How Computers Break (Serious) Puzzles with logic and (a different breed of) learning

Thomas Schiex

How Computers Break (Serious) Puzzles

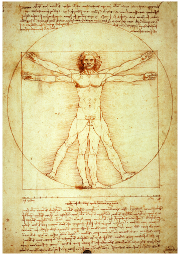
with logic and (a different breed of) learning

Thomas Schiex

February 2018

Académie des sciences, Paris, France

Superhuman performances of AI

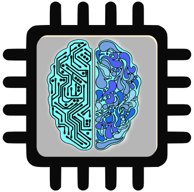


Human beings

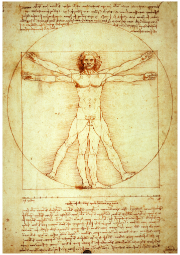
- Easily rely on quick "intuitions" (ill-defined problems)
- Extreme rigor is painful and slow (logic/arithmetic)

AIs (computers)

- Accessible to some "intuition" (problems defined by data)
- Fast and extreme rigor is the default (1 billion op./sec)



Superhuman performances of AI

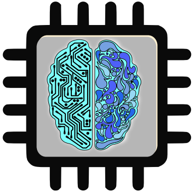


Human beings

- Easily rely on quick "intuitions" (ill-defined problems)
- Extreme rigor is painful and slow (logic/arithmetic)

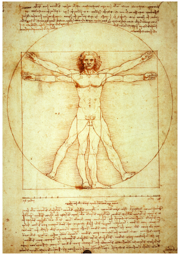
AIs (computers)

- Accessible to some "intuition" (problems defined by data)
- Fast and extreme rigor is the default (1 billion op./sec)



It was expected that machines would show superhuman "logical reasoning" performances

Superhuman performances of AI

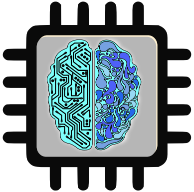


Human beings

- Easily rely on quick "intuitions" (ill-defined problems)
- Extreme rigor is painful and slow (logic/arithmetic)

AIs (computers)

- Accessible to some "intuition" (problems defined by data)
- Fast and extreme rigor is the default (1 billion op./sec)



It was expected that machines would show superhuman "logical reasoning" performances

1955: Newell & Simon "Logic Theorist" proved 38 of the 52 theorems in the *Principia Mathematica* (Russel and Whitehead), and even corrected a proof in it.

Then came complexity theory

NP-hard problems (Cook-Levin, 1970s)

- Some problems seems intrisically hard (for AIs at least)
- *Worst case asymptotic* exponential time (P $\neq$ NP)

Then came complexity theory

NP-hard problems (Cook-Levin, 1970s)

- Some problems seems intrisically hard (for AIs at least)
- *Worst case asymptotic* exponential time (P $\neq$ NP)

$n^2 \times n^2$ Sudoku

- NP-complete, 9×9: 10^{80} cases
- 10^{51} ages of the universe to examine them all
- Fast brute force will fail

							1	
					2			3
			4					
						5		
4		1	6					
		7	1					
	5					2		
				8			4	
	3		9	1				

Then came complexity theory

NP-hard problems (Cook-Levin, 1970s)

- Some problems seems intrisically hard (for AIs at least)
- *Worst case asymptotic* exponential time (P $\neq$ NP)

$n^2 \times n^2$ Sudoku

- NP-complete, 9×9: 10^{80} cases
- 10^{51} ages of the universe to examine them all
- Fast brute force will fail

	E		1	6					D		7	F		G	B
B				7				A		1		9	D		4
A			2	C			1	B						E	
		7					G	F	2						3
	B			1										7	
	9	A			6			1	E				G	B	
		8	C				5		7					1	6
G					2	A	F			3		5			C
E		5	G				B				9				
	C			4			8	5	6						
		9	F	3	1	C		D		E					
	8	4		5		F	D	3		G	1			6	
		C			7	6					3		F		G
5			D	F			C			8			9		
	F	2					3			4					
		6								D			C		5

Then came complexity theory

NP-hard problems (Cook-Levin, 1970s)

- Some problems seems intrisically hard (for AIs at least)
- *Worst case asymptotic* exponential time (P $\neq$ NP)

$n^2 \times n^2$ Sudoku

- NP-complete, 9×9: 10^{80} cases
- 10^{51} ages of the universe to examine them all
- Fast brute force will fail

B		A	P			L				C			M				1		J	H	4	7		
			G	F			A		O		7	9	3	L	C			P	M				B	K
N	M			I					4	P			6	A	E	F	G				C	D	3	
5		K										4		D	L		O		6					
C	H	1			D	F	G	N		B	E												9	P
7	A									6	4					G	H			2	1	F		
G			J	N	L	E	8				B		C	7	1		P		F			H	O	
2				6	K					J	1	P		G	N			A		B	3	M		
	F	I		8	6	C	7		3	D	O	N		H				M		K	E	P		
		3			P						I	2	F		K		J				9	N		
L				K	C	9	N	E	6	H		A	B	M	3		8					2	7	J
	N		3		B		M						D			P			K					O
		5		B	I	O			K	9				3	A			J	2	6		L		
P					F			L			6						4		G		H		K	
1	8	G					2		5	4	L	J		K	I	6	C	B	D	E				9
		F	6				4		7		9	H	G						8			C		
		C	L	2		1				N		F	K	O	7		D	G	3	I		9	E	
		P	A	4		H			L	E		1	J	2					C	O				8
	9	D			8		C		E	L	M		P				F	I	B	1	K			7
		B	8	M			D	6					7	C									G	H
3	G												H	6		O	M	C	E			I	P	1
					1		L		P	3		B										K		F
	C	8	2				5	O	J	G	A			F	9					3			4	B
D	6				M	3			G	8	2	5	9		H		I			A	O			
		H	I	O	7		9				J			E				5			L	6		M

Then came complexity theory

NP-hard problems (Cook-Levin, 1970s)

- Some problems seems intrisically hard (for AIs at least)
- *Worst case asymptotic* exponential time (P $\neq$ NP)

$n^2 \times n^2$ Sudoku

- NP-complete, 9×9: 10^{80} cases
- 10^{51} ages of the universe to examine them all
- Fast brute force will fail
- Can be solved in milliseconds

							1	
					2			3
			4					
						5		
4		1	6					
		7	1					
	5					2		
				8			4	
	3		9	1				

Logic — From a well defined problem to a solution

- We have a set of variables (Sudoku cells contain a number from 1 to 9)

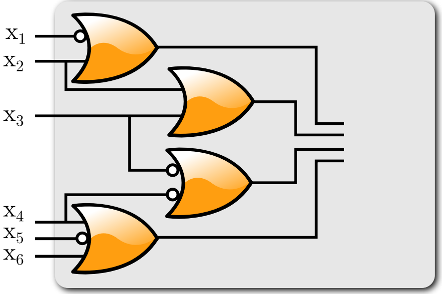
x_1
x_2

x_3

x_4
x_5
x_6

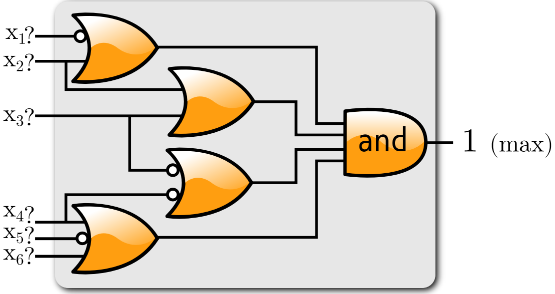
Logic — From a well defined problem to a solution

- We have a set of variables (Sudoku cells contain a number from 1 to 9)
- We have a set of properties on these variables (all different rows, columns, super-cells)



Logic — From a well defined problem to a solution

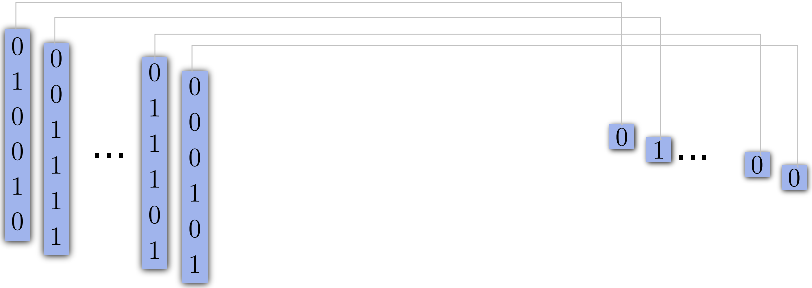
- We have a set of variables (Sudoku cells contain a number from 1 to 9)
- We have a set of properties on these variables (all different rows, columns, super-cells)
- We want to find an input that satisfies all properties (or prove none exists: refutation).



Intuition

Intuition (DL) — From examples to a classifier

- We have a set of digital inputs (in $\mathbb{B}^n$) and output (class: one bit).
- We want a function that best predicts seen (and unseen) data in most cases.



Intuition (DL) — From examples to a classifier

- We have a set of digital inputs (in $\mathbb{B}^n$) and output (class: one bit).
- We want a function that best predicts seen (and unseen) data in most cases.

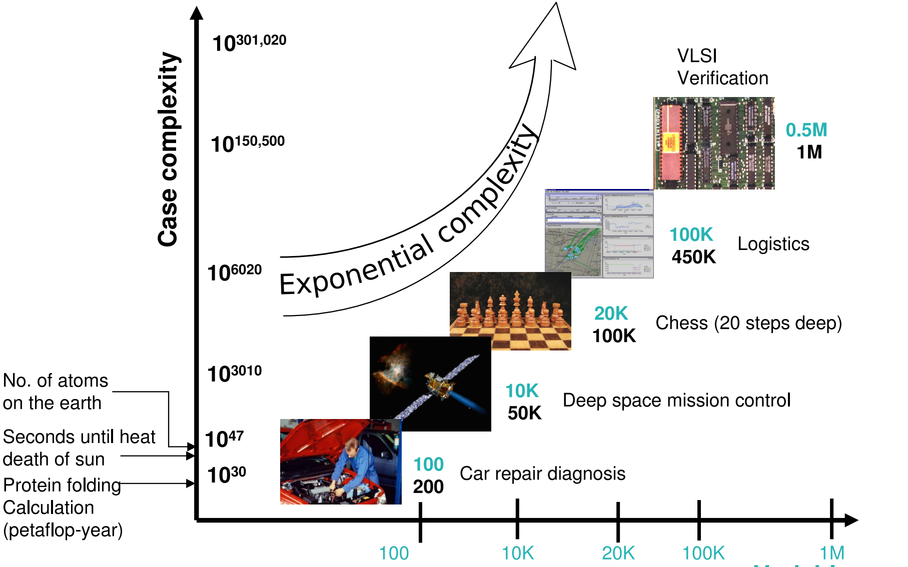
The modern world needs rigourous logic

Technological progress

- Increasingly complex useful objects planes, computers, software, cars, AIs
- That must be highly reliable (lives at stake)
- We cannot fully get them under control anymore

Increasing system complexity

- Hardware: Pentium FDIV bug (1994, 3.1 million transistors)
- Software: the Therac-25 (radiation-therapy) kills 6 patients
- Tesla cars: said to carry 100 millions lines of codes
- Convolutional NN: may have billions of parameters



[Credit: Kumar, DARPA, cited in Computer World magazine]

Note: *rough estimates, for propositional reasoning*

Propositional logic as an example (SAT)

SAT

1. A set of Boolean variables x_i
2. A set of clauses (*disjunction* of variables or negation of) $(\neg x_1 \lor x_7)$
3. Must satisfy all clauses (or prove impossible)
4. Semantics: defines a function from $\mathbb{B}^n$ to $\mathbb{B}$

Propositional logic as an example (SAT)

SAT

1. A set of Boolean variables x_i
2. A set of clauses (*disjunction* of variables or negation of) $(\neg x_1 \vee x_7)$
3. Must satisfy all clauses (or prove impossible)
4. Semantics: defines a function from $\mathbb{B}^n$ to $\mathbb{B}$

Sudoku

1. cell (i, j) contains k — x_{ijk} true
2. At least one number per cell i, j — $(x_{ij1} \vee \ldots \vee x_{ij9})$
3. At most one number per cell i, j — $(\forall k > k' \neg x_{ijk} \vee \neg x_{ijk'})$
4. Cell (i, j) and (i, j') must be different — $(\neg x_{ijk} \vee \neg x_{ij'k})$

SAT is the simplest

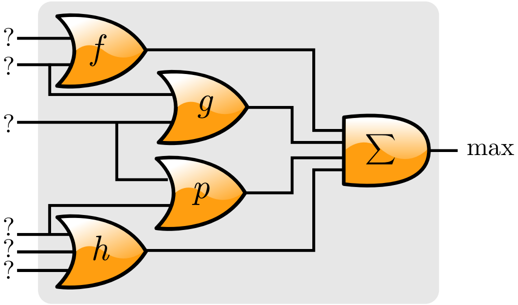
More sophisticated/practical function description

- propositions over theories — SAT Modulo Theory[9]
- non Boolean variables — Constraint Satisfaction, Constraint Programming[30]
- numerical output — Weighted MaxSAT[25]/CSP,[5] Graphical models[18]

SAT is the simplest

More sophisticated/practical function description

- propositions over theories — SAT Modulo Theory[9]
- non Boolean variables — Constraint Satisfaction, Constraint Programming[30]
- numerical output — Weighted MaxSAT[25]/CSP,[5] Graphical models[18]



What can we do with SAT, CP et al.?

NP-complete: can express all NP-complete problems

- the logical puzzles you like (Sudoku, Nonograms...)
- or not (configuration, scheduling, test pattern generation...)
- robot planning
- digital circuit verification (Bounded Model Checking)
- or software verification (FOL, grounding, abstraction)

What can we do with SAT, CP et al.?

NP-complete: can express all NP-complete problems

- the logical puzzles you like (Sudoku, Nonograms...)
- or not (configuration, scheduling, test pattern generation...)
- robot planning
- digital circuit verification (Bounded Model Checking)
- or software verification (FOL, grounding, abstraction)

NP-complete, so intractable

Standard argument for less realistic problem reformulation, heuristics or stochastic search

What can we do with SAT, CP et al.?

NP-complete: can express all NP-complete problems

- the logical puzzles you like (Sudoku, Nonograms...)
- or not (configuration, scheduling, test pattern generation...) SIEMENS THALES
- robot planning (Rosetta-Philae probe plan, CP, LAAS/Toulouse) cnes
- digital circuit verification (Bounded Model Checking) intel IBM
- or software verification (FOL, grounding, abstraction) cadence

NP-complete, so intractable

Standard argument for less realistic problem reformulation, heuristics or stochastic search

Real SAT instances with millions of variables/clauses can be solved (with a proof)

IBM Bounded Model Checking SAT instance (SATLIB)

```
p cnf 51639 368352
-1 7 0
-1 6 0
-1 5 0
-1 -4 0
-1 3 0
-1 2 0
-1 -8 0
-9 15 0
-9 14 0
-9 13 0
-9 -12 0
-9 11 0
-9 10 0
-9 -16 0
```

$51,639$ variables, $368,352$ constraints

$\neg x_1 \vee x_7$

$\neg x_1 \vee x_6$

$\ldots$

10 Pages later

```
185 -9 0
185 -1 0
177 169 161 153 145 137 129
121 113 105 97 89 81 73 65 57
49 41 33 25 17 9 1 -185 0
186 -187 0
186 -188 0
...
```

$$(x_{177} \vee x_{169} \vee x_{161} \vee x_{153} \vee \cdots \vee x_{17} \vee x_{9} \vee x_{1} \vee \neg x_{185})$$

4,000 Pages later

```
10236 -10050 0
10236 -10051 0
10236 -10235 0
10008 10009 10010 10011 10012 10013 10014 10015 10016 10017 10018
10019 10020 10021 10022 10023 10024 10025 10026 10027 10028 10029
10030 10031 10032 10033 10034 10035 10036 10037 10086 10087 10088
10089 10090 10091 10092 10093 10094 10095 10096 10097 10098 10099
10100 10101 10102 10103 10104 10105 10106 10107 10108 -55 -54 53 -52
-51 50 10047 10048 10049 10050 10051 10235 -10236 0
10237 -10008 0
10237 -10009 0
10237 -10010 0
...
```

Finally 15,000 Pages later

```
-7 260 0
7 -260 0
1072 1070 0
-15 -14 -13 -12 -11 -10 0
-15 -14 -13 -12 -11 10 0
-15 -14 -13 -12 11 -10 0
-15 -14 -13 -12 11 10 0
-7 -6 -5 -4 -3 -2 0
-7 -6 -5 -4 -3 2 0
-7 -6 -5 -4 3 -2 0
-7 -6 -5 -4 3 2 0
185 0
```

Finally 15,000 Pages later

```
-7 260 0
7 -260 0
1072 1070 0
-15 -14 -13 -12 -11 -10 0
-15 -14 -13 -12 -11 10 0
-15 -14 -13 -12 11 -10 0
-15 -14 -13 -12 11 10 0
-7 -6 -5 -4 -3 -2 0
-7 -6 -5 -4 -3 2 0
-7 -6 -5 -4 3 -2 0
-7 -6 -5 -4 3 2 0
185 0
```

Search space

$2^{50,000} \approx 3.1\ 10^{15,051}$

Finally 15,000 Pages later

```
-7 260 0
7 -260 0
1072 1070 0
-15 -14 -13 -12 -11 -10 0
-15 -14 -13 -12 -11 10 0
-15 -14 -13 -12 11 -10 0
-15 -14 -13 -12 11 10 0
-7 -6 -5 -4 -3 -2 0
-7 -6 -5 -4 -3 2 0
-7 -6 -5 -4 3 -2 0
-7 -6 -5 -4 3 2 0
185 0
```

Search space

$2^{50,000} \approx 3.1\ 10^{15,051}$

Solved in one second

Finally 15,000 Pages later

```
-7 260 0
7 -260 0
1072 1070 0
-15 -14 -13 -12 -11 -10 0
-15 -14 -13 -12 -11 10 0
-15 -14 -13 -12 11 -10 0
-15 -14 -13 -12 11 10 0
-7 -6 -5 -4 -3 -2 0
-7 -6 -5 -4 -3 2 0
-7 -6 -5 -4 3 -2 0
-7 -6 -5 -4 3 2 0
185 0
```

Search space

$2^{50,000} \approx 3.1\ 10^{15,051}$

Solved in one second

How does it work?

Consensus/Resolution (1960's)[8,29]

$$\frac{(x_1 \vee \overbrace{l_1 \vee \cdots \vee l_n}^{L}) \quad (\neg x_1 \vee \overbrace{r_1 \vee \cdots \vee r_m}^{R})}{(L \vee R)}$$

Boolean Constraint Propagation (BCP): unit clauses[7] (*L* or *R* empty)

$$\frac{(x_1) \quad (\neg x_1 \vee \overbrace{r_1 \vee \cdots \vee r_m}^{R})}{(R)}$$

Boolean Constraint Propagation (BCP): unit clauses[7] (L or R empty)

$$\frac{(x_1) \quad (\neg x_1 \vee \overbrace{r_1 \vee \cdots \vee r_m}^{R})}{(R)}$$

- A clause is shortened by one litteral

Boolean Constraint Propagation (BCP): unit clauses[7] (L or R empty)

$$\frac{(x_1) \quad (\neg x_1 \vee \overbrace{r_1 \vee \cdots \vee r_m}^{R})}{(R)}$$

- A clause is shortened by one litteral
- This may create new unit clauses (propagation)

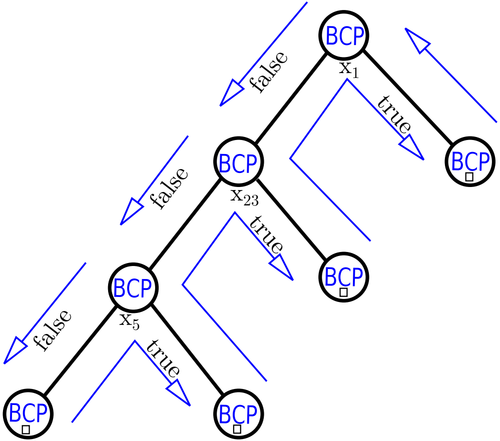
Logic: Do systematic simple logical deductions

Boolean Constraint Propagation (BCP): unit clauses[7] (*L* or *R* empty)

$$\frac{(x_1) \quad (\neg x_1 \vee \overbrace{r_1 \vee \cdots \vee r_m}^{R})}{(R)}$$

- A clause is shortened by one litteral
- This may create new unit clauses (propagation)
- If the empty clause $\square$ appears: no solution

Logic: Try to guess and reconsider (DPLL[7])



SAT state-of-the-art in 1990

Hundreds of variables

Thousands of clauses

Logic: Learn from failure

Long line of research in "symbolic" Artificial Intelligence[3,10,23,24,32]

- Trace back failure to guesses through propagation[a]
- Do backward resolution from conflict
- Add a new implied clause to the set of clauses

[a] Richard M Stallman and Gerald J Sussman. "Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis". In: *Artificial intelligence* 9.2 (1977), pp. 135–196.

- Forces to reconsider an earlier guess
- Prevents refailing for a related reason (safe generalization)

Learns a more effective formulation of the problem as it solves it

Intuition: Choose a variable and try to guess its value

Learning by "Activity based heuristics"[26]

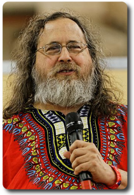
- On-line estimation of how often a variable is involved in recent clauses/failures
- Try guessing this variable first

Learns weak spots in the problem as it is solved (safe)

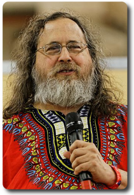
Human intuition based on...

A lot of free data and free code...

- International competitions ($> 50,000$ benchmarks with many real problems)
- Open source solvers (autocatalytic)



Human intuition based on...



A lot of free data and free code...

- International competitions ($> 50,000$ benchmarks with many real problems)
- Open source solvers (autocatalytic)

Strong French presence

- Award winning solvers (Glucose,[2] toulbar2[15])
- Constraint programming solver/startup (Choco)
- Strong presence in international conferences (# of accepted papers in CP[4])

2017: proving an “alien” theorem?

A conjecture in combinatorics ∞

When one splits $\mathbb{N}$ in 2, one part must contain a Pythagorean triple $(a^2 = b^2 + c^2)$

2017: proving an "alien" theorem?

A conjecture in combinatorics ∞

When one splits $\mathbb{N}$ in 2, one part must contain a Pythagorean triple $(a^2 = b^2 + c^2)$

No known proof, puzzled mathematicians for decades (one offered a 100 \$ reward)

2017: proving an "alien" theorem?

A conjecture in combinatorics

When one splits $\mathbb{N}$ in 2, one part must contain a Pythagorean triple $(a^2 = b^2 + c^2)$

No known proof, puzzled mathematicians for decades (one offered a 100 $ reward)

SAT solver proof[14,22]

200 *TB* proof, compressed to 86 *GB* (stronger proof system)[a]

[a] Oliver Kullmann. "The Science of Brute Force". In: *Communications of the ACM* (2017).

A finitized Gödelian flavor (K. Gödel, 1931)

Whether it's maths or not... Size matters!

- Not only there exists true unprovable statements (in powerful enough consistent sets of axioms[12])
- There may be true provable statements we will never be able to prove because of their extremely long proofs[20]

Is it bio-compatible?

Biology

- Many discrete object ($\{A, T/U, G, C\}$, amino acids, genes, alleles, enzymes...)
- Lots of experimental data

Is it bio-compatible?

Biology

- Many discrete object ($\{A, T/U, G, C\}$, amino acids, genes, alleles, enzymes...)
- Lots of experimental data

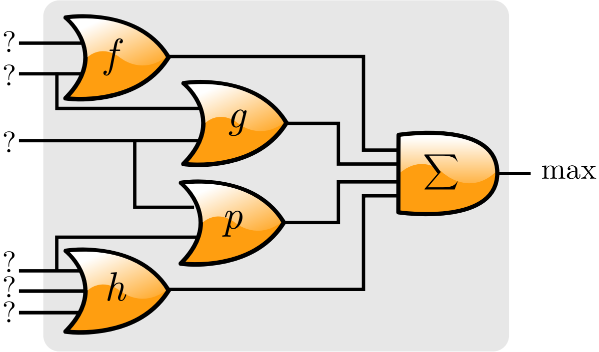
Exploiting Data + knowledge: Machine Learning

- (Stochastic) models can be built from knowledge and data
- And used to predict a "Most Likely/Optimal State" $\Rightarrow$ easily NP-hard

Is it bio-compatible?

Biology

- Many discrete object ($\{A, T/U, G, C\}$, amino acids, genes, alleles, enzymes...)
- Lots of experimental data

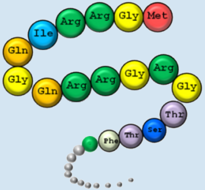


Proteins

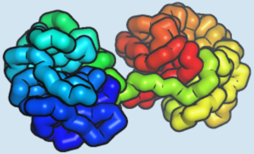
Most active molecules of life

Sequence of “amino-acids”, each chosen among a set of 20 natural ones

Folding



→



→ Fiber

Transporter, binder, regulator, motor, catalyst...
Hemoglobine, TAL effector, ATPase, dehydrogenases...

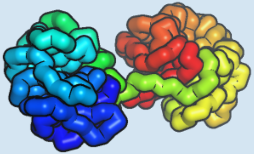
Protein Design

Most active molecules of life

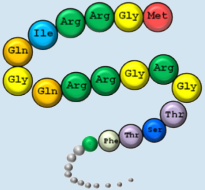
Sequence of “amino-acids”, each chosen among a set of 20 natural ones

Inverse folding

Fiber →



→



Transporter, binder, regulator, motor, catalyst...
Hemoglobine, TAL effector, ATPase, dehydrogenases...

Why is it worth designing new proteins

Eco-friendly chemical/structural nano-agents

- New catalysts for biomass transformation (biofuels, food and feed, cosmetics...),
- New drugs for medicine
- New components for nanotechnologies

Why is it worth designing new proteins

Eco-friendly chemical/structural nano-agents

- New catalysts for biomass transformation (biofuels, food and feed, cosmetics...),
- New drugs for medicine
- New components for nanotechnologies

20^n sequences! intractable for experimental techniques

Why is it worth designing new proteins

Eco-friendly chemical/structural nano-agents

- New catalysts for biomass transformation (biofuels, food and feed, cosmetics...),
- New drugs for medicine
- New components for nanotechnologies

20^n sequences! intractable for experimental techniques

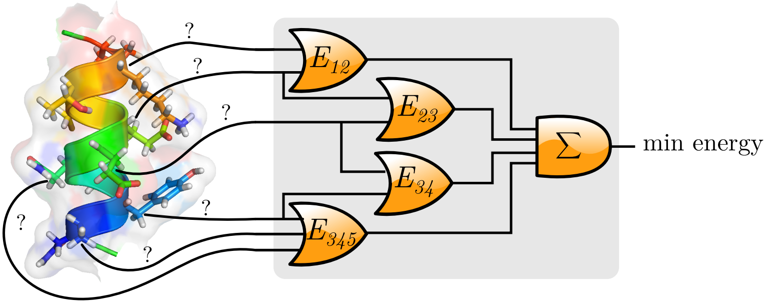
CPD: From bits to atoms From information to functional matter

- mind blowing mass 3d printing-like capacities at atomic level (bacterias)
- structural and functional purposes (powerful origami)
- produced new folds,[19] catalysts,[31] nano-components[36]

Protein Design as the inverse of folding

Ingredients

- Full atom model of a protein backbone (assumed to be rigid)
- Catalog of all 20 amino acids in different conformations ($\approx$ 400 overall)
- Full atom energy function (bonds, electrostatics, solvant, statistics...)
- Maximum stability $\equiv$ Minimum energy NP-hard[28]

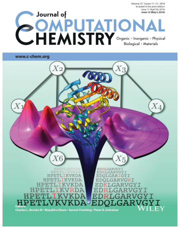


Exact vs. Stochastic search

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

Toulbar2 is able to...

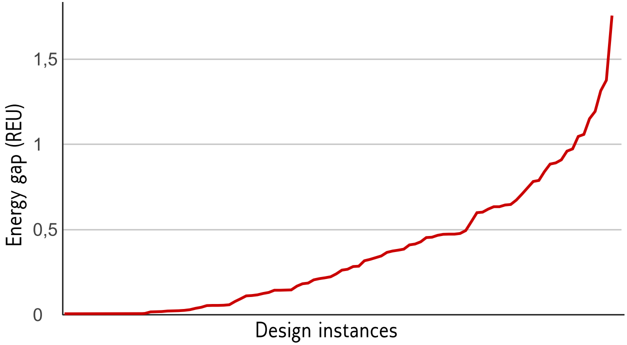
- provide a proven minimum energy solution
- exhaustively enumerate sequences close to it
- in spaces of size $> 10^{200}$



Showed that an highly tuned biased Monte Carlo 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



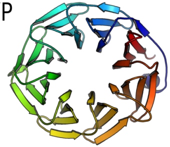
Asymptote: Size matters!

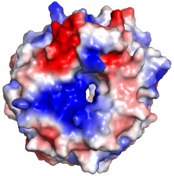
Asymptotic convergence can be arbitrarily slow...

From bits to atoms (TBS, col. A. Voet, KU Leuven, D. Simoncini, INRA/INSA)

C8 pseudo-symetric 2OVP symmetrized into a nano-component

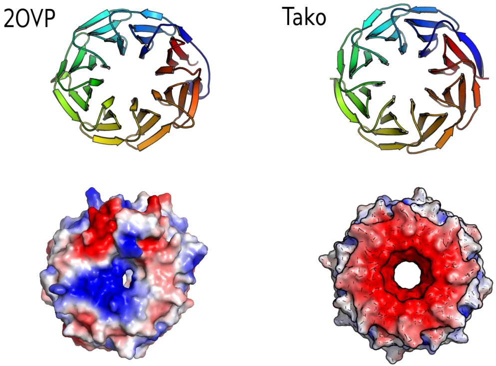
2OVP





C8 pseudo-symetric 2OVP symmetrized into a nano-component

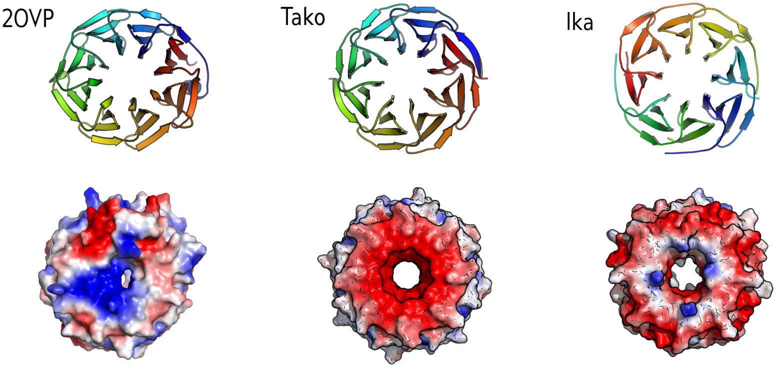
-  Tako: (R)evolution + Rosetta/talaris14 8 fold



From bits to atoms (TBS, col. A. Voet, KU Leuven, D. Simoncini, INRA/INSA)

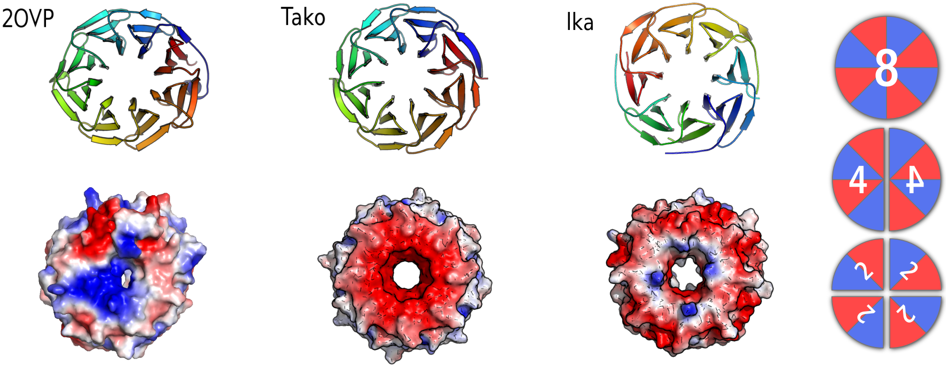
C8 pseudo-symetric 2OVP symmetrized into a nano-component

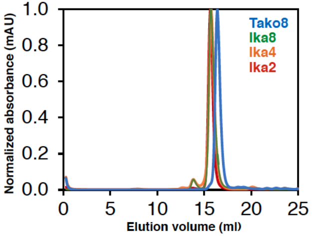
-  Tako: (R)evolution + Rosetta/talaris14 — 8 fold
-  Ika: toulbar2 + talaris14 — 4 fold

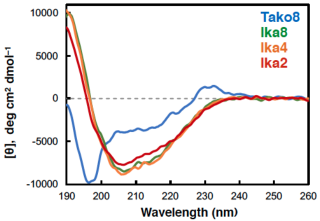


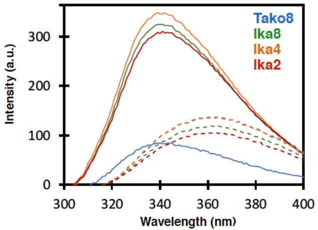
C8 pseudo-symetric 2OVP symmetrized into a nano-component

-  Ika: toulbar2 + talaris14 — 4 fold



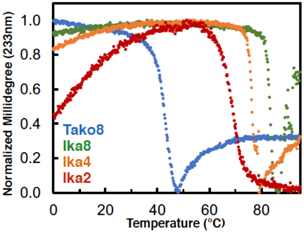


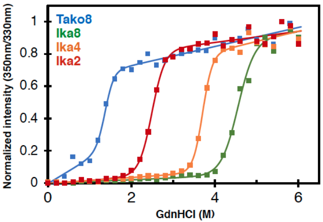




Assemble as 8-bladed propeller

- Ika* more stable than Tako8
- Temperature
- Chemical denaturation





Final messages

Asymptotes: size matters

NP is not exactly as we tend to think

- AIs have made drastic progress in their logical capacities
- This progress also comes from (gradient-free) learning
- More progress is needed to supplement our limited human capacities

Synergies between Logic and Intuition

- Logic can analyze and exploit learnt models (not only Neural Nets)
- Intuition can help logic without tainting it (guidance)

Thanks

AI/toulbar2

S. de Givry (INRA)
G. Katsirelos (INRA)
M. Zytnicki (PhD, INRA)
D. Allouche (INRA)
H. Nguyen (PhD, 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)
D. Simoncini (INSA, Toulouse)
S. Barbe (INSA, Toulouse)
S. Traoré (PhD, CEA)
C. Viricel (PhD)
RosettaCommons (U. Washington)
W. Sheffler (U. Washington)
PyRosetta (U. John Hopkins)
B. Donald (U. North Carolina)
K. Roberts (U. North Carolina)
T. Simonson (Polytechnique)
J. Cortes (LAAS/CNRS)

References I

[1] Albert Atserias, Johannes Klaus Fichte, and Marc Thurley. "Clause-learning algorithms with many restarts and bounded-width resolution". In: *Journal of Artificial Intelligence Research* 40 (2011), pp. 353–373.

[2] Gilles Audemard and Laurent Simon. "Predicting Learnt Clauses Quality in Modern SAT Solvers.". In: *International Joint Conference in AI*. Vol. 9. 2009, pp. 399–404.

[3] Maurice Bruynooghe and Luıs Moniz Pereira. "Deduction revision by intelligent backtracking". In: (1984).

[4] Association for Constraint Programming. *Publication statistics in CP per country every year*. URL http://www.a4cp.org/cparchive/countries_by_year.

[5] Martin C Cooper et al. "Soft arc consistency revisited". In: *Artificial Intelligence* 174.7 (2010), pp. 449–478.

[6] Matthieu Courbariaux et al. "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1". In: *arXiv preprint arXiv:1602.02830* (2016).

[7] Martin Davis, George Logemann, and Donald Loveland. "A machine program for theorem-proving". In: *Communications of the ACM* 5.7 (1962), pp. 394–397.

[8] Martin Davis and Hilary Putnam. "A computing procedure for quantification theory". In: *Journal of the ACM (JACM)* 7.3 (1960), pp. 201–215.

[9] Leonardo De Moura and Nikolaj Bjørner. "Satisfiability modulo theories: introduction and applications". In: *Communications of the ACM* 54.9 (2011), pp. 69–77.

[10] Rina Dechter. "Enhancement schemes for constraint processing: Backjumping, learning, and cutset decomposition". In: *Artificial Intelligence* 41.3 (1990), pp. 273–312.

References II

[11] S Foissac et al. “Genome Annotation in Plants and Fungi: EuGène as a Model Platform”. In: *Current Bioinformatics* 3.2 (2008), pp. 87–97.

[12] Kurt Gödel. “Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I”. In: *Monatshefte für mathematik und physik* 38.1 (1931), pp. 173–198.

[13] Carla P Gomes, Bart Selman, Henry Kautz, et al. “Boosting combinatorial search through randomization”. In: *AAAI/IAAI* 98 (1998), pp. 431–437.

[14] Marijn JH Heule, Oliver Kullmann, and Victor W Marek. “Solving and verifying the boolean pythagorean triples problem via cube-and-conquer”. In: *International Conference on Theory and Applications of Satisfiability Testing*. Springer. 2016, pp. 228–245.

[15] Barry Hurley et al. “Multi-language evaluation of exact solvers in graphical model discrete optimization”. In: *Constraints* (2016), pp. 1–22.

[16] Peter Jeavons and Justyna Petke. “Local consistency and SAT-solvers”. In: *Journal of Artificial Intelligence Research* 43 (2012), pp. 329–351.

[17] Guy Katz et al. “Reluplex: An efficient SMT solver for verifying deep neural networks”. In: *International Conference on Computer Aided Verification*. Springer. 2017, pp. 97–117.

[18] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[19] Brian Kuhlman et al. “Design of a novel globular protein fold with atomic-level accuracy”. In: *science* 302.5649 (2003), pp. 1364–1368.

References III

[20] Oliver Kullmann. "The Science of Brute Force". In: *Communications of the ACM* (2017).

[21] John Lafferty, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In: *Proc. of the 18^{t}h International Conference on Machine Learning (ICML)*. 2001.

[22] Evelyn Lamb. "Maths proof smashes size record: supercomputer produces a 200-terabyte proof–but is it really mathematics?" In: *Nature* 534.7605 (2016), pp. 17–19.

[23] João P Marques-Silva and Karem A Sakallah. "GRASP: A search algorithm for propositional satisfiability". In: *IEEE Transactions on Computers* 48.5 (1999), pp. 506–521.

[24] David A McAllester. *An Outlook on Truth Maintenance*. Tech. rep. Massachusetts Inst Of Tech, Cambridge, Artificial Intelligence Lab., 1980.

[25] Antonio Morgado et al. "Iterative and core-guided MaxSAT solving: A survey and assessment". In: *Constraints* 18.4 (2013), pp. 478–534.

[26] Matthew W Moskewicz et al. "Chaff: Engineering an efficient SAT solver". In: *Proceedings of the 38th annual Design Automation Conference*. ACM. 2001, pp. 530–535.

[27] Nina Narodytska et al. "Verifying properties of binarized deep neural networks". In: *arXiv preprint arXiv:1709.06662* (2017).

[28] Niles A Pierce and Erik Winfree. "Protein design is NP-hard". In: *Protein engineering* 15.10 (2002), pp. 779–782.

[29] John Alan Robinson. "A machine-oriented logic based on the resolution principle". In: *Journal of the ACM (JACM)* 12.1 (1965), pp. 23–41.

[30] Francesca Rossi, Peter Van Beek, and Toby Walsh. *Handbook of constraint programming*. Elsevier, 2006.

References IV

[31] Daniela Röthlisberger et al. "Kemp elimination catalysts by computational enzyme design". In: *Nature* 453.7192 (2008), p. 190.

[32] Thomas Schiex and Gérard Verfaillie. "Nogood recording for static and dynamic constraint satisfaction problems". In: *International Journal on Artificial Intelligence Tools* 3.02 (1994), pp. 187–207.

[33] 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`.

[34] Richard M Stallman and Gerald J Sussman. "Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis". In: *Artificial intelligence* 9.2 (1977), pp. 135–196.

[35] Christian Szegedy et al. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013).

[36] Arnout RD Voet et al. "Computational design of a self-assembling symmetrical β-propeller protein". In: *Proceedings of the National Academy of Sciences* 111.42 (2014), pp. 15102–15107.

Absolutely Reliable but...

We do not understand the sources of their efficiency

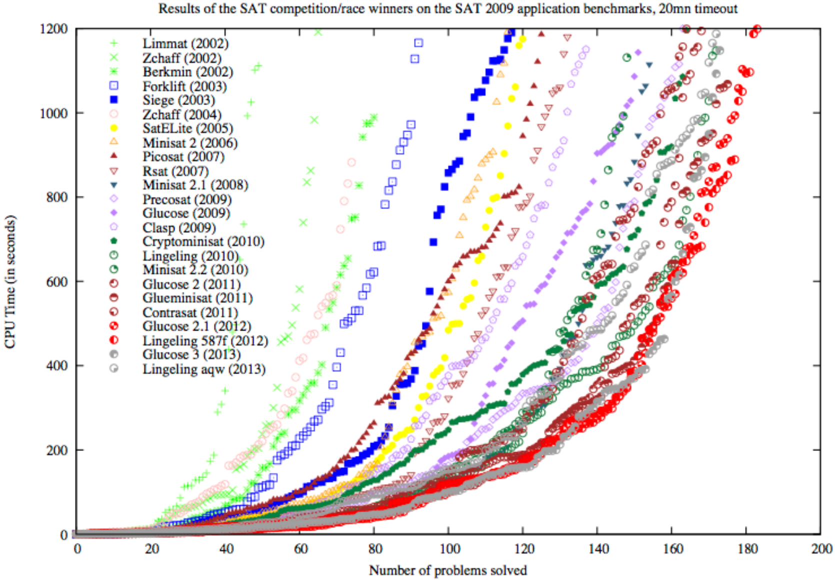
CDCL solvers have an expected polynomial $O(n^k)$ runtime on SAT instances whose primal (Gaifman) graph has treewidth k.

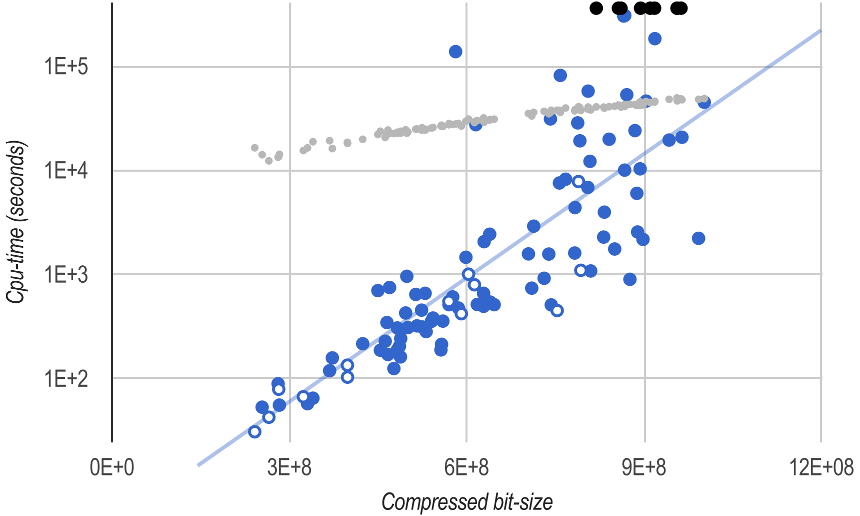
Without ever trying to compute a treewidth/decomposition (NP hard).

Solving PSPACE problems ?

Go on a $n \times n$ goban is PSPACE-hard

- PSPACE-hard to decide if there is a winning strategy
- AlphaGo 0 does not solve 19×19 Go
- It plays better than humans (and that's amazing!)


Results of the SAT competition/race winners on the SAT 2009 application benchmarks, 20mn timeout
CPU Time (in seconds)
Number of problems solved
0
200
400
600
800
1000
1200
0
20
40
60
80
100
120
140
160
180
200
Limmat (2002)
Zchaff (2002)
Berkmin (2002)
Forklift (2003)
Siege (2003)
Zchaff (2004)
SatELite (2005)
Minisat 2 (2006)
Picosat (2007)
Rsat (2007)
Minisat 2.1 (2008)
Precosat (2009)
Glucose (2009)
Clasp (2009)
Cryptominisat (2010)
Lingeling (2010)
Minisat 2.2 (2010)
Glucose 2 (2011)
Glueminisat (2011)
Contrasat (2011)
Glucose 2.1 (2012)
Lingeling 587f (2012)
Glucose 3 (2013)
Lingeling aqw (2013)


1E+5
1E+4
1E+3
1E+2
Cpu-time (seconds)
0E+0
3E+8
6E+8
9E+8
12E+08
Compressed bit-size

Intuition: Restart, forget useless stuff, speed by lazyness

Additional ingredients (patented for some)

(I) stops, restarts with a better understanding of the problem[13]

(I) forgets learnt information predicted as "useless" (Glue clauses[2])

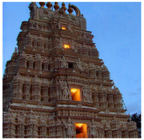
- Lazy data structures[26]

- Absolutely reliable combination of logic and intuition
- but we don't really understand why it can be so efficient[1,16]

Neural nets and safety critical settings

It doesn't seem too hard to fool a standard Convolutional Neural Net[a]

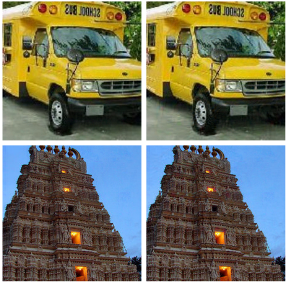
[a] Christian Szegedy et al. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013).



Neural nets and safety critical settings

It doesn't seem too hard to fool a standard Convolutional Neural Net[a]

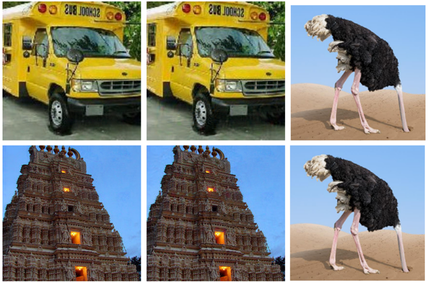
[a] Christian Szegedy et al. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013).



Neural nets and safety critical settings

It doesn't seem too hard to fool a standard Convolutional Neural Net[a]

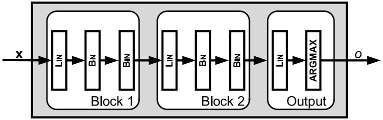
[a] Christian Szegedy et al. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013).



From deep Neural Nets to SAT

Binarized Deep NN: ± 1 activations/weights[6]

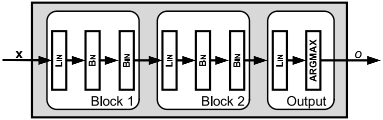
- Lin: affine transformation with learnt binary weights (float bias).
- Bn: (Batch normalization) rescaling with learnt floats.
- Bin: binarization using the *Sign* function.



From deep Neural Nets to SAT

Binarized Deep NN: ± 1 activations/weights[6]

- Lin: affine transformation with learnt binary weights (float bias).
- Bn: (Batch normalization) rescaling with learnt floats.
- Bin: binarization using the *Sign* function.



A learnt block can be described as a SAT[a] formula (SMT(LI)[17] for ReLU)

[a] Nina Narodytska et al. "Verifying properties of binarized deep neural networks". In: *arXiv preprint arXiv:1709.06662* (2017).

Resistance to manipulation

Adversarial Robustness of a classifier

A positive test input cannot be slightly modified to change class

Resistance to manipulation

Adversarial Robustness of a classifier

A positive test input cannot be slightly modified to change class

Certified robusteness by SAT

As a SAT formula: Neural Net + input + bounded perturbation + missclassification

Resistance to manipulation

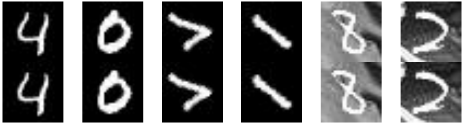
Adversarial Robustness of a classifier

A positive test input cannot be slightly modified to change class

Certified robusteness by SAT

As a SAT formula: Neural Net + input + bounded perturbation + missclassification

- MNIST dataset, 4 blocks BNN with 100 to 200 neurons per layer, L_∞ norm
- Millions of clauses: Glucose[2] certifies (non) robustness for most input in $< 5'$ CPU time

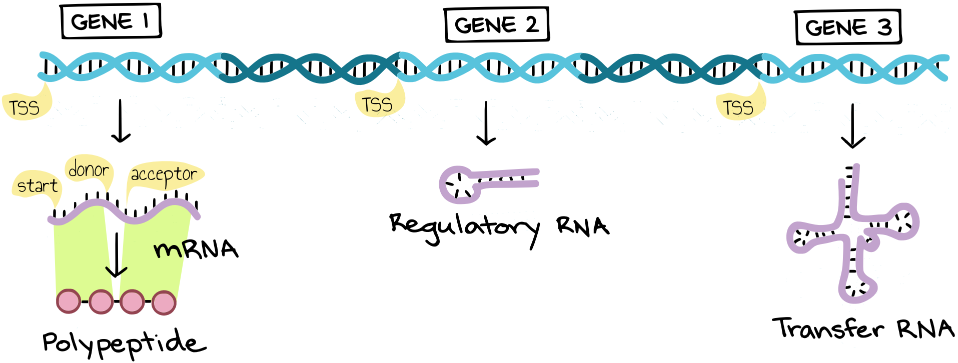


Deciphering genomic DNA

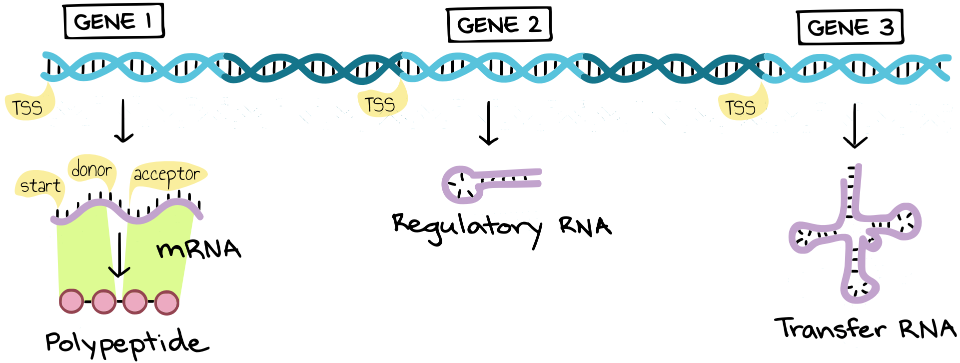
DNA

TATCATACCTTGTGACGAATGCTTTGGAAAATAATTTTGAGATGAGCTTTTAAGGACATAACACAATTGCAAAATGGAACACAATTGCAAAATGGAACACAATTGTTTTACAAACCGAGAGCATCGAGAGTAGGTCGGATGTCACATCGATTAAATGTGACCCTCGTGCACTGTGAGCTCTAGCC

Segmenting genomic DNA



Segmenting genomic DNA



Similarities	Conservation	Sequence stats	Site predictors
RNA-Seq Rfam UniProt		Markov chains	SVM Neural nets

Optimization + decomposable probability distribution (Semi-CRF)[21]

- Derived from an actual human processor (S. Rumbauts, PhD)[a]
- Discriminative learning (don't try to model evidence!)
- Optimizes an empirical loss function (performance on a testing set: quality is crucial)

[a] S Foissac et al. "Genome Annotation in Plants and Fungi: EuGène as a Model Platform". In: *Current Bioinformatics* 3.2 (2008), pp. 87–97.






...

Prediction is in P

Main difficulty: collecting evidence, training and testing.