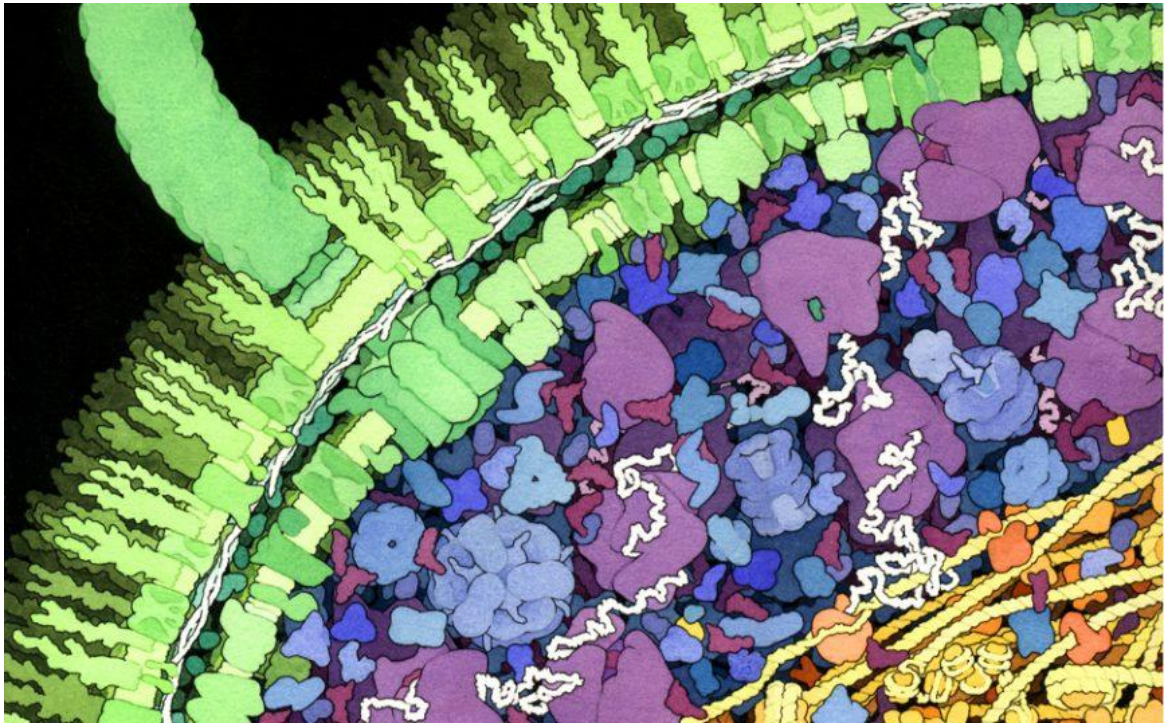
Probabilistic programming and the protein folding problem

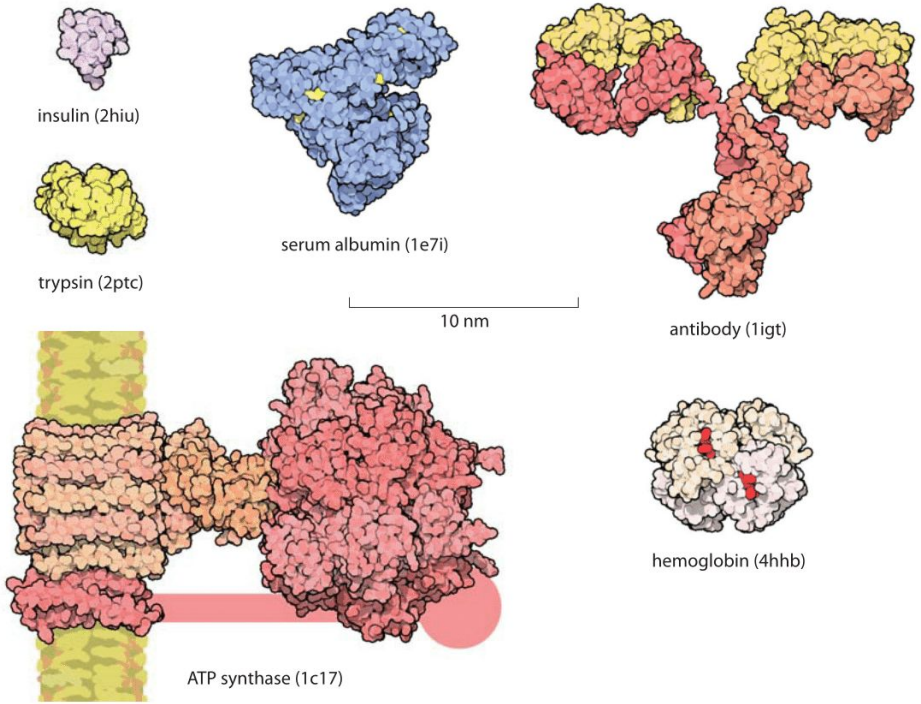
Thomas Hamelryck
Dpt. of Biology and Dpt. of Computer Science (DIKU)
University of Copenhagen, Denmark
AI Seminar, KU, Sep. 2019

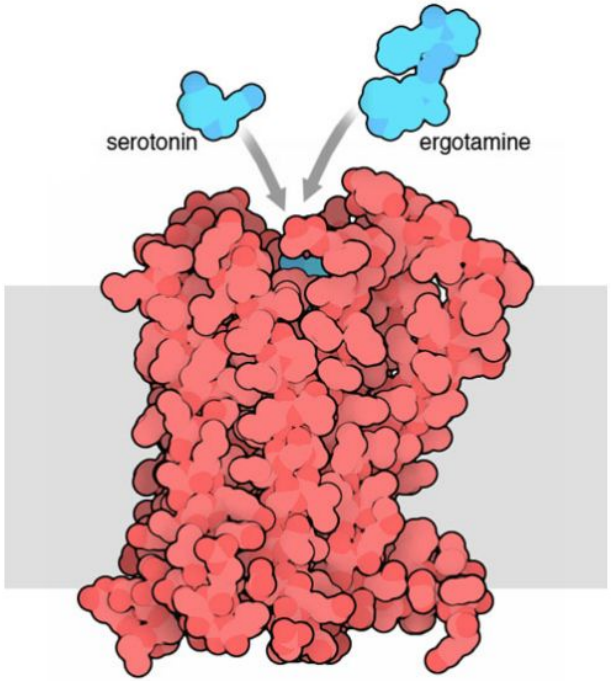
The cell - 50% protein



Picture: A bacterial cell by David Goodsell

Protein structure



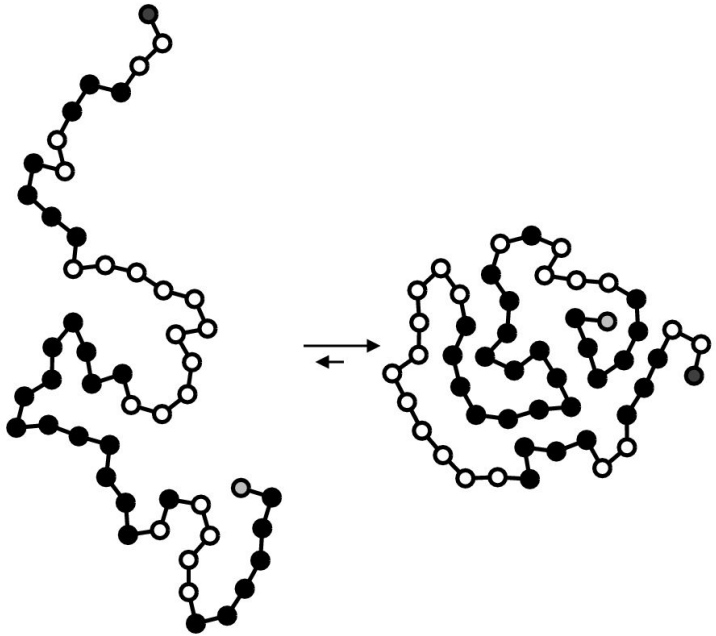


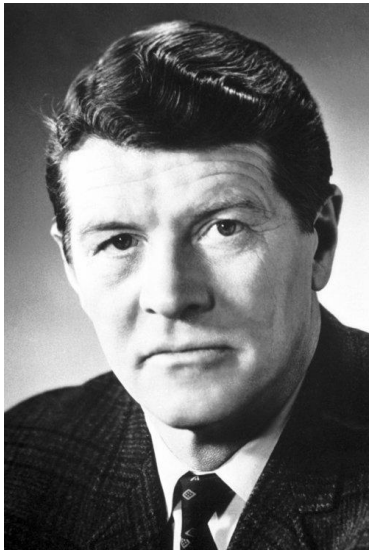
Serotonin receptor with an antimigraine drug. The membrane is shown schematically in gray.

Pictures: Protein structures by David Goodsell

Protein sequence determines fold

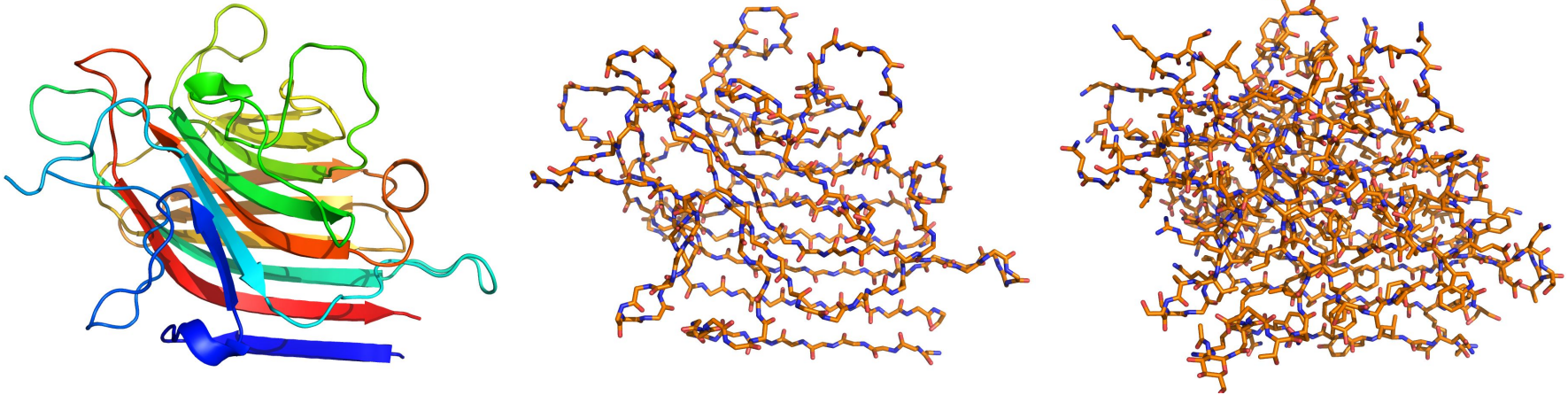
ANERGHPKLA
AFGVGHSAQW
PNTSVHGFDZ
PLQACVTSSH
KLRTNNMKLW





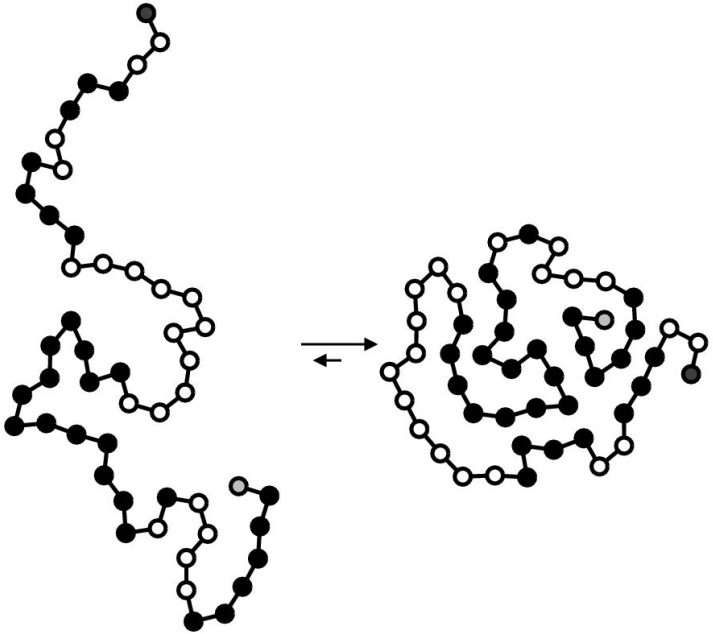
Christian Anfinsen
(1916-1995)

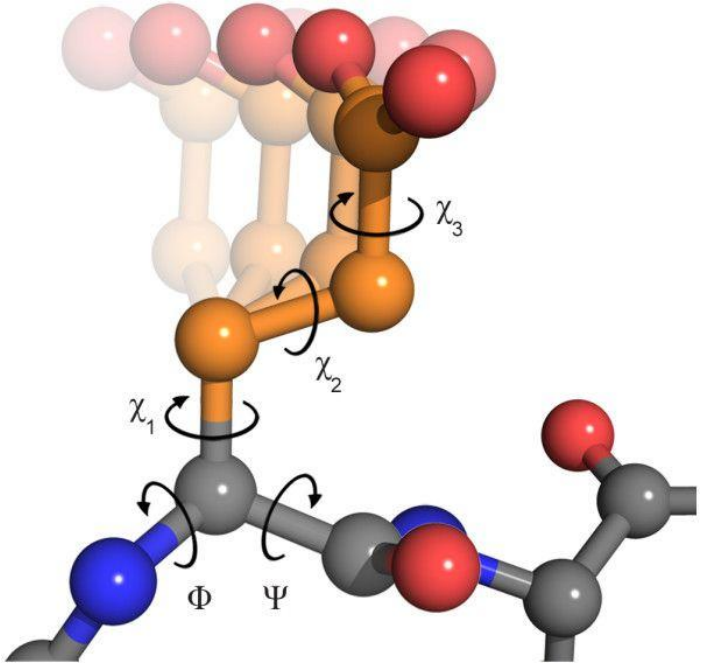
It's complicated



Pictures: PHA-L, 231 amino acids (PDB: 1FAT, Hamelryck et al., 1996)

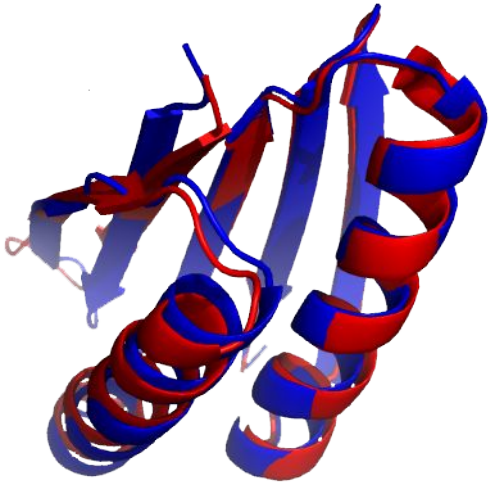
Protein folding and dihedral angles

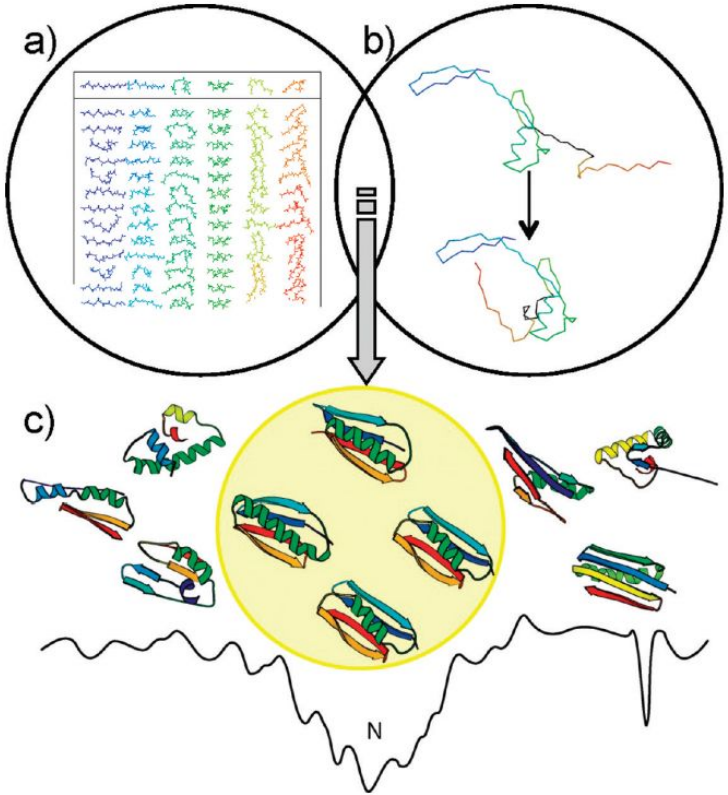




1997: Rosetta

Simons *et al.*, **J. Mol. Biol.**, 1997

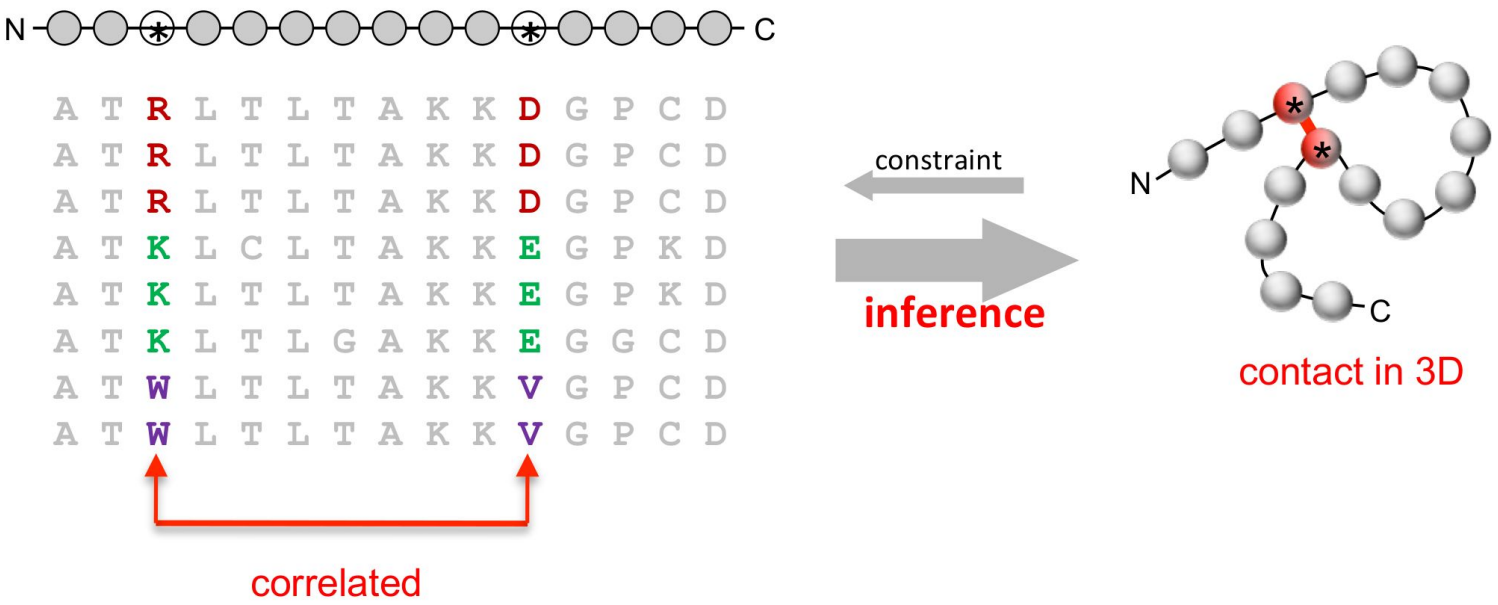




Pictures: Kaufman et al., Biochemistry, 2010 and Wikipedia

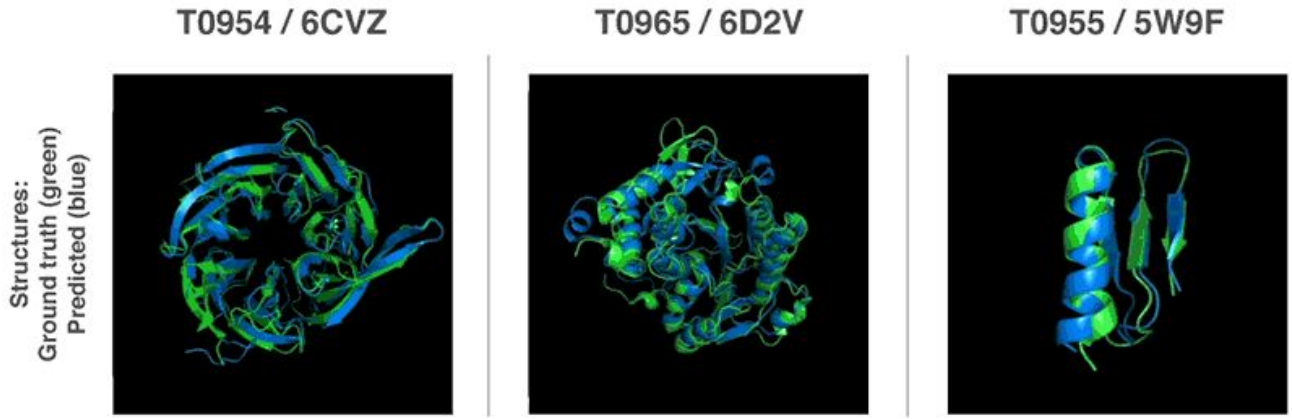
2011: EVfold

Marks *et al.*, **PLoS ONE**, 2011



2018: DeepMind's AlphaFold

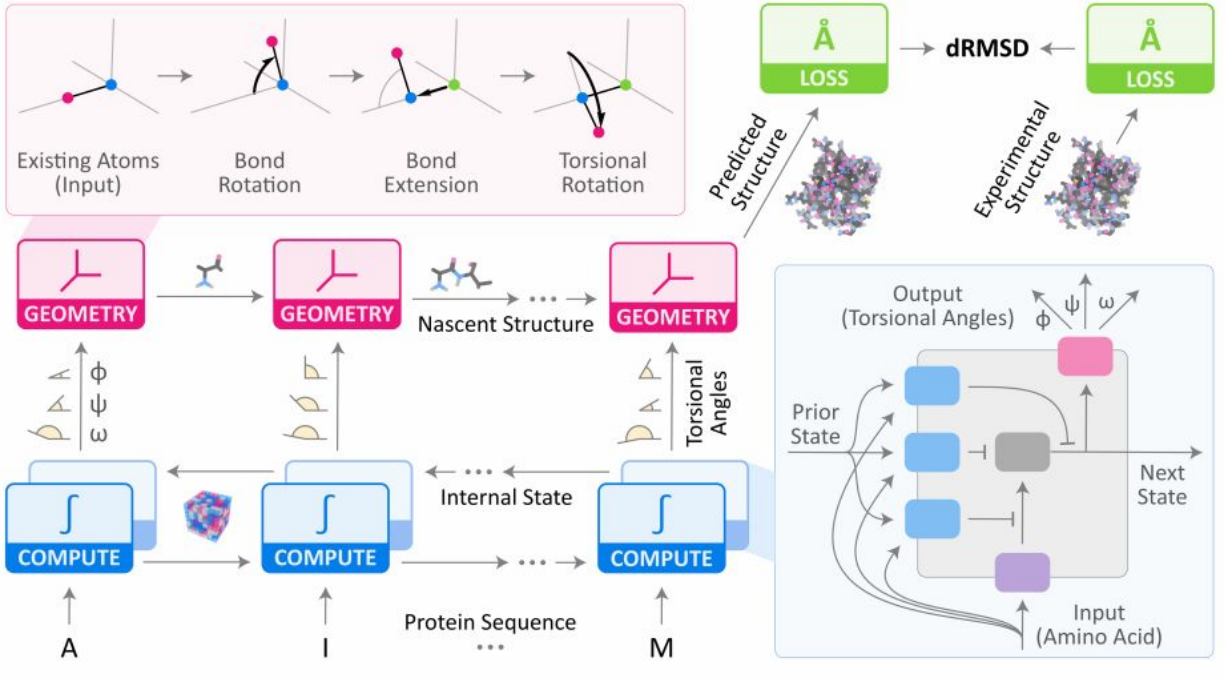
- "*Unprecedented progress in the ability of computational methods to predict protein structure.*", CASP13
- AlphaFold: 25/43 most accurate prediction
  - Second best: 3/43



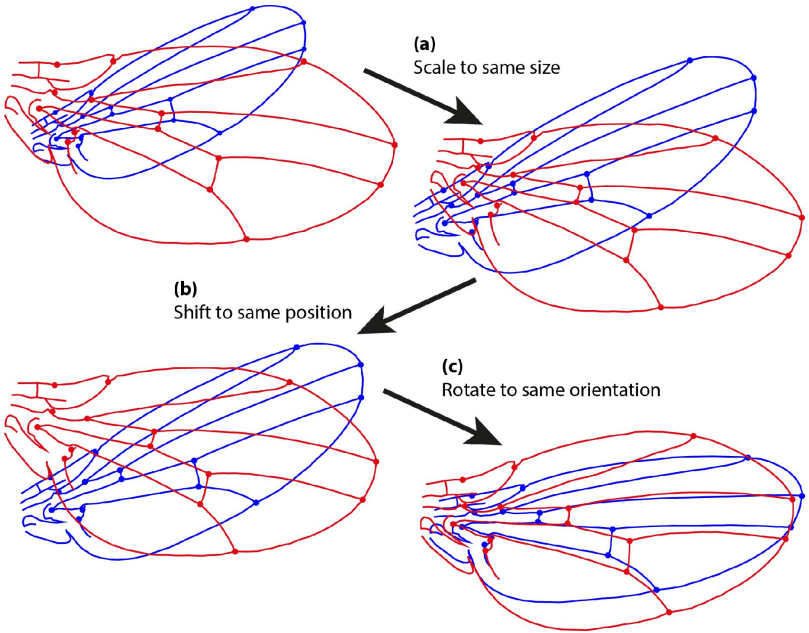
Pictures: AlphaFold

2018: AlQuraishi's end-to-end prediction

- Mohammed AlQuraishi, **Cell Systems**, 2019 (**bioRxiv**, 2018)



Procrustes problems

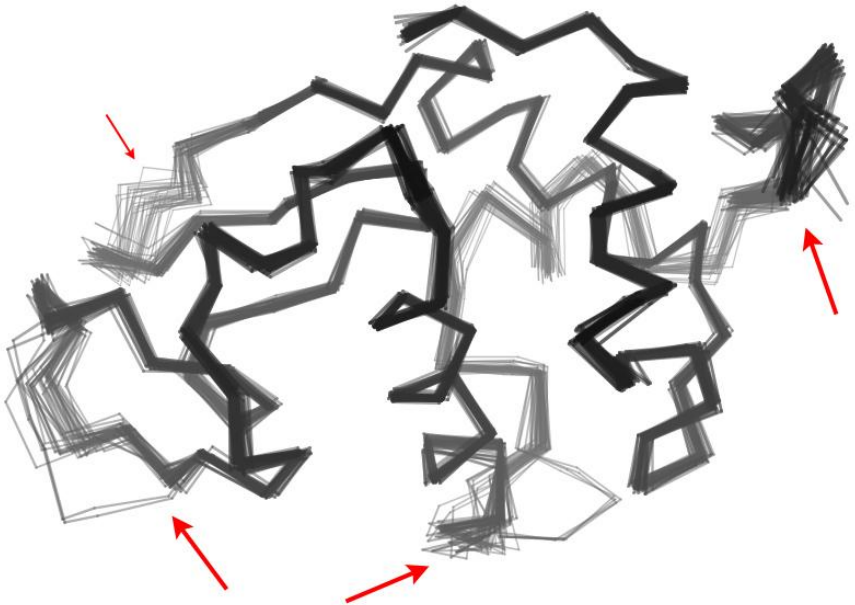


Theseus slaying Procrustes.

Pictures: Wikipedia / ancientgreecereloaded.com

Point estimates versus distributions

- **Epistemic randomness**
  - Randomness due to modelling
  - Measurement noise
  - Limited data
  - Poor model
  - Estimation procedures
- **Aleatory randomness**
  - Intrinsic randomness



An "ensemble" of 44 crystal structures of lysozyme.

Picture: Wikipedia

The Bayesian calculus

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

$$p(\theta \mid \mathbf{d}) = \frac{p(\mathbf{d} \mid \theta)\pi(\theta)}{p(\mathbf{d})}$$

Thomas Bayes
(1701-1761)

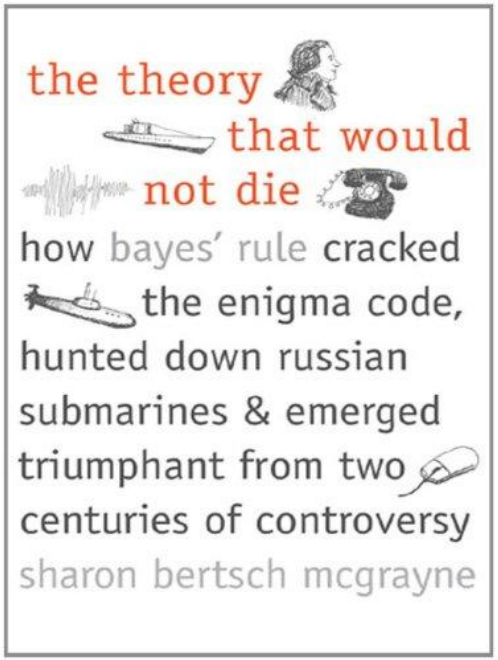
The Bayesian calculus

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

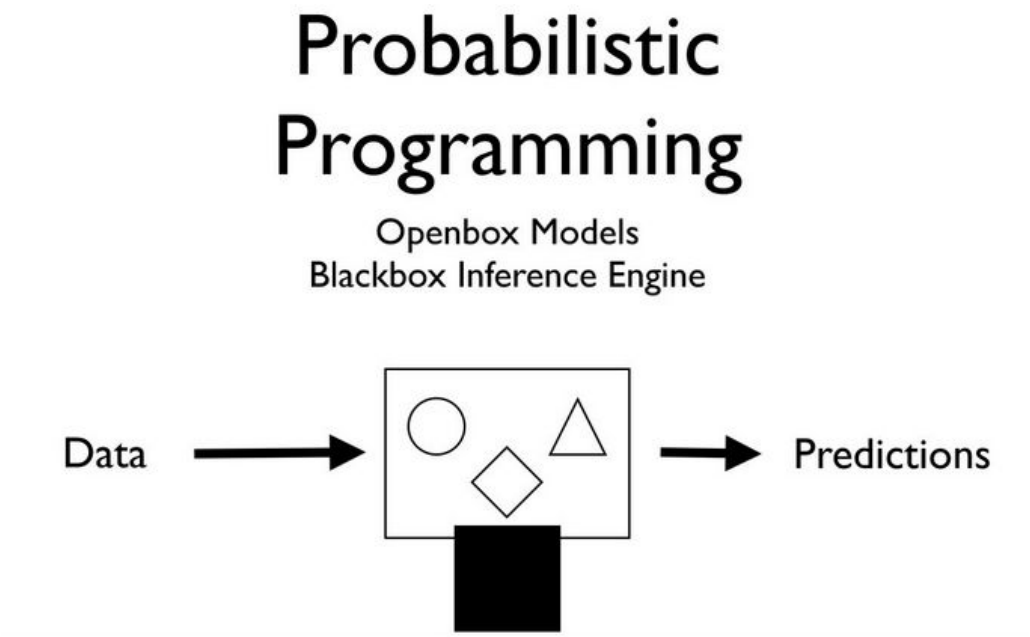
$$p(\boldsymbol{\theta} \mid \mathbf{d}) = \frac{p(\mathbf{d} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{p(\mathbf{d})}$$

Pierre-Simon Laplace
(1749-1827)

The theory that would not die



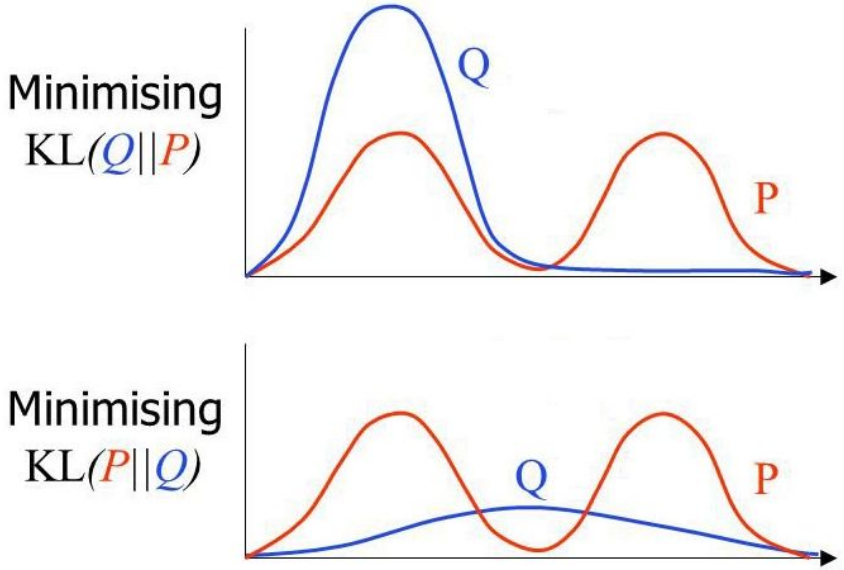
Bayes in the 21st century



Picture: Olivier Grisel's talk on ML

Automatic inference

$$\frac{\partial \theta}{\partial t} = \frac{\partial E_{kin}}{\partial p} = \frac{p}{m}$$
$$\frac{\partial p}{\partial t} = -\frac{\partial E_{pot}}{\partial \theta}$$



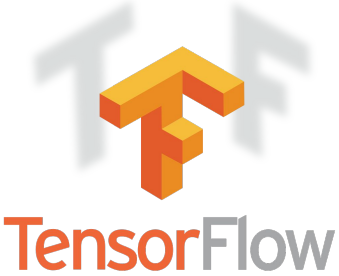
Sampling
Hamiltonian Monte Carlo / NUTS (2011)

Optimisation
Variational inference

Pictures: Mathieu Lê / John Winn, Microsoft Research Cambridge (adapted)

Deep probabilistic programming

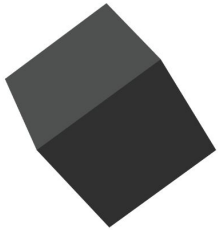
theano



PYTORCH

PyMC3

Edward



Pyro

Bayesian linear model

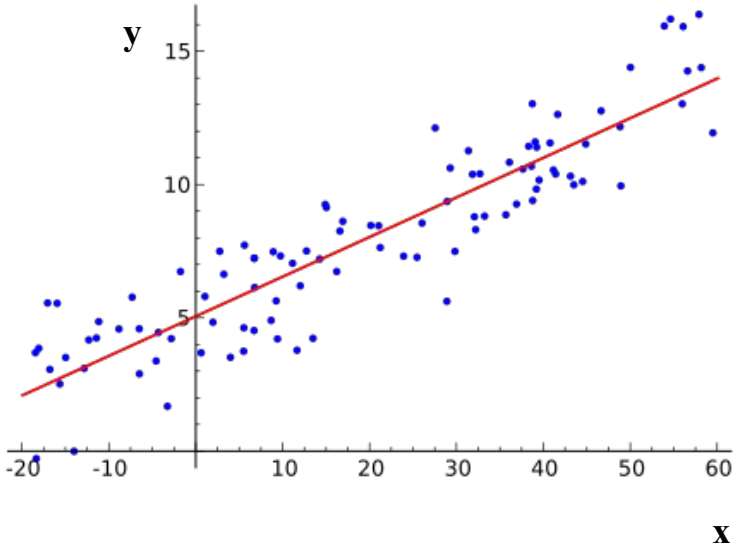
$$a \sim N(a \mid 0, 1)$$

$$b \sim N(b \mid 0, 1)$$

$$\sigma \sim N_+(0, 1)$$

$$y \sim N(y \mid \mu, \sigma)$$

$$\mu = a + bx$$



Picture (right): Wikipedia

Bayesian linear model in PyMC3

$$a \sim N(a \mid 0, 1)$$

$$b \sim N(b \mid 0, 1)$$

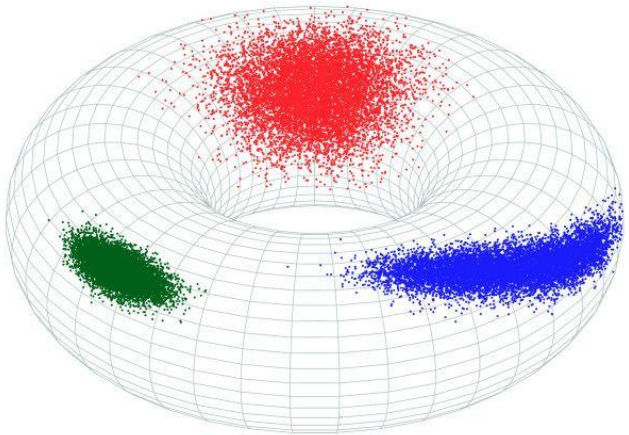
$$\sigma \sim N_+(0, 1)$$

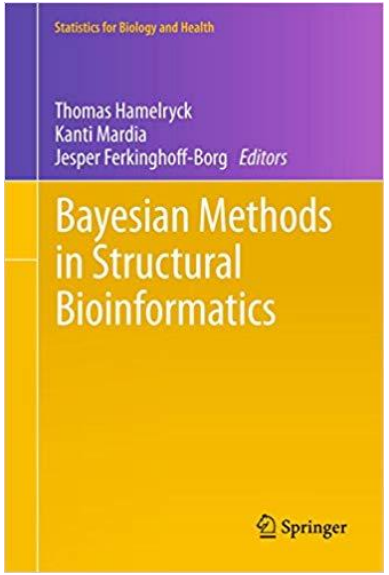
$$y \sim N(y \mid \mu, \sigma)$$

$$\mu = a + bx$$

```
from pymc3 import *

with Model() as model:
    # Define priors
    a = Normal('a', 0, sd=1)
    b = Normal('b', 0, sd=1)
    sigma = HalfNormal('sigma', 1)

    # Define likelihood
    likelihood = Normal('y', mu=a+b*x,
        sd=sigma, observed=y)

    # Inference!
    start = find_MAP() # Find starting value
    step = NUTS() # Sampling
    trace = sample(2000, step, start=start)
```

How I got involved



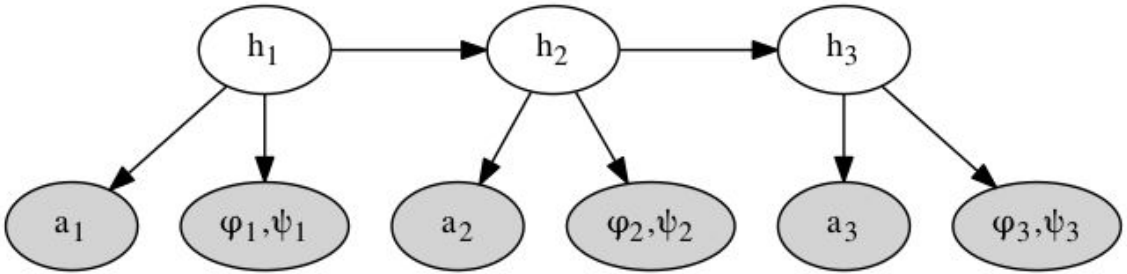


$$f(\phi,\psi) = Z_c(\kappa_1,\kappa_2,\kappa_3)\ \exp[\kappa_1\cos(\phi-\mu)+\kappa_2\cos(\psi-\nu)-\kappa_3\cos(\phi-\mu-\psi+\nu)]$$

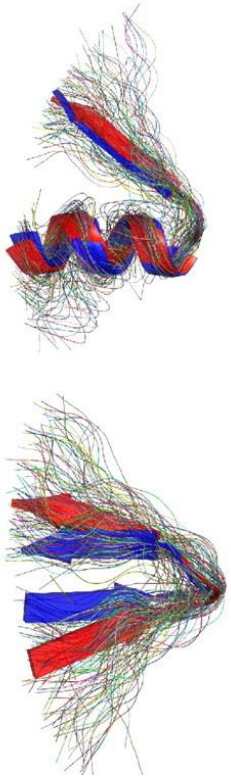
References: TorusDBN, PNAS, 2008, 2014; EvoTorusDBN, Mol. Biol. Evol., 2017

Generative models of local structure

- **TorusDBN**, Boomsma *et al.*, **PNAS**, 2008, 2014
- **EvoTorusDBN**, Golden *et al.*, **Mol. Biol. Evol.**, 2017

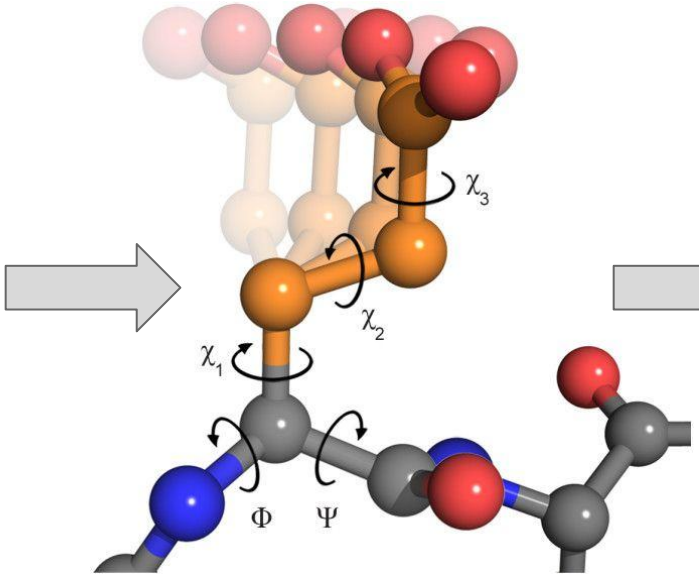


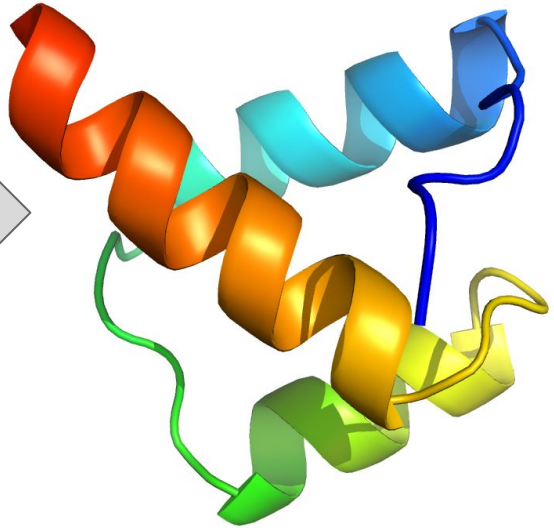
$$p(\boldsymbol{\phi}, \boldsymbol{\psi}, \mathbf{a} \mid \boldsymbol{\theta})$$



Bayesian end-to-end prediction

ANERGHPKLAAF
GVGHSAQWPNT
SVHGFDZPLQAC
VTSSHKLRTNNM
KLWERHKLNCAS
PTWKHERGHPKL





$$p(\boldsymbol{\phi}, \boldsymbol{\psi}, \mathbf{a} \mid \boldsymbol{\theta})$$

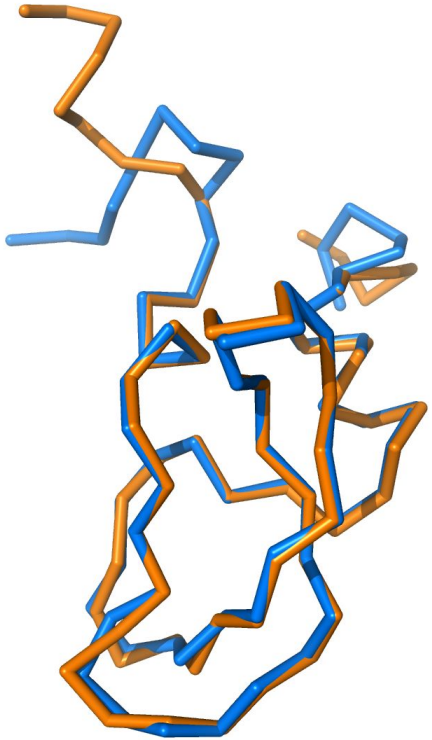
$$p(\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\psi}, \mathbf{a} \mid \boldsymbol{\theta})$$

Protein structure superposition

- Minimize the root mean square deviation (RMSD) of the atomic positions
  - Kabsch method
  - Acta Crystallographica, 1976
  - Singular value decomposition
- Problems due to unrealistic assumptions
  - Assumes atoms have **equal variance**
  - Assumes atoms are **uncorrelated**

Protein structure superposition

- Minimize the root mean square deviation (RMSD) of the atomic positions
  - Kabsch method
  - Acta Crystallographica, 1976
  - Singular value decomposition
- Problems due to unrealistic assumptions
  - Assumes atoms have **equal variance**
  - Assumes atoms are **uncorrelated**

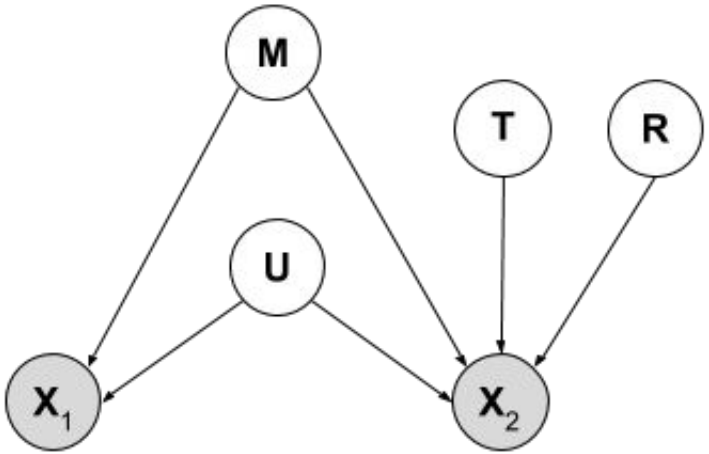


Theseus maximum likelihood model

- Douglas Theobald, PNAS, 2006
  - Based on **Mardia-Dryden model, 1989**

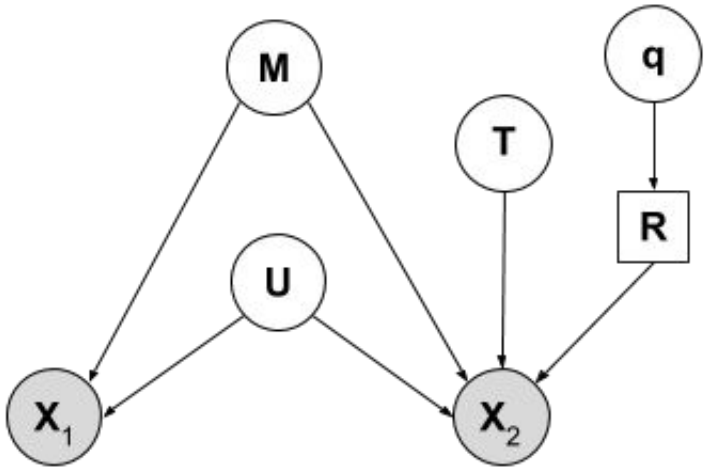
$$\mathbf{X}_{1,i} \sim \mathcal{N}_3(\mathbf{M}_i, \sigma_i \mathbf{I}_3)$$

$$\mathbf{X}_{2,i} \sim \mathcal{N}_3(\mathbf{R}\mathbf{M}_i + \mathbf{T}, \sigma_i \mathbf{I}_3)$$

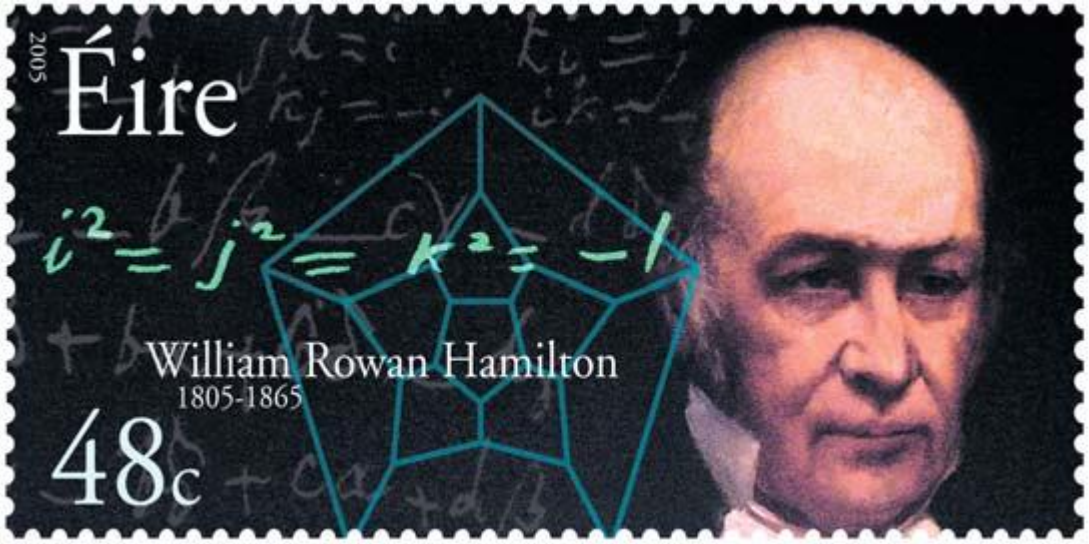


Bayesian Theseus-PPL model

- Implemented in **Pyro / PyTorch**
- Priors and likelihood
  - Student's t for M and likelihood
  - Normal prior for T
  - HalfNormal for diag(U)
  - Quaternion prior for R
- **MAP estimation**
  - Gradient descent
- **Bayesian posterior**
  - Hamiltonian Monte Carlo/NUTS
  - Variational inference



Quaternions and rotations

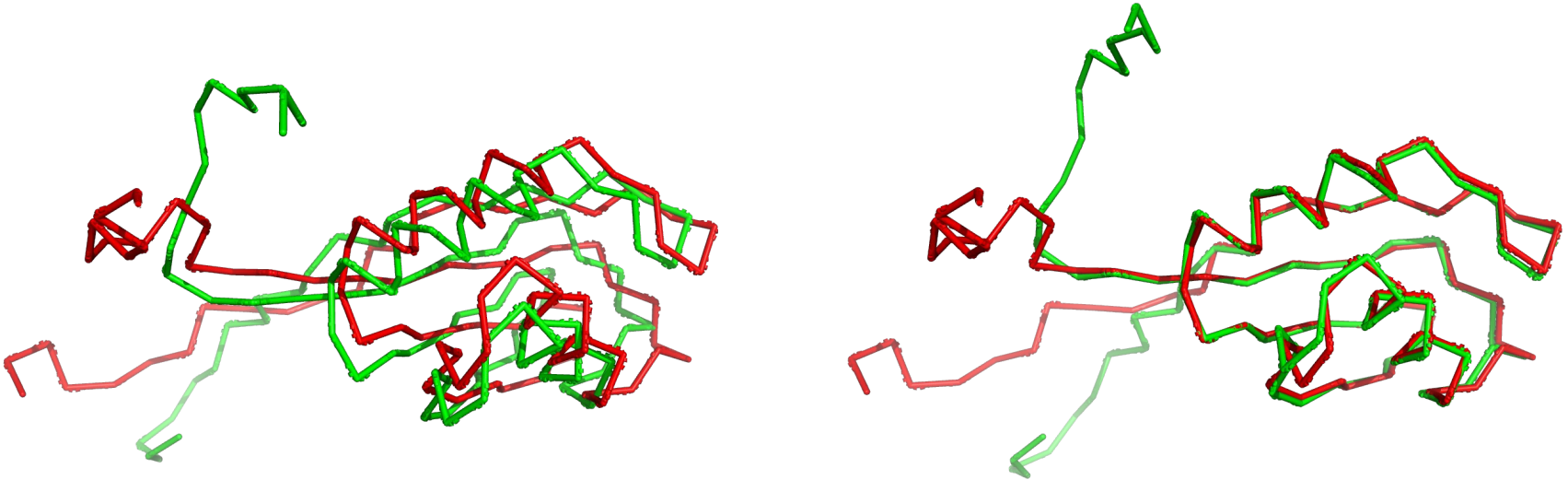


×	1	*i*	*j*	*k*
1	1	*i*	*j*	*k*
i	*i*	−1	*k*	−*j*
j	*j*	−*k*	−1	*i*
k	*k*	*j*	−*i*	−1

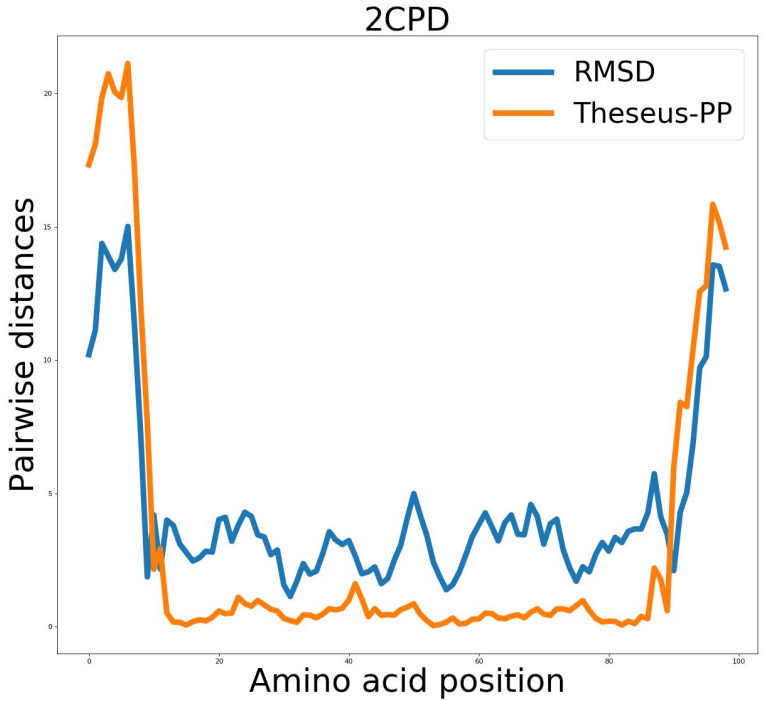
$$\mathbf{q} = w + xi + yj + zk$$

$$i^2 = j^2 = k^2 = ijk = -1$$

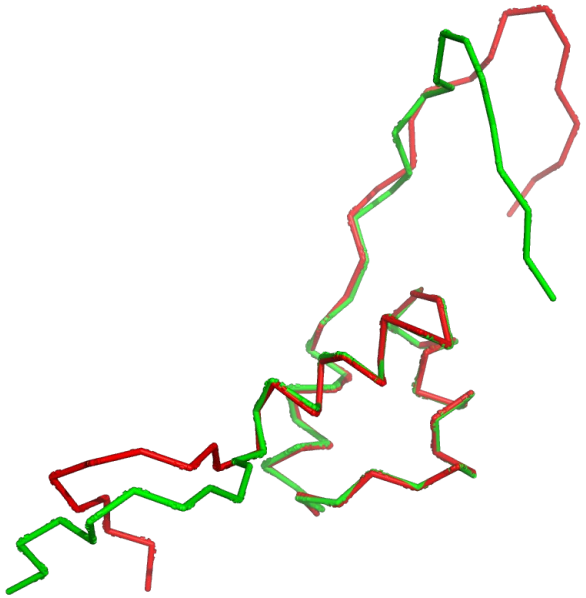
First example: 2CPD



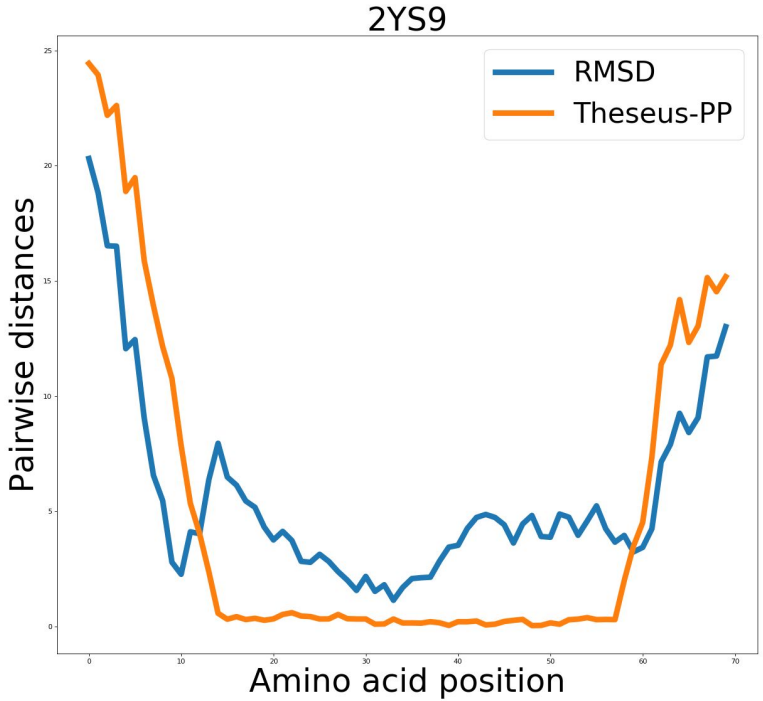
First example: 2CPD



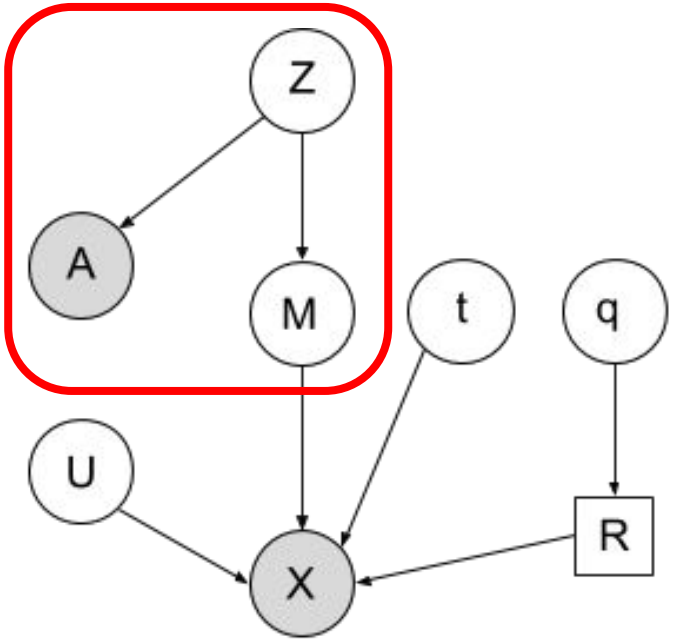
Second example: 2YS9



Second example: 2YS9



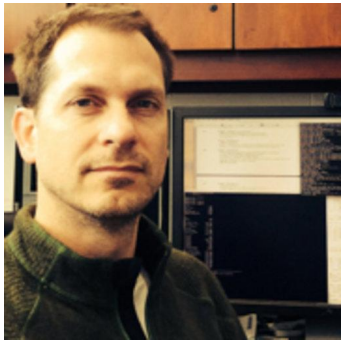
Towards a deep generative model



Conclusions

- Probabilistic Programming is an **emerging paradigm** on par with Deep Learning and Big Data Analytics
- We conjecture it will soon be possible to perform **Bayesian end-to-end protein structure prediction** using **deep probabilistic programming**
- Theseus-PPL provides a suitable **likelihood function**
  - Rotation and translation invariant
  - Robust w.r.t. prediction errors
  - Low dimensional and on the “right space”
- Protein structure prediction can serve as a **paradigm problem** for deep probabilistic programming

Kanti Mardia, Leeds



Douglas Theobald, Brandeis

Lys Sanz Moreta, DIKU

Jotun Hein, Michael Golden, Oxford

Ahmad Salim Al-Sibahi, DIKU

Fritz Henglein, DIKU

Funding & reference

“A probabilistic programming approach to protein structure superposition”, **16th IEEE-CIBCB conference**, Tuscany, Italy, July 2019