscape Library

`http://www.bioinf.uni-freiburg.de/sw/ell/`

Thanks for attention . . .
. . . and see you in Freiburg !?!



Appendix

Appendix

References and Further Reading

CPSP-tools and approach

- **CPSP-tools - exact and complete algorithms for high-throughput 3D lattice protein studies**
  Martin Mann, Sebastian Will, and Rolf Backofen, *BMC Bioinformatics*, 9:230, 2008.
- **A constraint-based approach to fast and exact structure prediction in 3D protein models**
  Rolf Backofen and Sebastian Will, *Constraints*, 11(1):5–30, 2006.
- **CPSP-web-tools : a server for 3D lattice protein studies**
  Martin Mann, Cameron Smith, Mohamad Rabbath, Marlien Edwards, Sebastian Will, and Rolf Backofen, *Bioinformatics*, 2009.

Proteinlike Sequences

- **Classifying protein-like sequences in arbitrary lattice protein models using LatPack**
  Martin Mann, Daniel Maticzka, Rhodri Saunders, and Rolf Backofen, *HFSP Journal*, 2:396, 2008.
- **The energy landscape library - a platform for generic algorithms**
  Martin Mann, Sebastian Will, and Rolf Backofen, *Proc. of BIRD'07*, 217:83-86, OGC, 2007.

Constraint Programming

- **Decomposition during search for propagation-based constraint solvers**
  Martin Mann, Guido Tack, and Sebastian Will, *arxiv*, 2008.
- **Excluding symmetries in constraint-based search**
  Rolf Backofen and Sebastian Will, *Constraints*, 7:333–349, 2002.