Coupling CP with Deep Learning for Molecular Design

and SARS-CoV2 variants exploration

Thomas Schiex

August 29 2023

CP2023, Toronto, Canada

Thank you!

- For inviting me and for accepting a remote presentation
- I'd love to be with you
- It saved 2 tons of CO_2!

What we will see

- What is a protein, why is it exciting to design new ones?
- What connection with CP?
- How does it enable SARS-CoV2 variants exploration?
- How Deep Learning can learn the rules of protein design (or Sudoku BTW)?

Thank you!

- For inviting me and for accepting a remote presentation
- I'd love to be with you
- It saved 2 tons of CO_2!

What we will see

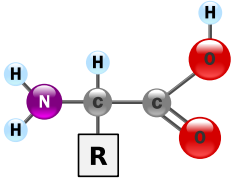
- What is a protein, why is it exciting to design new ones?
- What connection with CP?
- How does it enable SARS-CoV2 variants exploration?
- How Deep Learning can learn the rules of protein design (or Sudoku BTW)?

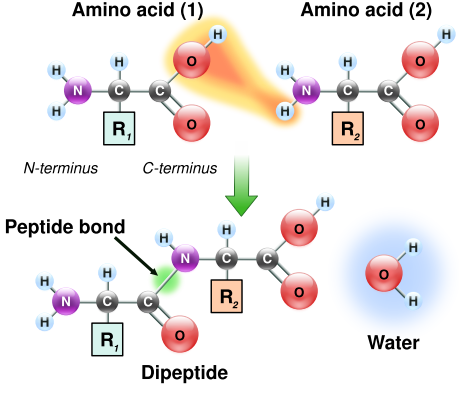
Proteins?

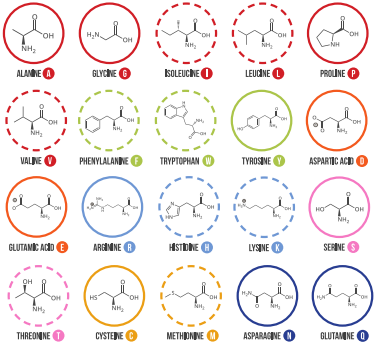
Thanks to wikipedia

Most active molecules of life

Sequence of "amino-acids", each chosen among 20 natural ones



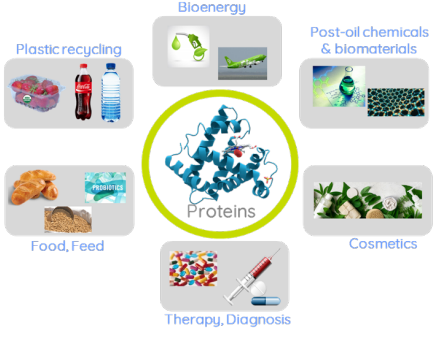




Why should we want to design proteins?

Eco-friendly chemical/structural nano-agents present in all living organisms

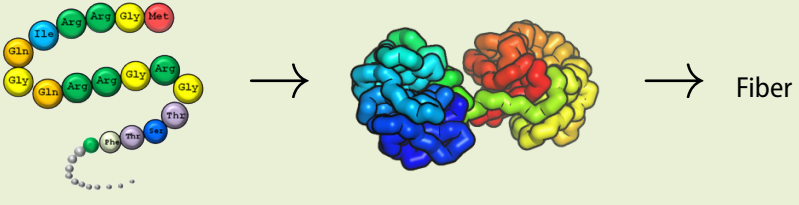
- New drugs for health (human, animals, plants)
- New catalysts (environment, recycling, biofuels, food and feed, cosmetics…),
- Can be synthesized by inexpensive microscopic 3D-printers (bacterias, yeast, …)
- Biodegradable



Globular proteins

- Acquire their properties through their 3D structure
- In solvent, the fold is defined by the protein sequence
- This is what AlphaFold2 predicts

Folding

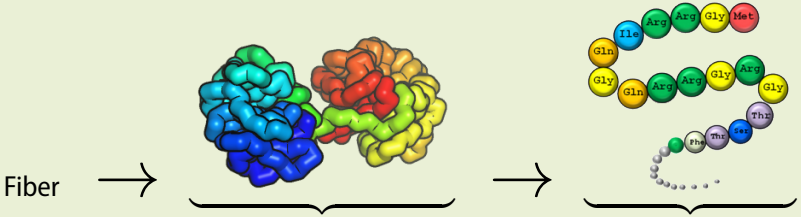


Protein folding and protein design

Globular proteins

- Acquire their properties through their 3D structure
- In solvent, the fold is defined by the protein sequence
- This is what AlphaFold2 predicts

Inverse folding



The Computational Protein Sequence Design Problem

Informal definition (globular proteins)

Produce a sequence s of n amino-acids
that spontaneously adopts a target fold.

Proteins are flexible: simplifications

The "rigid backbone, discrete rotamers" model

1. The backbone structure is fixed (rigid).
2. Sequence s is discrete, the side-chain geometries are discretized.

Rotamer libraries: Tuffery,[19] Penultimate,[8] Dunbrack[15] …

Catalog of (amino acid, side-chain conformations) pairs build from the PDB (typically 400 or more rotamers)

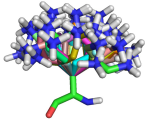
Proteins are flexible: simplifications

The "rigid backbone, discrete rotamers" model

1. The backbone structure is fixed (rigid).
2. Sequence $\mathbf{s}$ is discrete, the side-chain geometries are discretized.

Rotamer libraries: Tuffery,[19] Penultimate,[8] Dunbrack[15] …

Catalog of (amino acid, side-chain conformations) pairs build from the PDB (typically 400 or more rotamers)



Atomic forces and entropic effects

- Current "truth": quantum mechanics but quickly intractable
- Use approximate descriptions of forces (Coulomb, bonds, van der Waals,…)
- Captured inside an "energy function"

Thermodynamics[2]

The probability of sequence s in conformation X is defined by its energy $E(\mathrm{s}, X)$.

$$p(\mathrm{s}, X) \propto e^{-E(\mathrm{s},X)} \qquad p(\mathrm{s}, X) = \frac{e^{-E(\mathrm{s},X)}}{Z}$$

Atomic forces and entropic effects

- Current "truth": quantum mechanics but quickly intractable
- Use approximate descriptions of forces (Coulomb, bonds, van der Waals,…)
- Captured inside an "energy function"

Thermodynamics[2]

The probability of sequence $\mathbf{s}$ in conformation X is defined by its energy $E(\mathbf{s}, X)$.

$$p(\mathbf{s}, X) \propto e^{-E(\mathbf{s},X)} \qquad p(\mathbf{s}, X) = \frac{e^{-E(\mathbf{s},X)}}{Z}$$

More simplifications

Use a "pairwise decomposable energy"

1. The energy function $E(\mathrm{s}, X)$ is pairwise decomposable — Rosetta β-nov16[1]
2. Only an approximation of the real (intractable to compute) energy

Decomposability: precomputed energy tables

i_r: rotamer r at position i

$$E(\mathrm{s}, X) = E_{\varnothing} + \sum_{i=1}^{n} E_i(i_r) + \sum_{(i,j)\in I} E_{ij}(i_r, j_s)$$

- We need to minimize E.
- We optimize the sequence, physics will optimize the geometry in water.

Mostly solved by Monte-Carlo algorithms (Rosetta simulated annealing)[7]

More simplifications

Use a "pairwise decomposable energy"

1. The energy function $E(\mathrm{s}, X)$ is pairwise decomposable — Rosetta β-nov16[1]
2. Only an approximation of the real (intractable to compute) energy

Decomposability: precomputed energy tables

i_r: rotamer r at position i

$$E(\mathrm{s}, X) = E_{\varnothing} + \sum_{i=1}^{n} E_i(i_r) + \sum_{(i,j)\in I} E_{ij}(i_r, j_s)$$

- We need to minimize E.
- We optimize the sequence, physics will optimize the geometry in water.

Mostly solved by Monte-Carlo algorithms (Rosetta simulated annealing)[7]

More simplifications

Use a "pairwise decomposable energy"

1. The energy function $E(\mathbf{s}, X)$ is pairwise decomposable — Rosetta β-nov16[1]
2. Only an approximation of the real (intractable to compute) energy

Decomposability: precomputed energy tables

i_r: rotamer r at position i

$$E(\mathbf{s}, X) = E_{\varnothing} + \sum_{i=1}^{n} E_i(i_r) + \sum_{(i,j)\in I} E_{ij}(i_r, j_s)$$

- We need to minimize E.
- We optimize the sequence, physics will optimize the geometry in water.

Mostly solved by Monte-Carlo algorithms (Rosetta simulated annealing)[7]

More simplifications

Use a "pairwise decomposable energy"

1. The energy function $E(\mathbf{s}, X)$ is pairwise decomposable — Rosetta β-nov16[1]
2. Only an approximation of the real (intractable to compute) energy

Decomposability: precomputed energy tables

i_r: rotamer r at position i

$$E(\mathbf{s}, X) = E_{\varnothing} + \sum_{i=1}^{n} E_i(i_r) + \sum_{(i,j)\in I} E_{ij}(i_r, j_s)$$

- We need to minimize E.
- We optimize the sequence, physics will optimize the geometry in water.

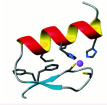
Mostly solved by Monte-Carlo algorithms (Rosetta simulated annealing)[7]

1985

Calmodulin-binding peptide

[DeGrado et al. 1985]

1997



Zinc Finger

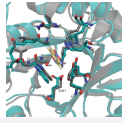
[Dehiyat & Mayo 1997]

2003

Novel Topology (top7)

[Kuhlman et al. 2003]

2008



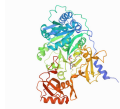
Functional Enzyme

[Rothlisberger et al. 2008]

Enzyme for Multi-Step Reaction

[Jiang et al. 2008]

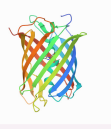
2009



Enzyme activity

[Chen et al. 2009]

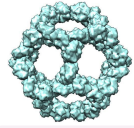
2011



Longer Emission Wave Length Fluorescence

[Chica et al. 2011]

2016



Self-Assembling Nanocage

[Hsia et al. 2016]

2019

Auto-Assembling Symmetrical Protein

[Niguchi et al. 2019]

Cost Function Network (a type of Graphical model)

Cost function network (X, E)

- a sequence X of discrete variables x_i, domain D_i
- a set E of cost functions e_S (possibly infinite costs)
- e_S is a cost function over variables in S expressed as a table
- a solution minimizes the **joint cost function** $E = \sum_{e_S \in E} e_S$ (WCSP, NP-hard)

Graphical models?

- Variables are vertices
- Connected by an edge if they interact (participate together in a function)
- Stochastic graphical models (Markov Random Fields):

$$p(X) \propto e^{-E(\mathbf{X})} \qquad p(X) = \frac{e^{-E(X)}}{Z}$$

Cost Function Network (a type of Graphical model)

Cost function network (X, E)

- a sequence X of discrete variables x_i, domain D_i
- a set E of cost functions e_S (possibly infinite costs)
- e_S is a cost function over variables in S expressed as a table
- a solution minimizes the **joint cost function** $E = \sum_{e_S \in E} e_S$ (WCSP, NP-hard)

Graphical models?

- Variables are vertices
- Connected by an edge if they interact (participate together in a function)
- Stochastic graphical models (Markov Random Fields):

$$p(X) \propto e^{-E(\mathbf{X})} \qquad p(X) = \frac{e^{-E(X)}}{Z}$$

Cost Function Network (a type of Graphical model)

Cost function network (X, E)

- a sequence X of discrete variables x_i, domain D_i
- a set E of cost functions e_S (possibly infinite costs)
- e_S is a cost function over variables in S expressed as a table
- a solution minimizes the **joint cost function** $E = \sum_{e_S \in E} e_S$ (WCSP, NP-hard)

Graphical models?

- Variables are vertices
- Connected by an edge if they interact (participate together in a function)
- Stochastic graphical models (Markov Random Fields):

$$p(X) \propto e^{-E(\mathbf{X})} \qquad p(X) = \frac{e^{-E(X)}}{Z}$$

Cost Function Network (a type of Graphical model)

Cost function network (X, E)

- a sequence X of discrete variables x_i, domain D_i
- a set E of cost functions e_S (possibly infinite costs)
- e_S is a cost function over variables in S expressed as a table
- a solution minimizes the **joint cost function** $E = \sum_{e_S \in E} e_S$ (WCSP, NP-hard)

Graphical models?

- Variables are vertices
- Connected by an edge if they interact (participate together in a function)
- Stochastic graphical models (Markov Random Fields):

$$p(X) \propto e^{-E(\mathbf{X})} \qquad p(X) = \frac{e^{-E(X)}}{Z}$$

Cost Function Network (a type of Graphical model)

Cost function network (X, E)

- a sequence X of discrete variables x_i, domain D_i
- a set E of cost functions e_S (possibly infinite costs)
- e_S is a cost function over variables in S expressed as a table
- a solution minimizes the **joint cost function** $E = \sum_{e_S \in E} e_S$ (WCSP, NP-hard)

Graphical models?

- Variables are vertices
- Connected by an edge if they interact (participate together in a function)
- Stochastic graphical models (Markov Random Fields):

$$p(X) \propto e^{-E(\mathbf{X})} \qquad p(X) = \frac{e^{-E(X)}}{Z}$$

Cost Function Network (a type of Graphical model)

Cost function network (X, E)

- a sequence X of discrete variables x_i, domain D_i
- a set E of cost functions e_S (possibly infinite costs)
- e_S is a cost function over variables in S expressed as a table
- a solution minimizes the **joint cost function** $E = \sum_{e_S \in E} e_S$ (WCSP, NP-hard)

Graphical models?

- Variables are vertices
- Connected by an edge if they interact (participate together in a function)
- Stochastic graphical models (Markov Random Fields):

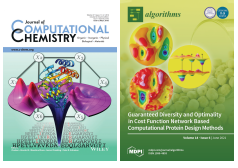
$$p(X) \propto e^{-E(\mathbf{X})} \qquad p(X) = \frac{e^{-E(X)}}{Z}$$

Exact vs. Stochastic search

Large input (> 1GB) **NP-hard problem**

For pratical sizes of problems, toulbar2 is able to…

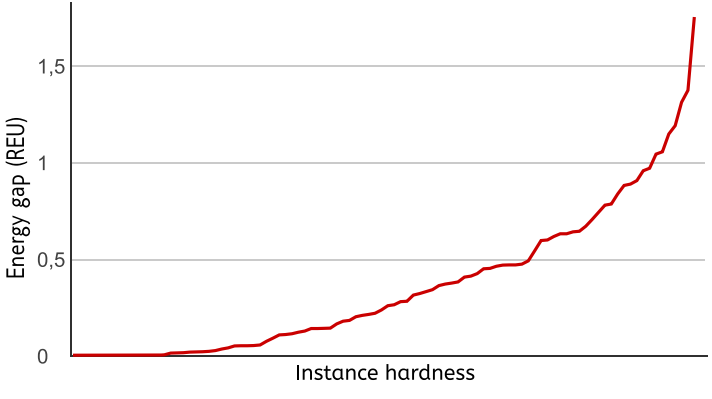
- provide a proven minimum energy solution[17]
- exhaustively enumerate sequences close to it[18]
- provide sequence libraries with guaranteed diversity[14]



Rosetta's Monte Carlo Simulated Annealer increasingly fails to find the optimal sequence[a]

[a] David Simoncini et al. "Guaranteed Discrete Energy Optimization on Large Protein Design Problems". In: *Journal of Chemical Theory and Computation* 11.12 (2015), pp. 5980–5989. DOI: 10.1021/acs.jctc.5b00594.

Unbounded error



Taking the best solution over 1000 runs of Rosetta SA (fixbb)

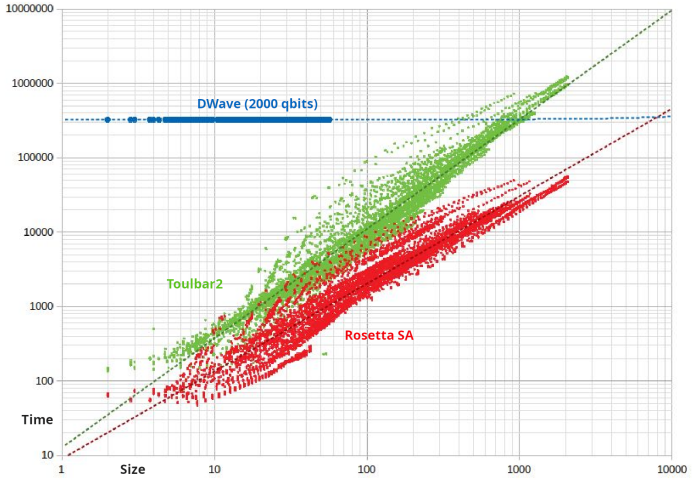
Asymptotic convergence can be arbitrarily slow…

Guaranteed Discrete Energy Optimization on Large Protein Design Problems

David Simoncini[†], David Allouche[†], Simon de Givry[†], Céline Delmas[†], Sophie Barbe[‡§⊥], and Thomas Schiex[*†]

JCTC Journal of Chemical Theory and Computation

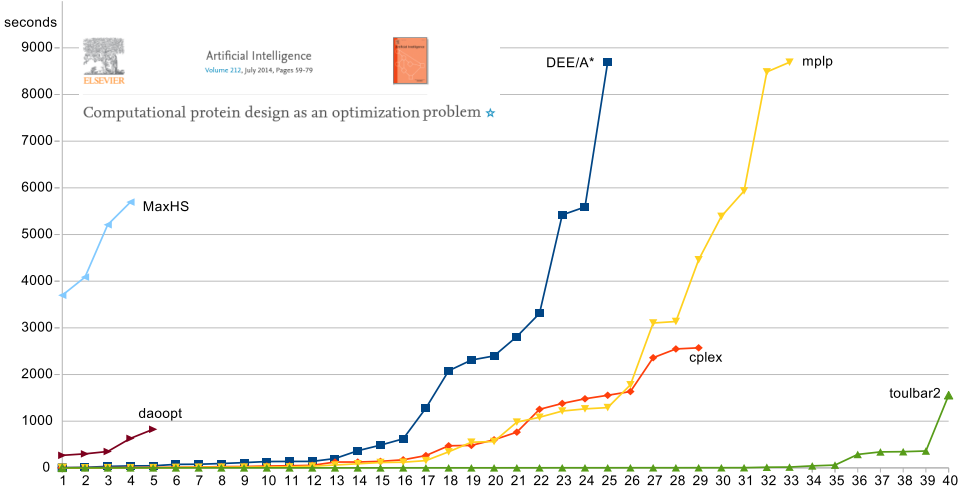
QUBO and Quantum annealing (DWave),Toulbar2 & SA[1]



DWave approximations		*kcal/mol*
gap > 1.16, 90% of the time	> 4.35, 50% of the time	> 8.45, 10% of the time

[1] Vikram Khipple Mulligan et al. "Designing Peptides on a Quantum Computer". In: *bioRxiv* (2019), p. 752485.

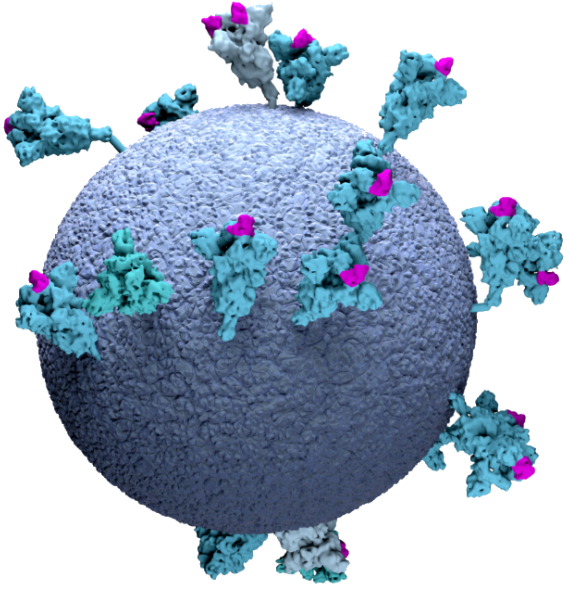
Toulbar2 vs. CPLEX, MaxHS…(real instances)



# of instances solved (*X*) within a per instance cpu-time limit (*Y*)

"The Toulbar2 package for WCSPs significantly improved the state-of-the-art efficiency for protein design." Com. ACM-20, B. Donald et al.

SARS-CoV2, Spike & RBD

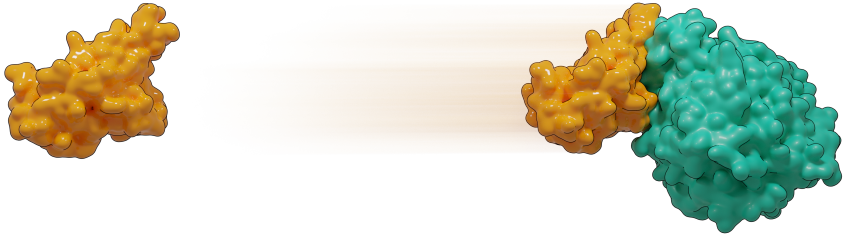


MRC Laboratory of Molecular Biology. Ke, Z., Briggs, J. et al. Nature (2020).

Predicting possible CoViD variants with toulbar2

Crucial step in CoViD infection (Col. C. Bahl - Boston)

- The spike protein (RBD) must bind to the human ACE2 receptor
- March 2020: A structure of the spike RBD bound to ACE2 is published
- Predicting variants would allow for blocking polyclonal vaccines



Stable

Affine

What does this means in terms of energies?

- RBD alone and ACE2 alone $E^{RBD} + E^{ACE2}$
- RBD bound to ACE2 $E^{RBD+ACE2}$
- Thermodynamics says (very simplified) that binding increases with

$$-\Delta E = (E^{RBD} + E^{ACE2}) - E^{RBD+ACE2}$$

Could we try to optimize binding?

- This is a $\Sigma_2^p = NP^{NP}$-hard problem[20]
- Side-chain geometry is free in water. We are playing against Physics.

What does this means in terms of energies?

- RBD alone and ACE2 alone $E^{RBD} + E^{ACE2}$
- RBD bound to ACE2 $E^{RBD+ACE2}$
- Thermodynamics says (very simplified) that binding increases with

$$-\Delta E = (E^{RBD} + E^{ACE2}) - E^{RBD+ACE2}$$

Could we try to optimize binding?

- This is a $\Sigma_2^p = NP^{NP}$-hard problem[20]
- Side-chain geometry is free in water. We are playing against Physics.

We rely on exhaustive enumeration capabilities

1. the ACE2 sequence is fixed
2. We allow only the 27 interface amino acids of RBD to mutate
3. We allow a shell of 25 amino acids around them to change geometry
4. We exhaustively enumerate low $E^{RBD+ACE2}$ sequences[18]

Result: 91 millions sequences at less than 8 $kcal/mol$ of optimum

- Remove those that destabilize the RBD (E^{RBD})
- Geometry is free in water: we need to solve 91 million (NP-hard) problems
- Embarassingly parallel job (cluster)
- 4.5 millions of sequences, with 3,272 local optima
- Bioinformatics: 59 clusters each with a centroid sequence

We rely on exhaustive enumeration capabilities

1. the ACE2 sequence is fixed
2. We allow only the 27 interface amino acids of RBD to mutate
3. We allow a shell of 25 amino acids around them to change geometry
4. We exhaustively enumerate low $E^{RBD+ACE2}$ sequences[18]

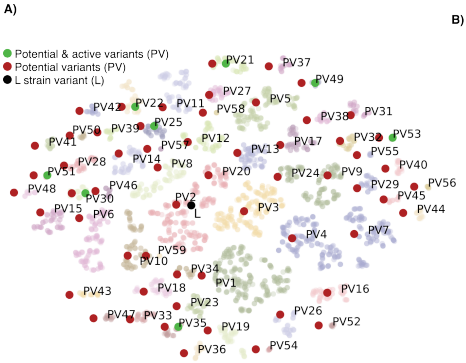
Result: 91 millions sequences at less than $8\ kcal/mol$ of optimum

- Remove those that destabilize the RBD (E^{RBD})
- Geometry is free in water: we need to solve 91 million (NP-hard) problems
- Embarassingly parallel job (cluster)
- 4.5 millions of sequences, with 3,272 local optima
- Bioinformatics: 59 clusters each with a centroid sequence

Yeast display

Yeast Display

- 11/59 variants bind to ACE2
- Select 8 best, 7 purified properly
- Affinity measured by BLI (55nM, ≈WT)



Pseudo viruses vs. HEK293 human cells

Measures

Infectivity and resistance to antibodies

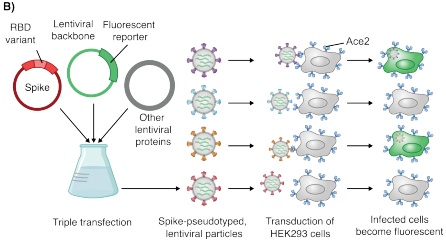
A)

KDs of the indicated soluble RBDs to Fc-Ace2 and therapeutic IgGs

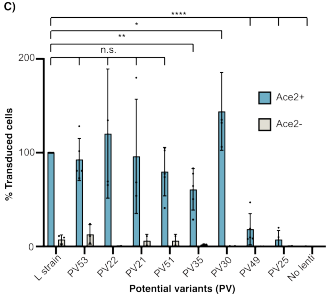
	Fc-Ace2	IgG LY-CoV016	IgG Regn10933	IgG Regn10987
L strain	41.7 ± 7.4 nM	203 ± 63.5 nM	14.4 ± 5.8 nM	74.2 ± 10.8 nM
PV21	155 ± 10.5 nM	n.d.	n.d.	216 ± 29 nM
PV22	118 ± 14.2 nM	n.d.	n.d.	n.d.
PV25	n.d.	n.d.	n.d.	n.d.
PV30	55.6 ± 7.3 nM	n.d.	n.d.	n.d.
PV49	440 ± 59 nM	n.d.	n.d.	n.d.
PV51	291 ± 40 nM	n.d.	4850 nM	n.d.
PV53	222 ± 49 nM	n.d.	n.d.	152 ± 53 nM

n.d.: binding not detected

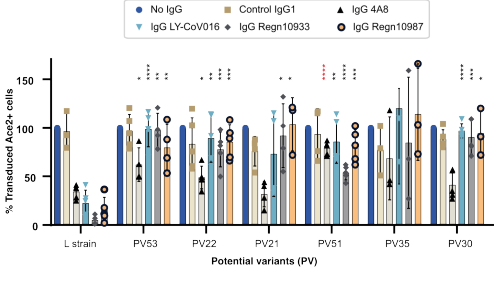
B)



C)



D)



Deep Learning a CFN generator from examples

Why and how (M. Defresne, PhD)

- Learn a (better) energy function from the structure and sequence of known proteins (PDB)
- Start by learning how to play Sudoku
  - $\rightarrow$ We know the answer
  - $\rightarrow$The position of cells influences the constraints acting on them

Existing differentiable DL Sudoku learners

Approach	Architecture	
RRN*	GNN-based	(NeurIPS'17)[11]
SATNet	Weighted MaxSAT SDP Relaxation	(ICML'19)[21]

Deep Learning a CFN generator from examples

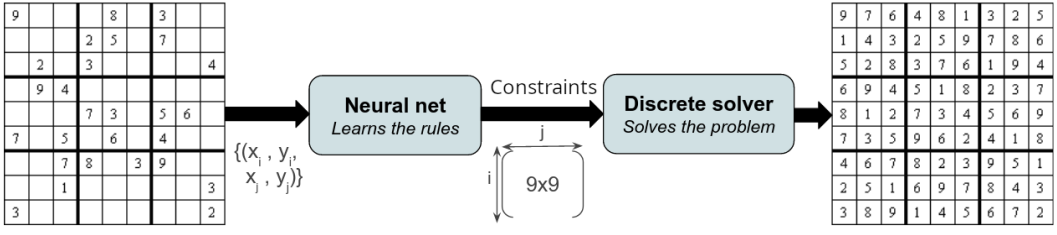
Why and how (M. Defresne, PhD)

- Learn a (better) energy function from the structure and sequence of known proteins (PDB)
- Start by learning how to play Sudoku
  - $\rightarrow$ We know the answer
  - $\rightarrow$The position of cells influences the constraints acting on them

Existing differentiable DL Sudoku learners

Approach	Architecture	
RRN*	GNN-based	(NeurIPS'17)[11]
SATNet	Weighted MaxSAT SDP Relaxation	(ICML'19)[21]

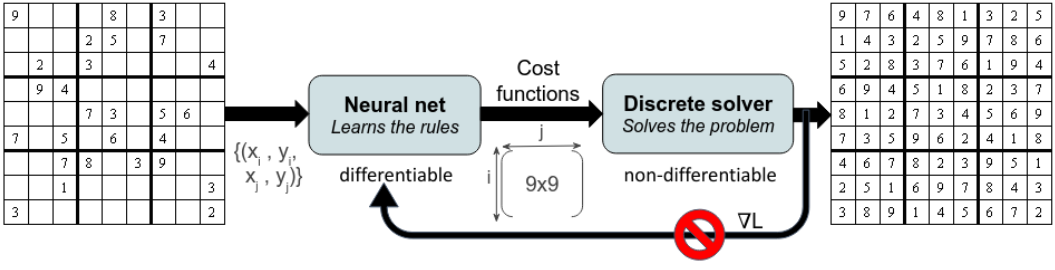
Discrete objects and Gradients!



Two different problems

- Discrete $\{0, \infty\}$ costs, how could we differentiate wrt them?
  $\rightarrow$ We relax the CP problem to Weighted CP (pairwise CFN)
- Discrete variables: loss gradient (Hamming distanc to solution) is zero or indefinite
  $\rightarrow$ We use the probabilistic interpretation of CFN to define the Loss
  $\rightarrow$ Maximize the probability of observed solutions (log-likelihood)

Discrete objects and Gradients!



Two different problems

- Discrete $\{0, \infty\}$ costs, how could we differentiate wrt them?
  $\rightarrow$ We relax the CP problem to Weighted CP (pairwise CFN)
- Discrete variables: loss gradient (Hamming distanc to solution) is zero or indefinite
  $\rightarrow$ We use the probabilistic interpretation of CFN to define the Loss
  $\rightarrow$ Maximize the probability of observed solutions (log-likelihood)

Log-likelihood and Pseudo-loglikelihood[2]

Loglikelihood: a nice constrastive but intractable loss

- log-likelihood of the i.i.d. training set $\mathbf{T}$:

$$\sum_{\mathbf{s}\in\mathbf{T}} \log(p(X=\mathbf{s}))$$

- $p(X=\mathbf{s}) = \frac{e^{-E(\mathbf{s})}}{Z}$ #P-hard

$$\underbrace{\sum_{\mathbf{s}\in\mathbf{T}} -E(s)}_{\text{training set cost}} \quad \underbrace{-\log(\sum_{\mathbf{x}} e^{-E(\mathbf{x})})}_{\text{SoftMin of all assigment costs}}$$

The PLL considers the value of X_i given all other variables values

$$PLL = \sum_{\mathbf{s}\in\mathbf{T}} \sum_{i} \log(p(X_i = s_i | s_{-i}))$$

Tractable and asymptotically consistent estimation

[2] Julian Besag. "Statistical analysis of non-lattice data". In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 24.3 (1975), pp. 179–195.

Log-likelihood and Pseudo-loglikelihood[2]

Loglikelihood: a nice constrastive but intractable loss

- log-likelihood of the i.i.d. training set $\mathbf{T}$:

$$\sum_{\mathbf{s}\in\mathbf{T}} \log(p(X=\mathbf{s}))$$

- $p(X=\mathbf{s}) = \frac{e^{-E(\mathbf{s})}}{Z}$ #P-hard

$$\underbrace{\sum_{\mathbf{s}\in\mathbf{T}} -E(s)}_{\text{training set cost}} \quad \underbrace{-\log(\sum_{\mathbf{x}} e^{-E(\mathbf{x})})}_{\text{SoftMin of all assigment costs}}$$

The PLL considers the value of X_i given all other variables values

$$PLL = \sum_{\mathbf{s}\in\mathbf{T}} \sum_{i} \log(p(X_i = s_i | s_{-i}))$$

Tractable and asymptotically consistent estimation

[2] Julian Besag. "Statistical analysis of non-lattice data". In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 24.3 (1975), pp. 179–195.

How well does it work?

Complete failure, accuracy 0%!

It learns only a small subset of all constraints (row difference constraints)

Contraints and logical consequence

- As soon as the row constraints are learned, $p(X_i|X_{-i})$ is close to one
- Vanishing gradients

Introducing the emmental PLL (dropout-like)

$$EPLL = \sum_{s \in \mathbf{T}} \sum_{i} \log(p(s_i|\text{a random subset of } s_{-i}))$$

[3] Marianne Defresne, Sophie Barbe, and Thomas Schiex. "Scalable Coupling of Deep Learning with Logical Reasoning". In: *Thirty-second International Joint Conference on Artificial Intelligence, IJCAI'2023*. 2023.

Contraints and logical consequence

- As soon as the row constraints are learned, $p(X_i|X_{-i})$ is close to one
- Vanishing gradients

Introducing the emmental PLL (dropout-like)

$$EPLL = \sum_{\mathbf{s}\in\mathbf{T}} \sum_{i} \log(p(s_i|\text{a random subset of } s_{-i}))$$

[3] Marianne Defresne, Sophie Barbe, and Thomas Schiex. "Scalable Coupling of Deep Learning with Logical Reasoning". In: *Thirty-second International Joint Conference on Artificial Intelligence, IJCAI'2023*. 2023.

Performances compared to other differentiable layers

Approach	Characteristic	Acc.	Grids	Trainset	Train time
RRN*	Pure DL	96.6%	Hard	180,000	Hours
SATNet	SDP Relaxation	99.8%	Easy	9,000	Hours
EPLL	Prob. loss	**100%**	Hard	200	15 min.

EPLL properties

- Solver out of the training loop
- Learns all redundant constraints
- Deals with many-solutions problems[10]
- End-to-end differentiable

Loss: last layer

Performances compared to other differentiable layers

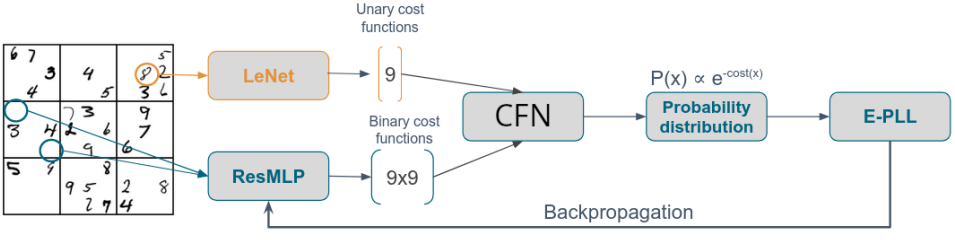
Approach	Characteristic	Acc.	Grids	Trainset	Train time
RRN*	Pure DL	96.6%	Hard	180,000	Hours
SATNet	SDP Relaxation	99.8%	Easy	9,000	Hours
EPLL	Prob. loss	**100%**	Hard	200	15 min.

EPLL properties

- Solver out of the training loop
- Learns all redundant constraints
- Deals with many-solutions problems[10]
- End-to-end differentiable

Loss: last layer

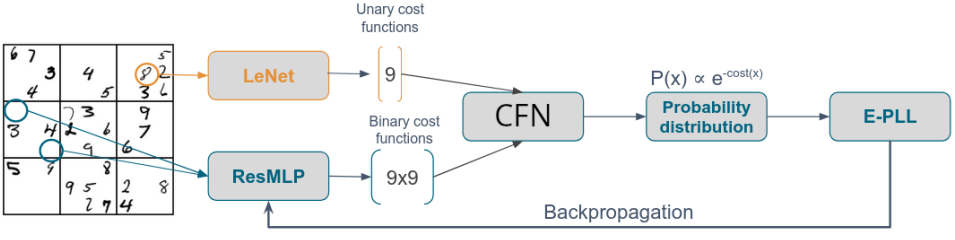
Visual Sudoku: learn to play and to decipher digit images



Using SATNet train and test sets

SATNet	Theoretical (no corrections)	Ours
63.2 %	74.2%	$\mathbf{94.1} \pm 0.8\%$

Visual Sudoku: learn to play and to decipher digit images

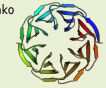


Using SATNet train and test sets

SATNet	Theoretical (no corrections)	Ours
63.2 %	74.2%	$\mathbf{94.1} \pm 0.8\%$

Learning the laws of protein design

- Main changes:
  - Train set up to 10,000 variables (variable size)
  - Conditioned by the input structure (interatomic distances,…)
- Intractable inference → approximate CFN solver (ICML'22)[6]



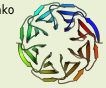
Outperforms SOTA decomposable score functions

	Rosetta[1]	Our
Similarity (↑)	17.9%	**27.8%**

[1] Hahnbeom Park et al. "Simultaneous Optimization of Biomolecular Energy Functions on Features from Small Molecules and Macromolecules". In: *Journal of Chemical Theory and Computation* 12.12 (2016), pp. 6201–6212

Learning the laws of protein design

- Main changes:
  - Train set up to 10,000 variables (variable size)
  - Conditioned by the input structure (interatomic distances,…)
- Intractable inference $\rightarrow$ approximate CFN solver (ICML'22)[6]



Outperforms SOTA decomposable score functions

	Rosetta[1]	Our
Similarity (↑)	17.9%	**27.8%**

[1] Hahnbeom Park et al. "Simultaneous Optimization of Biomolecular Energy Functions on Features from Small Molecules and Macromolecules". In: *Journal of Chemical Theory and Computation* 12.12 (2016), pp. 6201–6212

Pure DL-based sequence design[4]

Using an autoregressive GNN (ProteinMPNN)

- Learns $P(X_i | \text{structure, partial assignment})$ — arbitrary order
- Input: protein structure + a (potentially fully) masked sequence
- Output: a distribution over amino acid types for a chosen position i
- Repeated calls allow to produce a full solution
- Reliably samples high quality solutions — beyond pairwise
- Output cannot be arbitrarily constrained nor easily enumerated

[4] Justas Dauparas et al. "Robust deep learning–based protein sequence design using ProteinMPNN". In: *Science* 378.6615 (2022), pp. 49–56.

Pure DL-based sequence design[4]

Using an autoregressive GNN (ProteinMPNN)

- Learns $P(X_i|\text{structure, partial assignment})$ — arbitrary order
- Input: protein structure + a (potentially fully) masked sequence
- Output: a distribution over amino acid types for a chosen position i
- Repeated calls allow to produce a full solution
- Reliably samples high quality solutions — beyond pairwise
- Output cannot be arbitrarily constrained nor easily enumerated

[4] Justas Dauparas et al. "Robust deep learning–based protein sequence design using ProteinMPNN". In: *Science* 378.6615 (2022), pp. 49–56.

Pure DL-based sequence design[4]

Using an autoregressive GNN (ProteinMPNN)

- Learns $P(X_i | \text{structure, partial assignment})$ — arbitrary order
- Input: protein structure + a (potentially fully) masked sequence
- Output: a distribution over amino acid types for a chosen position i
- Repeated calls allow to produce a full solution
- Reliably samples high quality solutions — beyond pairwise
- Output cannot be arbitrarily constrained nor easily enumerated

[4] Justas Dauparas et al. "Robust deep learning–based protein sequence design using ProteinMPNN". In: *Science* 378.6615 (2022), pp. 49–56.

Pure DL-based sequence design[4]

Using an autoregressive GNN (ProteinMPNN)

- Learns $P(X_i|\text{structure, partial assignment})$ — arbitrary order
- Input: protein structure + a (potentially fully) masked sequence
- Output: a distribution over amino acid types for a chosen position i
- Repeated calls allow to produce a full solution
- Reliably samples high quality solutions — beyond pairwise
- Output cannot be arbitrarily constrained nor easily enumerated

[4] Justas Dauparas et al. "Robust deep learning–based protein sequence design using ProteinMPNN". In: *Science* 378.6615 (2022), pp. 49–56.

Pure DL-based sequence design[4]

Using an autoregressive GNN (ProteinMPNN)

- Learns $P(X_i | \text{structure, partial assignment})$ — arbitrary order
- Input: protein structure + a (potentially fully) masked sequence
- Output: a distribution over amino acid types for a chosen position i
- Repeated calls allow to produce a full solution
- Reliably samples high quality solutions — beyond pairwise
- Output cannot be arbitrarily constrained nor easily enumerated

[4] Justas Dauparas et al. "Robust deep learning–based protein sequence design using ProteinMPNN". In: *Science* 378.6615 (2022), pp. 49–56.

Conclusion

- Computational Protein Design is an exciting application domain for discrete optimization
- It combines knowledge, data and user preferences/constraints on discrete objects
- With ML and DL, CFNs can integrate all these information types together
- Pure autoregressive GNN-based DL approaches very competitive
- In a well defined domain, with correlated information & many examples
  $\rightarrow$ pure DL-based heuristic optimization works
- Post-hoc criteria/constraints language is limited (unary)
- No enumeration, only sampling

Conclusion

- Computational Protein Design is an exciting application domain for discrete optimization
- It combines knowledge, data and user preferences/constraints on discrete objects
- With ML and DL, CFNs can integrate all these information types together
- Pure autoregressive GNN-based DL approaches very competitive
- In a well defined domain, with correlated information & many examples
  $\rightarrow$ pure DL-based heuristic optimization works
- Post-hoc criteria/constraints language is limited (unary)
- No enumeration, only sampling

Conclusion

- Computational Protein Design is an exciting application domain for discrete optimization
- It combines knowledge, data and user preferences/constraints on discrete objects
- With ML and DL, CFNs can integrate all these information types together
- Pure autoregressive GNN-based DL approaches very competitive
- In a well defined domain, with correlated information & many examples
  $\rightarrow$ pure DL-based heuristic optimization works
- Post-hoc criteria/constraints language is limited (unary)
- No enumeration, only sampling

Conclusion

- Computational Protein Design is an exciting application domain for discrete optimization
- It combines knowledge, data and user preferences/constraints on discrete objects
- With ML and DL, CFNs can integrate all these information types together
- Pure autoregressive GNN-based DL approaches very competitive
- In a well defined domain, with correlated information & many examples
  $\rightarrow$ pure DL-based heuristic optimization works
- Post-hoc criteria/constraints language is limited (unary)
- No enumeration, only sampling

Conclusion

- Computational Protein Design is an exciting application domain for discrete optimization
- It combines knowledge, data and user preferences/constraints on discrete objects
- With ML and DL, CFNs can integrate all these information types together
- Pure autoregressive GNN-based DL approaches very competitive
- In a well defined domain, with correlated information & many examples
  $\rightarrow$ pure DL-based heuristic optimization works
- Post-hoc criteria/constraints language is limited (unary)
- No enumeration, only sampling

Conclusion

- Computational Protein Design is an exciting application domain for discrete optimization
- It combines knowledge, data and user preferences/constraints on discrete objects
- With ML and DL, CFNs can integrate all these information types together
- Pure autoregressive GNN-based DL approaches very competitive
- In a well defined domain, with correlated information & many examples
  $\rightarrow$ pure DL-based heuristic optimization works
- Post-hoc criteria/constraints language is limited (unary)
- No enumeration, only sampling

Conclusion

- Computational Protein Design is an exciting application domain for discrete optimization
- It combines knowledge, data and user preferences/constraints on discrete objects
- With ML and DL, CFNs can integrate all these information types together
- Pure autoregressive GNN-based DL approaches very competitive
- In a well defined domain, with correlated information & many examples
  $\rightarrow$ pure DL-based heuristic optimization works
- Post-hoc criteria/constraints language is limited (unary)
- No enumeration, only sampling

Thanks

AI/toulbar2

S. de Givry (INRA)
G. Katsirelos (INRA)
M. Zytnicki (PhD, INRA)
D. Allouche (INRA)
M. Ruffini (INRA)
V. Durante (ANITI, PhD student) H. Nguyen (PhD, INRA)
C. Brouard (ML, INRA)
M. Cooper (IRIT, Toulouse)
J. Larrosa (UPC, Spain)
F. Heras (UPC, Spain)
M. Sanchez (Spain)
E. Rollon (UPC, Spain)
P. Meseguer (CSIC, Spain)
G. Verfaillie (ONERA, ret.)
JH. Lee (CU. Hong Kong)
C. Bessiere (LIMM, Montpellier)
JP. Métivier (GREYC, Caen)
S. Loudni (GREYC, Caen)
M. Fontaine (GREYC, Caen),…

Protein Design

A. Voet (KU Leuven)
A. Olichon (INSERM)
D. Simoncini (UFT, Toulouse)
S. Barbe (INSA, Toulouse)
M. Defresne (INRAE, PhD student)
Y. Bouchiba (INSA, PhD student)
C. Dumont (INSA, Toulouse)
J. Vucinic (INRA/INSA)
S. Traoré (PhD, CEA)
C. Viricel (PhD)
K. Zhang (Riken, CBDR)
S. Tagami (Riken, CBDR)
RosettaCommons (U. Washington)
W. Sheffler (U. Washington)
V. Mulligan (Flatiron Institute, NY)
C. Bahl (IPI, Boston)
PyRosetta (U. John Hopkins)
B. Donald (U. North Carolina)
K. Roberts (U. North Carolina)
T. Simonson (Polytechnique)
J. Cortes (LAAS/CNRS),…

My apologies to those missing in these lists. Even imperfect lists seem better than no list

References I

[1] Rebecca F Alford et al. "The Rosetta all-atom energy function for macromolecular modeling and design". In: *Journal of chemical theory and computation* 13.6 (2017), pp. 3031–3048.

[2] C. Anfinsen. "Principles that govern the folding of protein chains". In: *Science* 181.4096 (1973), pp. 223–253.

[3] Julian Besag. "Statistical analysis of non-lattice data". In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 24.3 (1975), pp. 179–195.

[4] Justas Dauparas et al. "Robust deep learning–based protein sequence design using ProteinMPNN". In: *Science* 378.6615 (2022), pp. 49–56.

[5] Marianne Defresne, Sophie Barbe, and Thomas Schiex. "Scalable Coupling of Deep Learning with Logical Reasoning". In: *Thirty-second International Joint Conference on Artificial Intelligence, IJCAI'2023*. 2023.

[6] Valentin Durante, George Katsirelos, and Thomas Schiex. "Efficient low rank convex bounds for pairwise discrete Graphical Models". In: *Thirty-ninth International Conference on Machine Learning*. July 2022.

[7] Andrew Leaver-Fay et al. "ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules.". In: *Methods Enzymol.* 487 (2011), pp. 545–574. ISSN: 1557-7988.

[8] S C Lovell et al. "The penultimate rotamer library.". In: *Proteins* 40.3 (Aug. 2000), pp. 389–408. ISSN: 0887-3585. URL: `http://www.ncbi.nlm.nih.gov/pubmed/10861930`.

[9] Vikram Khipple Mulligan et al. "Designing Peptides on a Quantum Computer". In: *bioRxiv* (2019), p. 752485.

[10] Yatin Nandwani et al. "Neural Learning of One-of-Many Solutions for Combinatorial Problems in Structured Output Spaces". In: *International Conference on Learning Representations, ICLR'21*. 2021. URL: `https://openreview.net/forum?id=ATp1nW2FuZL`.

References II

[11] Rasmus Palm, Ulrich Paquet, and Ole Winther. "Recurrent Relational Networks". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[12] Hahnbeom Park et al. "Simultaneous Optimization of Biomolecular Energy Functions on Features from Small Molecules and Macromolecules". In: *Journal of Chemical Theory and Computation* 12.12 (2016), pp. 6201–6212.

[13] N. Pierce et al. "Conformational splitting: A more powerful criterion for dead-end elimination". In: *Journal of computational chemistry* 21.11 (2000), pp. 999–1009.

[14] Manon Ruffini et al. "Guaranteed Diversity and Optimality in Cost Function Network Based Computational Protein Design Methods". In: *Algorithms* 14.6 (2021), p. 168.

[15] Maxim V Shapovalov and Roland L Dunbrack. "A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions". In: *Structure* 19.6 (2011), pp. 844–858.

[16] David Simoncini et al. "Guaranteed Discrete Energy Optimization on Large Protein Design Problems". In: *Journal of Chemical Theory and Computation* 11.12 (2015), pp. 5980–5989. DOI: 10.1021/acs.jctc.5b00594.

[17] Seydou Traoré et al. "A New Framework for Computational Protein Design through Cost Function Network Optimization". In: *Bioinformatics* 29.17 (2013), pp. 2129–2136.

[18] Seydou Traoré et al. "Fast search algorithms for computational protein design". In: *Journal of Computational Chemistry* 37.12 (2016), pp. 1048–1058. ISSN: 1096-987X. DOI: 10.1002/jcc.24290. URL: http://dx.doi.org/10.1002/jcc.24290.

[19] P Tuffery et al. "A new approach to the rapid determination of protein side chain conformations". In: *Journal of Biomolecular structure and dynamics* 8.6 (1991), pp. 1267–1289.

[20] Jelena Vucinic et al. "Positive multistate protein design". In: *Bioinformatics* 36.1 (2020), pp. 122–130.

References III

[21] Po-Wei Wang et al. "SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver". In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 6545–6554.