Attention learning

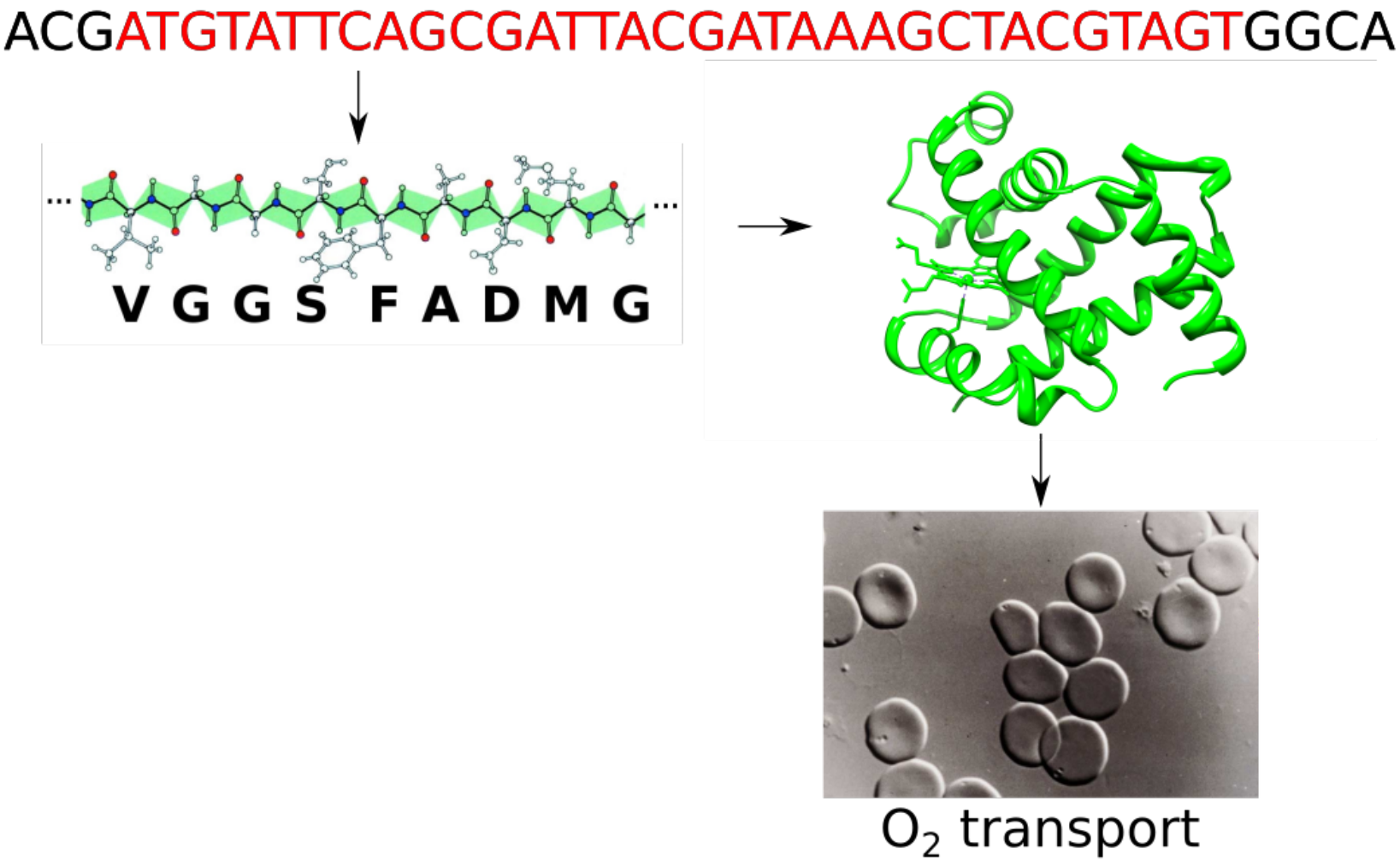
and structural biology application

Clovis Galiez

Grenoble
Statistiques pour les sciences du Vivant et de l'Homme

October 22, 2024

Protein sequence and structure



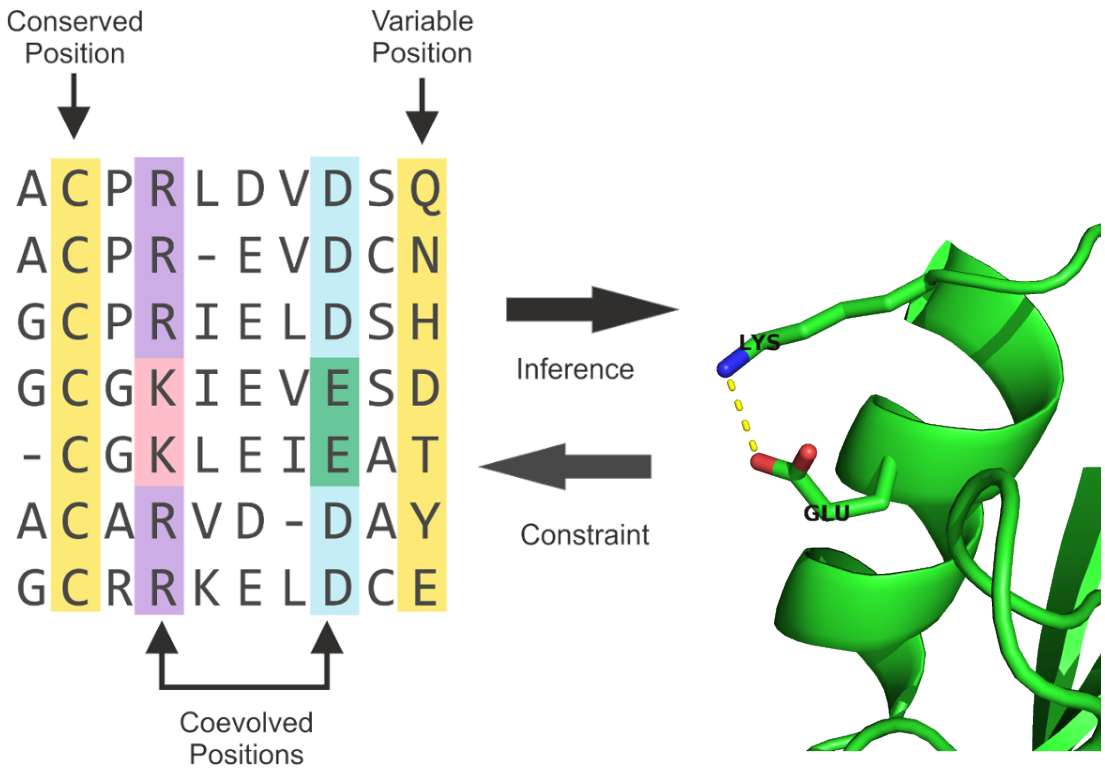
CASP competition

Blind competition. Simple principle:
- a sequence is given
- have to predict the structure.

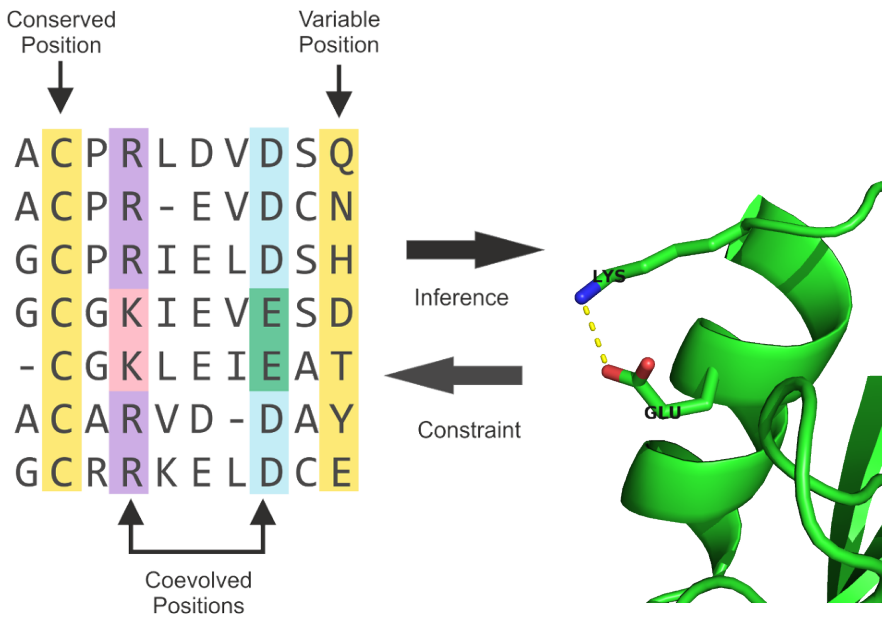
Prior to 2018 it used to be (pseudo) physical models that where best performing.

Preserving the function: coevolution of residues

As protein function is vital, **evolution selects mutations preserving structures**.
Leading to **compensatory** mutations:



A simple approach for protein structure prediction



- Build or get multiple manio acid sequence alignments (**e.g.** in Pfam database)
- Quantify coevolution between positions in the sequence
- Infer what are the position in contact

What measure for co-evolution? Correlation would work?

Conservation vs. co-evolution

A standard approach is to measure it through Mutual Information:

$$MI(i,j) = \sum_{a,b} p(x_i = a, x_j = b) \log \frac{p(x_i{=}a, x_j{=}b)}{p(x_i{=}a)\, p(x_j{=}b)}$$

Where

- x_i is the amino acid at position i
- $p(x_i = a)$ is estimated in the MSA by $\frac{\#\text{sequences having "a" at position } i}{N}$
- N the number of sequences in the MSA
- $p(x_i = a, x_j = b)$ is estimated in the MSA by $\frac{\#\text{sequence having "a" at } i \text{ and "b" at } j}{N}$

In paractice you need $N > 1,000$ to have reasonable estimation of $p(x_i = a, x_j = b)$.

From heuristic to learning

There are two drawbacks:

From heuristic to learning

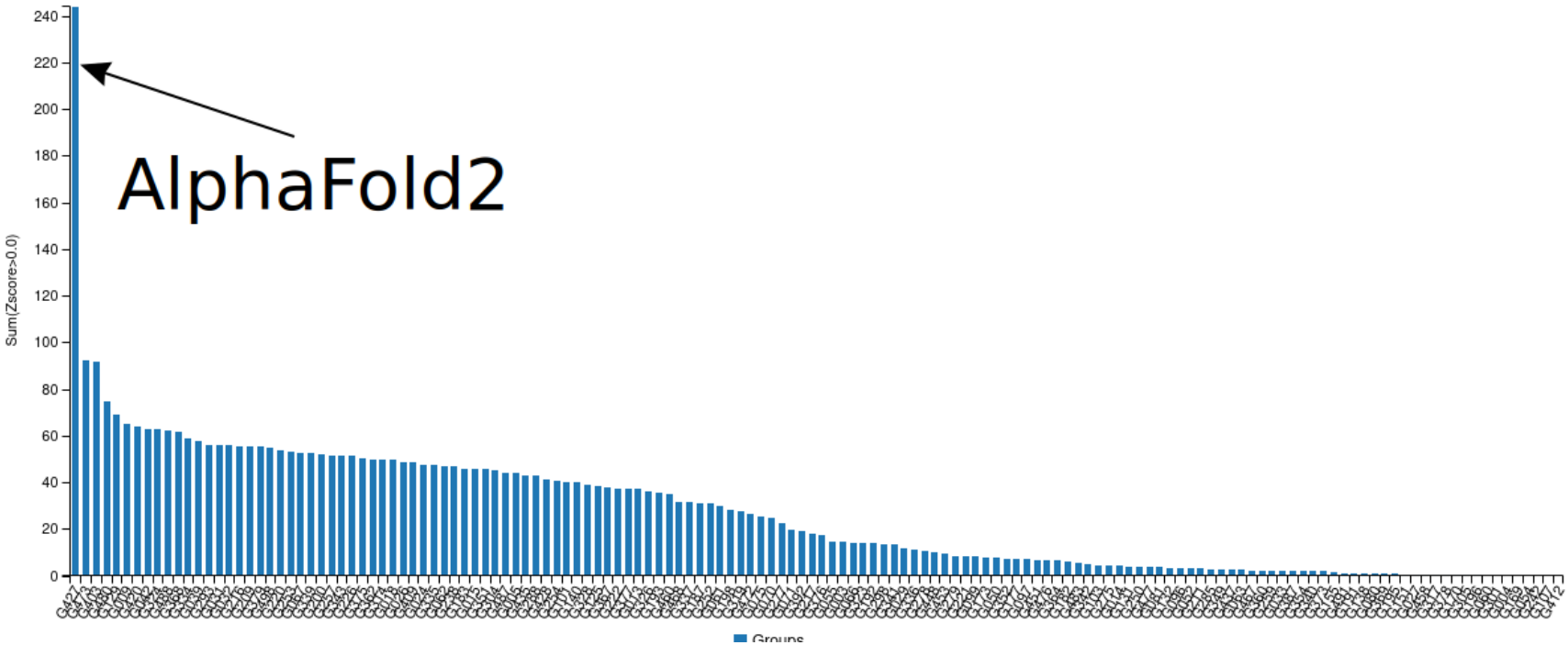
There are two drawbacks:

- Spurious correlation/noisy estimate of MI when having few aligned sequences
- MI may not be the best option to detect correlated mutation due to 3D interaction

This is where Transformers have been used: localize portion in the MSA that are of interest to predict 3D interactions.

CASP14 (2020)

"The big leap forward"



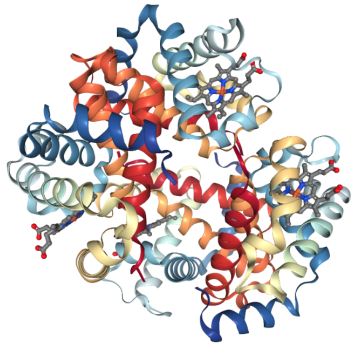
Alphafold2: attention-based learning on protein sequence alignments [Casp14.]

Nature's article.

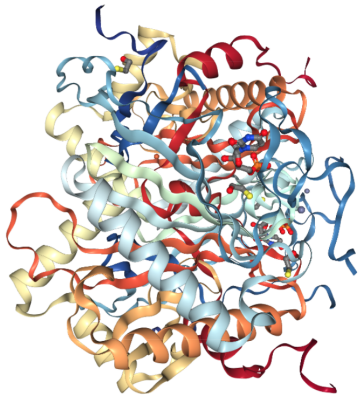
Protein structure prediction as "language translation"

Task: predict the structure from sequence

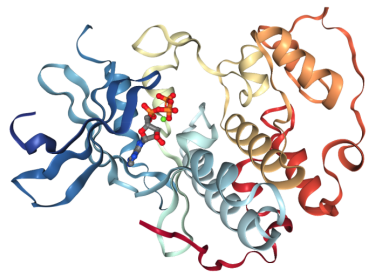
```
>1A3N:A|PDBID|CHAIN|SEQUENCE
VLSPADKTNVKAAWGKVGAHAGEYGAEALER
MFLSFPTTKTYFPHFDLSHGSAQVKGHGKKV
ADALTNAVAHVDDMPNALSALSDLHAHKLRV
DPVNFKLLSHCLLVTLAAHLPAEFTPAVHAS
LDKFLASVSTVLTSKYR
```



```
>1HXP:A|PDBID|CHAIN|SEQUENCE
MTQFNPVDHPHRRYNPLTGQWILVSPHRAKRPW
EGAQETPAKQVLPAHDPDCFLCAGNVRVTGDKN
PDYTGTYVFTNDFAALMSDTPDAPESHDPLMRC
QSARGTSRVICFSPDHSKTLPELSVAALTEIVK
TWQEQTAELGKTYPWVQVFENKGAAMGCSNPHP
HGQIWANSFLPNEAEREDRLQKEYFAEQKSPML
VDYVQRELADGSRTVVETEHWLAVVPYWAAWPF
ETLLLPKAHVLRITDLTDAQRSDLALALKKLTS
RYDNLFQCSFPYSMGWHGAPFNGEENQHWQLHA
HFYPPLLRSATVRKFMVGYEMLAETQRDLTAEQ
AAERLRAVSDIHFRESGV
```



```
>1HCK:A|PDBID|CHAIN|SEQUENCE
MENFQKVEKIGEGTYGVVYKARNKLTGEVVAL
KKIRLDTETEGVPSTAIREISLLKELNHPNIV
KLLDVIHTENKLYLVFEFLHQDLKKFMDASAL
TGIPLPLIKSYLFQLLQGLAFCHSHRVLHRDL
KPQNLLINTEGAIKLADFGLARAFGVPVRTYT
HEVVTLWYRAPEILLGCKYYSTAVDIWSLGCI
FAEMVTRRALFPGDSEIDQLFRIFRTLGTPDE
VVWPGVTSMPDYKPSFPKWARQDFSKVVPPLD
EDGRSLLSQMLHYDPNKRISAKAALAHPFFQD
VTKPVPHLRL
```



Attention learning

Introductory example: NL translation

Goal

Design an AI system that is able to translate texts from one natural language to another, *e.g.* English to French.

"Naïve" possibility:

- Use a pre-defined dictionary
- Couple the dictionnary with Formal Grammars for English (parsing) and French (Generative)

Introductory example: NL translation

Goal

Design an AI system that is able to translate texts from one natural language to another, *e.g.* English to French.

"Naïve" possibility:

- Use a pre-defined dictionary
- Couple the dictionnary with Formal Grammars for English (parsing) and French (Generative)

Issue

A lot of language engineering involved, need to re-do the work for all (pairs of) languages.

Machine learning for translation (2010's)

Goal

Use a corpus of texts i in two versions each (e_i, f_i), in English and in French, respectively.

Attempt 1 for ML-based translation

Any guess?

Attempt 1 for ML-based translation

Any guess?
Word-by-word translation: learn a dictionnary.

$$p(y_i|x_i) \tag{1}$$

Brain equivalent

- Look at a word, and translate it
- Forget about this word

Attempt 1 for ML-based translation

Any guess?
Word-by-word translation: learn a dictionnary.

$$p(y_i|x_i) \tag{1}$$

Brain equivalent

- Look at a word, and translate it
- Forget about this word

Issues

- Lack of context "What a nice example" $\rightarrow$ "Quoi un gentil exemple"
- Words are not aligned (y_i does not corresponds to x_i, nor even same length: "A writing machine" $\leftrightarrow$ "Une machine à écrire"

Attempt 2

Store some information about the context, possibly both from the input and already preocessed output sequence.

- $p(h_i|x_1, ...x_i)$: h_i describes the state of the ith input word
- $p(s_i|y_1, ...y_{i-1})$: s_i describes the state of the ith output word
- $p(y_i|x_i, h_i, s_i)$

Attempt 2

Store some information about the context, possibly both from the input and already preocessed output sequence.

- $p(h_i|h_{i-1}, x_i)$: h_i describes the state of the ith input word
- $p(s_i|s_{i-1}, y_{i-1})$: s_i describes the state of the ith output word
- $p(y_i|x_i, h_i, s_i)$

Attempt 2

Store some information about the context, possibly both from the input and already preocessed output sequence.

- $p(h_i|h_{i-1}, x_i)$: h_i describes the state of the ith input word
- $p(s_i|s_{i-1}, y_{i-1})$: s_i describes the state of the ith output word
- $p(y_i|x_i, h_i, s_i)$: the i^{th} output should depend on the i^{th} input and states

Issues

- Across languages, words does not align one-to-one

Attempt 3

Translate chunk (i.e. sentence) by chunk: learn a representation of the sentence, and then translate it.

[1] Note that vanilla RNNs have a vanishing gradient issue, prefer to use LSTM as in [Sutskever, NIPS'14]

Attempt 3

Translate chunk (i.e. sentence) by chunk: learn a representation of the sentence, and then translate it. Can be done using a encoder-decoder RNN[1] structure:

- $p(h_i|h_{i-1}, x_i)$
- $p(s_j|s_{j-1}, y_{j-1})$
- $p(y_j|h_L, s_j)$

Where L is the input sequence length.

[1] Note that vanilla RNNs have a vanishing gradient issue, prefer to use LSTM as in [Sutskever, NIPS'14]

Attempt 3: still some issues...

Brain equivalent

- Read a sentence, memorize its meaning
- Translate it

Any drawback?

Attempt 3: still some issues...

Brain equivalent

- Read a sentence, memorize its meaning
- Translate it

Any drawback?

Pros and cons

- + Circumvented the alignment problem
- - Need to get very expressive representation to store all information of a sentence into a latent variable $\Rightarrow$ for long sentences, the information is distored

Translation challenge

Look at this dataset ($n = 1$) of translations:
Hello World!/enihcam wen eht gniyrt m'I/It works/sretirw rof tluciffid kool yam ti tuB
Hello World!/I'm trying the new machine/It works/But it may look difficult for writers

Translation challenge

Look at this dataset ($n = 1$) of translations:
Hello World!/enihcam wen eht gniyrt m'I/It works/sretirw rof tluciffid kool yam ti tuB
Hello World!/I'm trying the new machine/It works/But it may look difficult for writers

Exercise

How would you translate "Hey/ti tog uoy kniht I/Well done!/!stargnoC"

Translation challenge

Look at this dataset ($n = 1$) of translations:
Hello World!/enihcam wen eht gniyrt m'I/It works/sretirw rof tluciffid kool yam ti tuB
Hello World!/I'm trying the new machine/It works/But it may look difficult for writers

Exercise

How would you translate "Hey/ti tog uoy kniht I/Well done!/!stargnoC"
"Hey/I think you got it/Well done!/Congrats!"

How can a machine learn from such examples how to translate new texts?
What's your brain doing for translating this?

Translation challenge: needed structure

- h_i: input state (parity of current line number)
- s_j: output state
- $\alpha_{i,j}$: alignment score matrix between the ith input state and the jth output state.

By having the α at hand, you know what letter of the input data you have to translate.

What would be perfect

- $p(h_i|x_1, ...x_i)$
- $p(s_j|y_1, ...y_{j-1})$
- $p(y_j|h_{\alpha(j)}, x_{\alpha(j)}, s_j)$

Where $\alpha(j)$ is the word of the input sequence aligned to j.

Breakthrough

In 2015, [Bahdanau, Cho and Bengio] made a big breakthrough by learning s, h and α from the data!

How is it possible?

It is just as surprising as conv nets manage to learn the patterns that are important to make a prediction (detection of ears, eyes, etc. to classify cats from planes images).

An attention system will learn where it is important to look at in the input and output data to make a prediction.

The principle is just as this simple.

Why it didn't appear before?

Any guess?

Why it didn't appear before?

Any guess?

- A lot of parameters $\rightarrow$ high computational power needed
- A lot of parameters $\rightarrow$ a lot of data to train on

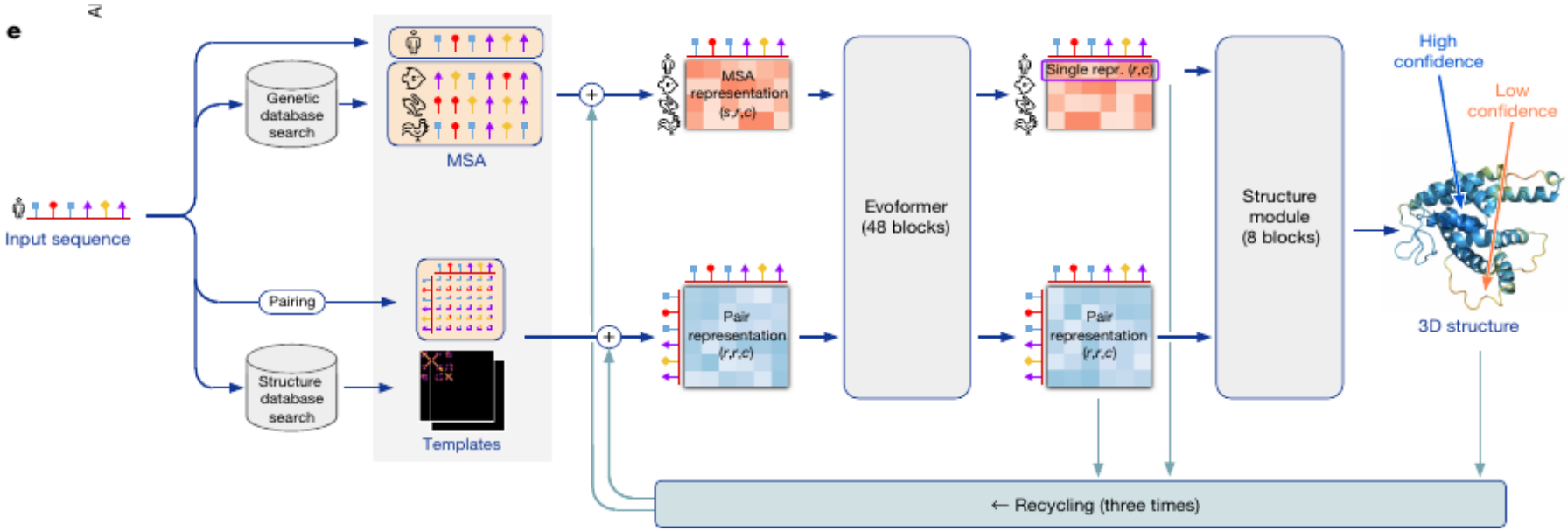
Labeled data (i.e. set of sentences in two different languages) for supervised learning is good, but much information can actually be catch unsupervisedly (and thus make use of bigger data). How?

Self-attention

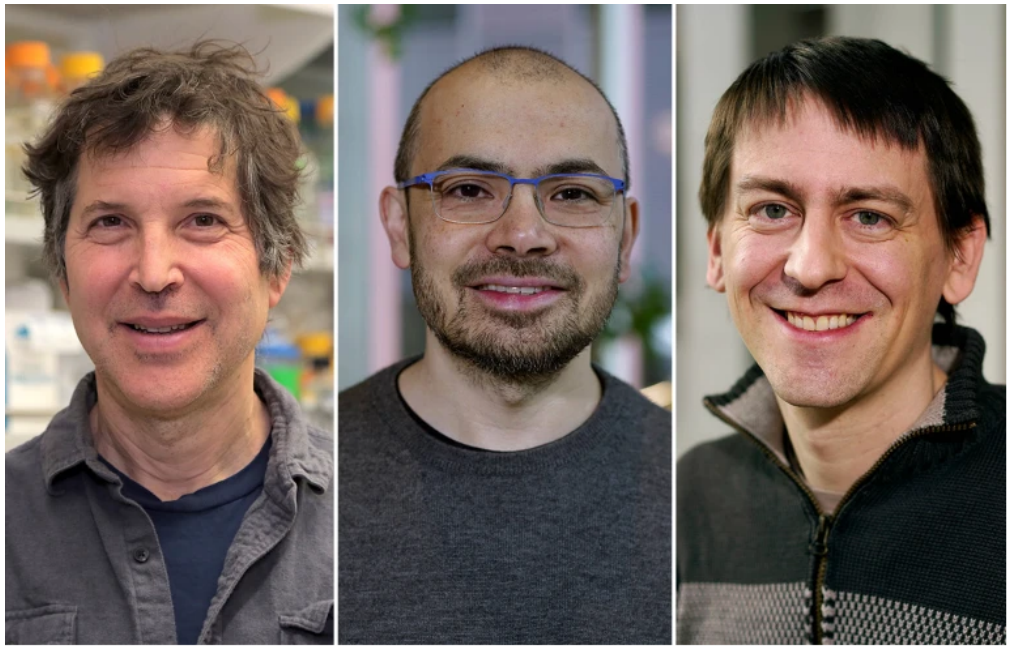
Just use the same mechanism, but trained on predicting a masked word in a sentence.
Most of the information constituting the important projections (e.g. type of words for queries, key and value projection) are actually already present in this data.

Alphafold2 Evoformer

Alphafold2 pipeline



2024's Nobel Prize in Chemistry



David Baker, Demis Hassabis and John Jumper

Discussion and questions?