UNIVERSITÄT
HEIDELBERG

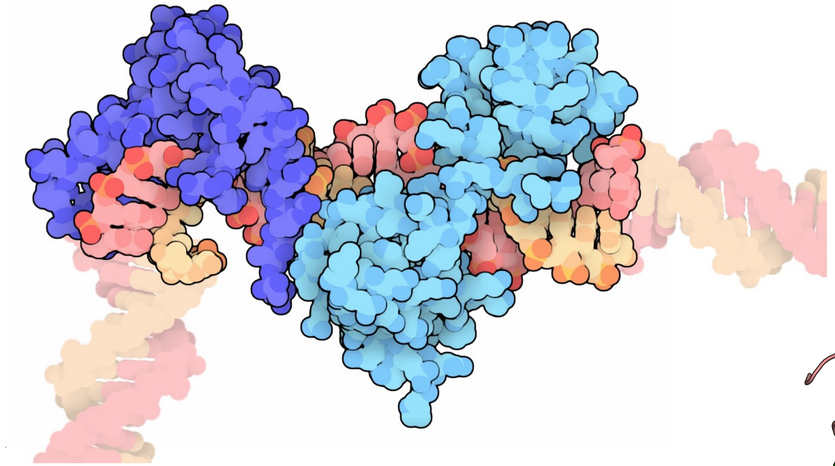
Introduction

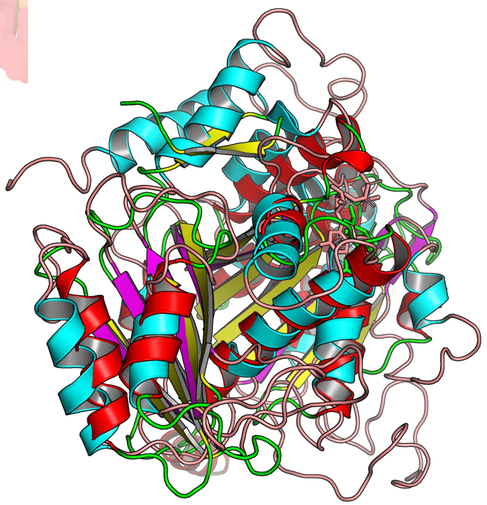
L1, Structural Bioinformatics

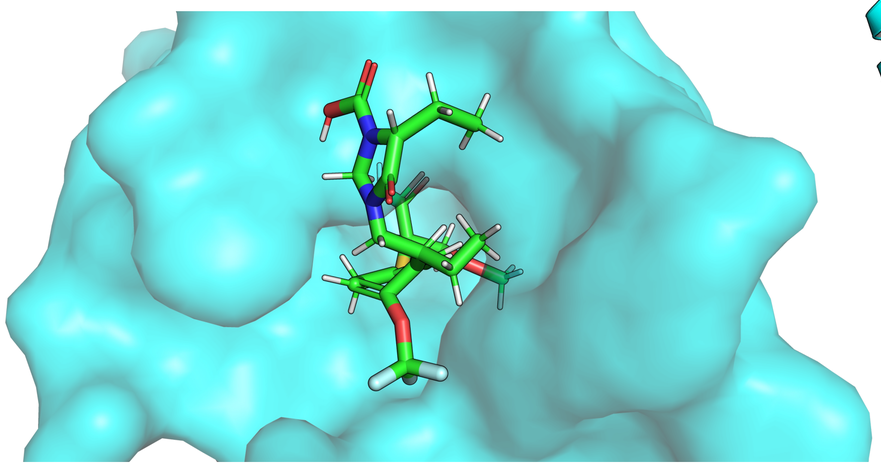
WiSe 2023/24, Heidelberg University
Kieran Didi

Structural Bioinformatics

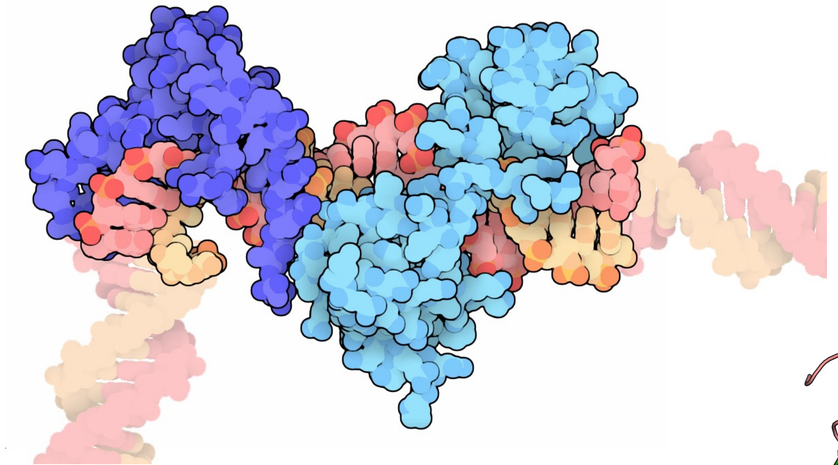
Structural Bioinformatics

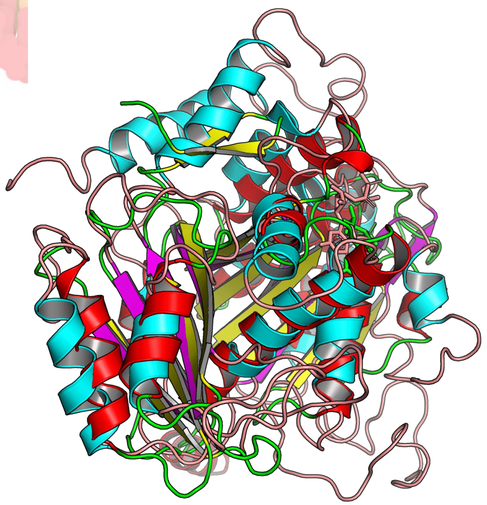


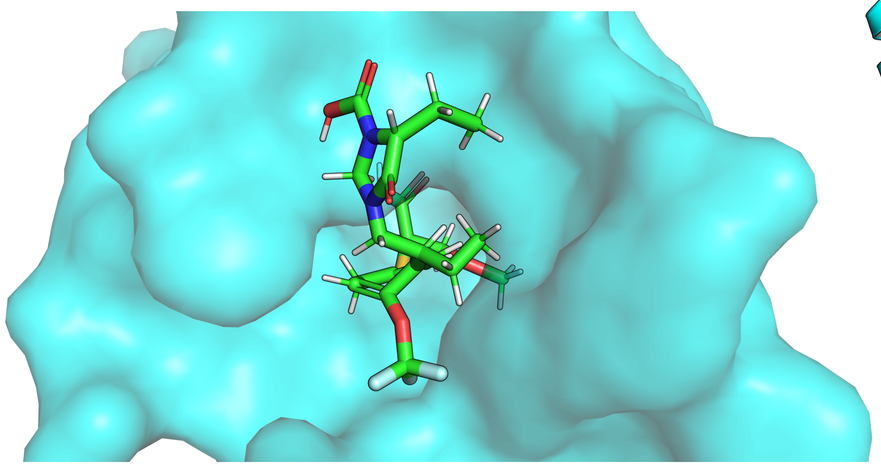


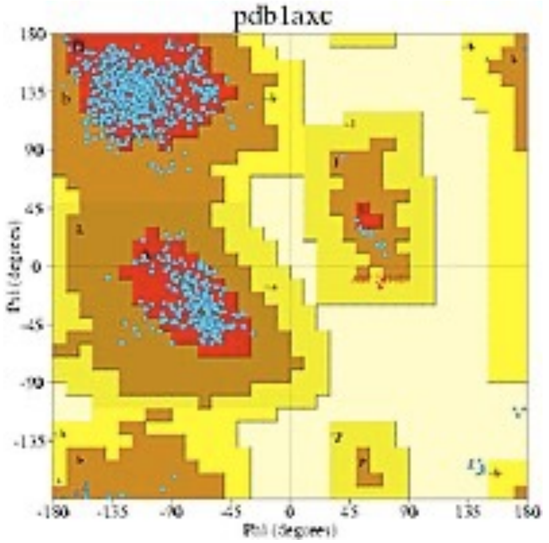


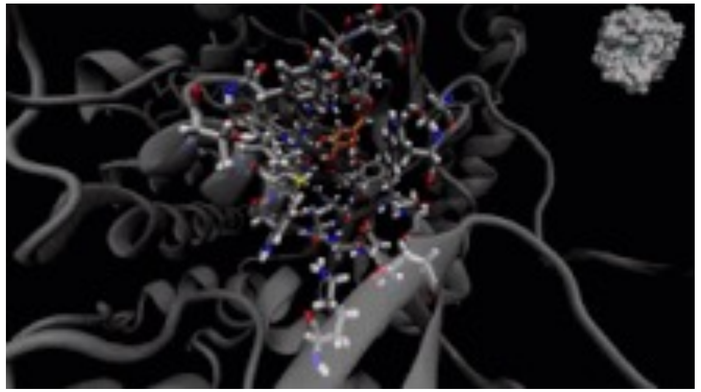
Structural Bioinformatics



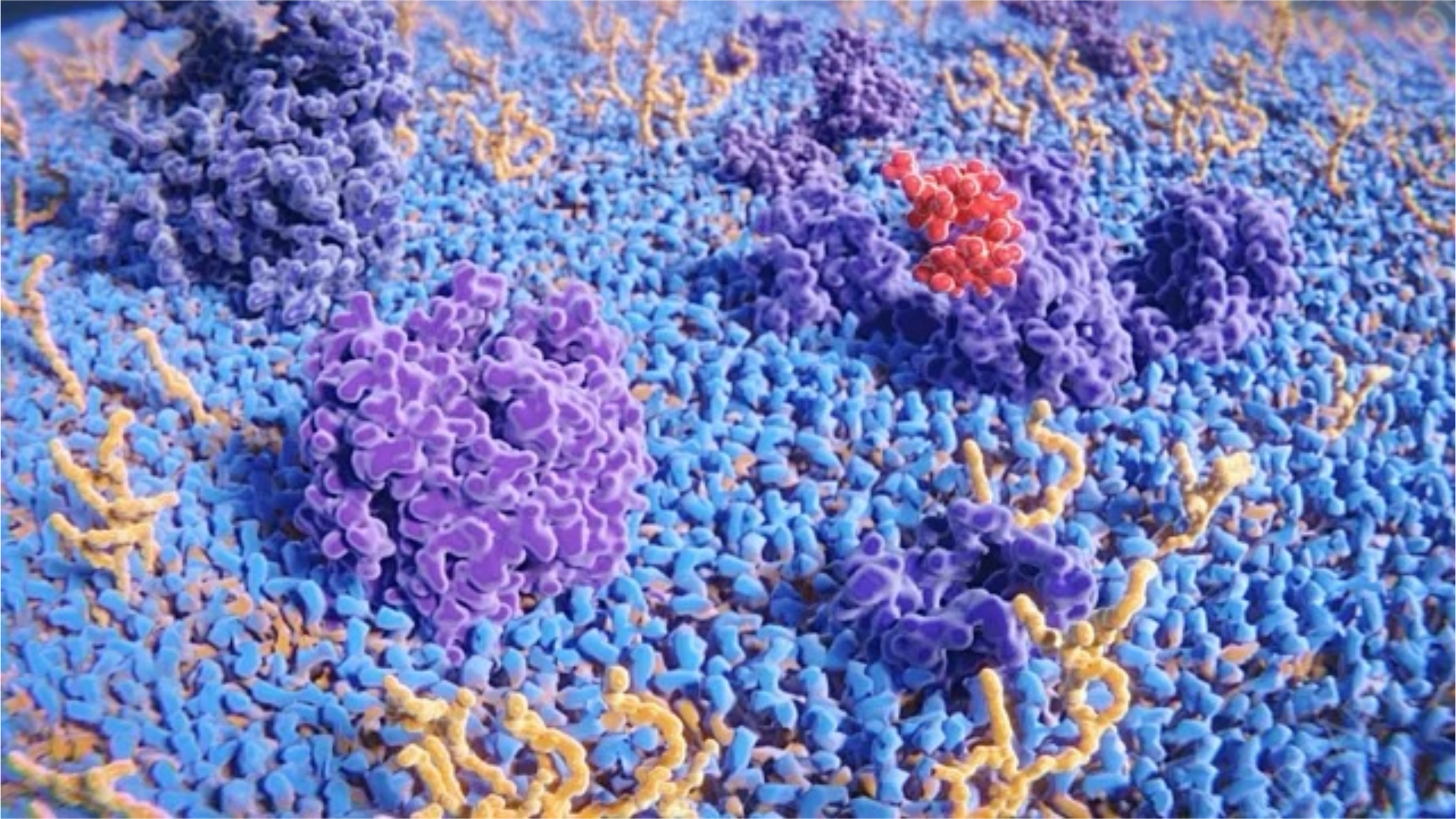








Source: Wikipedia.org

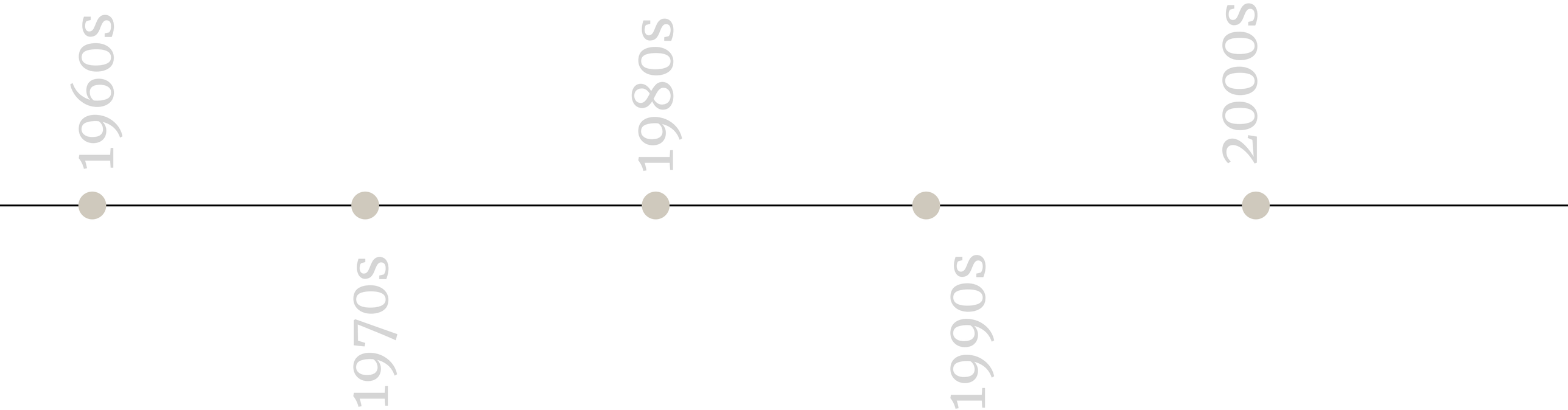


Overview

1. A Brief History of the Field
2. Where we are and where we are headed
3. This course
4. To-Dos for you!

Where do we come from?

Bioinformatics did not start structural, and not with DNA



Jeff Gauthier and others, A brief history of bioinformatics, *Briefings in Bioinformatics*, Volume 20, Issue 6, November 2019, Pages 1981–1996, https://doi.org/10.1093/bib/bby063

Where do we come from?

Bioinformatics did not start structural, and not with DNA

1960s

PROTEIN SEQUENCE
ASSEMBLY+ALIGNMENT

Edman Sequencing, Dayhoff,
Needleman-Wunsch (1970)

1970s

1980s

1990s

2000s

```
            Thr-His-Glu-Cys         [Peptide]
                    Glu-Cys-Ala-Thr [Peptide]
        Lys-Thr-His                 [Peptide]
Met-Ile-Lys                         [Peptide]
-----------------------------------------
Met-Ile-Lys-Thr-His-Glu-Cys-Ala-Thr [Protein]
```

Jeff Gauthier and others, A brief history of bioinformatics, *Briefings in Bioinformatics*, Volume 20, Issue 6, November 2019, Pages 1981–1996, https://doi.org/10.1093/bib/bby063

Where do we come from?

Bioinformatics did not start structural, and not with DNA

1960s

PROTEIN SEQUENCE ASSEMBLY+ALIGNMENT

Edman Sequencing, Dayhoff, Needleman-Wunsch (1970)

1970s

THE SHIFT TO DNA

Sanger Sequencing, Phylogenetics

```
            Thr-His-Glu-Cys          [Peptide]
                    Glu-Cys-Ala-Thr  [Peptide]
        Lys-Thr-His                  [Peptide]
Met-Ile-Lys                          [Peptide]
-----------------------------------------------
Met-Ile-Lys-Thr-His-Glu-Cys-Ala-Thr  [Protein]
```

1980s

1990s

2000s

Jeff Gauthier and others, A brief history of bioinformatics, *Briefings in Bioinformatics*, Volume 20, Issue 6, November 2019, Pages 1981–1996, https://doi.org/10.1093/bib/bby063

Where do we come from?

Bioinformatics did not start structural, and not with DNA

1960s

PROTEIN SEQUENCE ASSEMBLY+ALIGNMENT

Edman Sequencing, Dayhoff, Needleman-Wunsch (1970)

1970s

THE SHIFT TO DNA

Sanger Sequencing, Phylogenetics

```
            Thr-His-Glu-Cys          [Peptide]
                    Glu-Cys-Ala-Thr  [Peptide]
        Lys-Thr-His                  [Peptide]
Met-Ile-Lys                          [Peptide]
-----------------------------------------------
Met-Ile-Lys-Thr-His-Glu-Cys-Ala-Thr  [Protein]
```

1980s

THE PC MOVEMENT

Open-source, Journals, Perl&Python

1990s

2000s

Jeff Gauthier and others, A brief history of bioinformatics, *Briefings in Bioinformatics*, Volume 20, Issue 6, November 2019, Pages 1981–1996, https://doi.org/10.1093/bib/bby063

Where do we come from?

Bioinformatics did not start structural, and not with DNA

1960s

PROTEIN SEQUENCE ASSEMBLY+ALIGNMENT

Edman Sequencing, Dayhoff, Needleman-Wunsch (1970)

1970s

THE SHIFT TO DNA

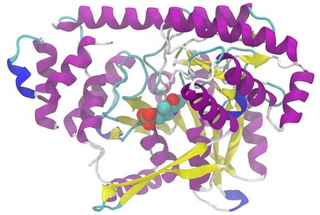
Sanger Sequencing, Phylogenetics

```
            Thr-His-Glu-Cys          [Peptide]
                    Glu-Cys-Ala-Thr  [Peptide]
        Lys-Thr-His                  [Peptide]
Met-Ile-Lys                          [Peptide]
-----------------------------------------------
Met-Ile-Lys-Thr-His-Glu-Cys-Ala-Thr  [Protein]
```

1980s

THE PC MOVEMENT

Open-source, Journals, Perl&Python



1990s

WWW, GENOMES & STRUCTURES

Human Genome, Swiss-Prot, NCBI, Webtools

2000s

Jeff Gauthier and others, A brief history of bioinformatics, *Briefings in Bioinformatics*, Volume 20, Issue 6, November 2019, Pages 1981–1996, https://doi.org/10.1093/bib/bby063

Where do we come from?

Bioinformatics did not start structural, and not with DNA

1960s

PROTEIN SEQUENCE ASSEMBLY+ALIGNMENT

Edman Sequencing, Dayhoff, Needleman-Wunsch (1970)

1970s

```
            Thr-His-Glu-Cys          [Peptide]
                    Glu-Cys-Ala-Thr  [Peptide]
        Lys-Thr-His                  [Peptide]
Met-Ile-Lys                          [Peptide]
-----------------------------------------------
Met-Ile-Lys-Thr-His-Glu-Cys-Ala-Thr  [Protein]
```

THE SHIFT TO DNA

Sanger Sequencing, Phylogenetics

1980s

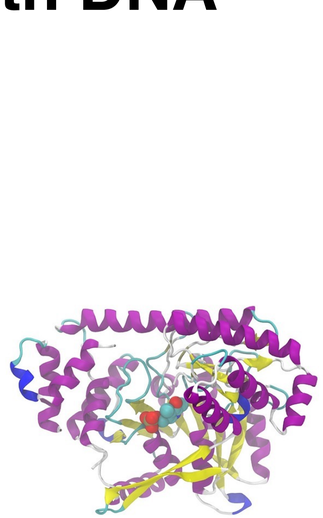
THE PC MOVEMENT

Open-source, Journals, Perl&Python

1990s

WWW, GENOMES & STRUCTURES

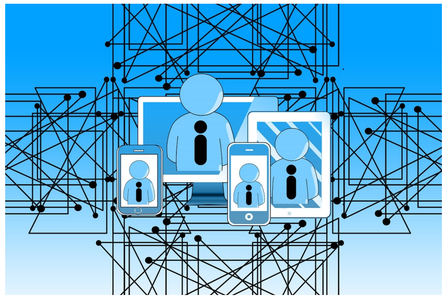
Human Genome, Swiss-Prot, NCBI, Webtools



2000s

HIGH THROUGHPUT

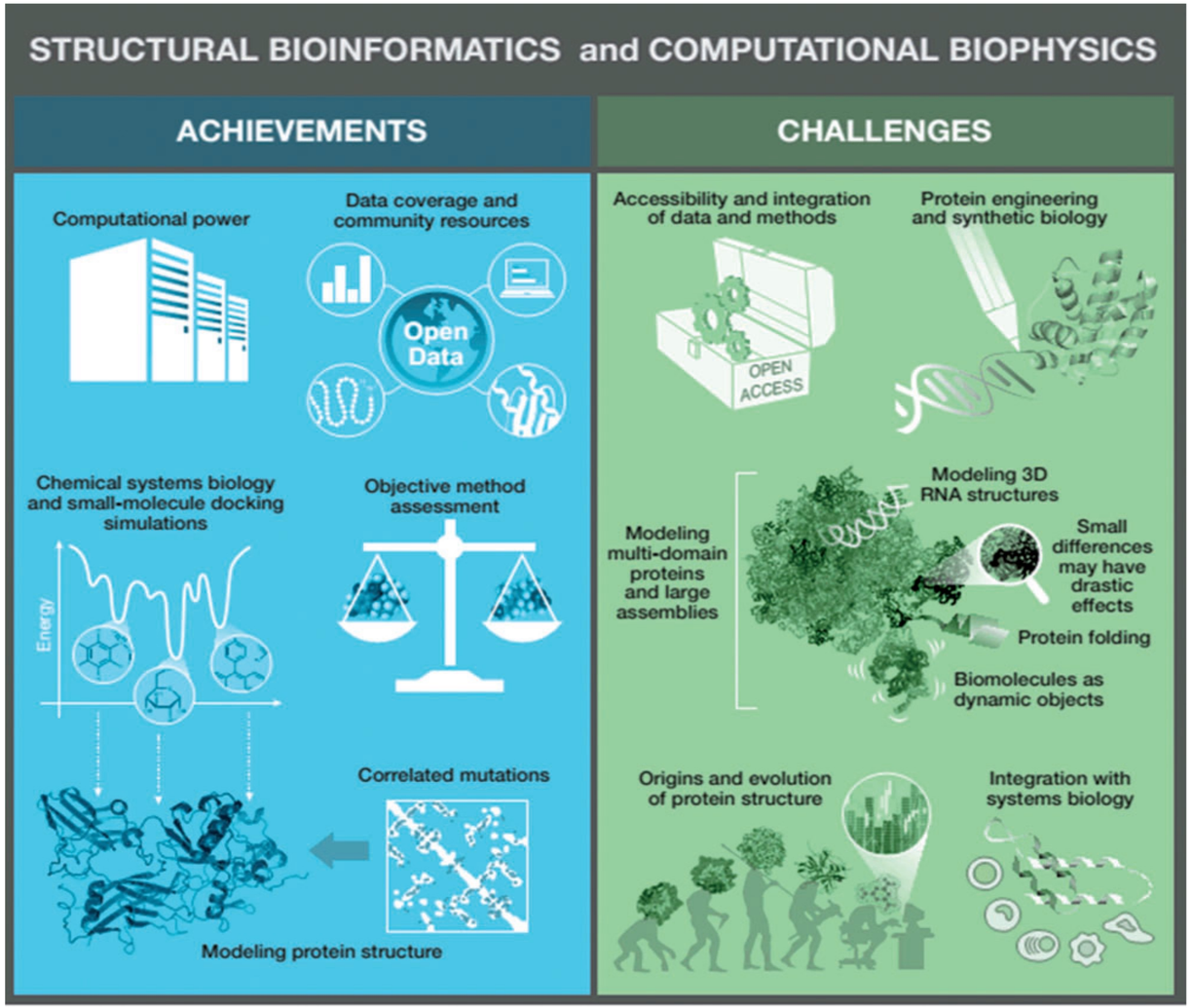
NGS, Compute Clusters, PDB 3000 -> 8000 entries



Jeff Gauthier and others, A brief history of bioinformatics, *Briefings in Bioinformatics*, Volume 20, Issue 6, November 2019, Pages 1981–1996, https://doi.org/10.1093/bib/bby063

Where are we today?

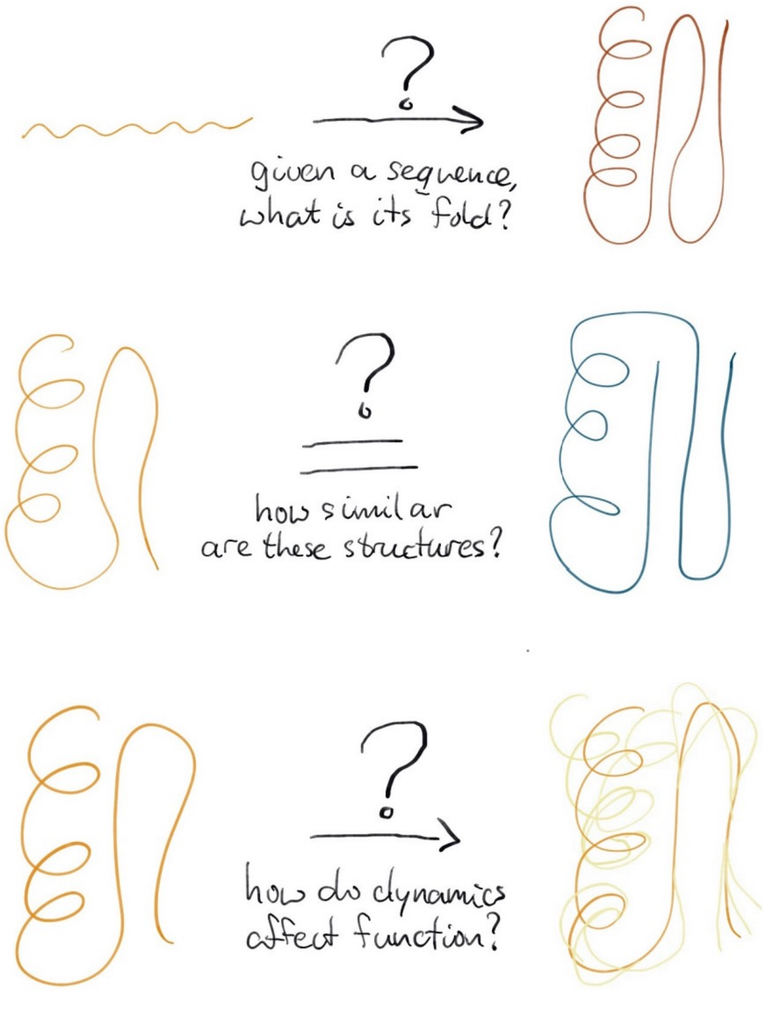
Achievements in Structural Bioinformatics (2014)



Bioinformatics, Volume 31, Issue 1, January 2015, Pages 146–150, https://doi.org/10.1093/bioinformatics/btu769.

What are the question we ask?

And how can we answer them?



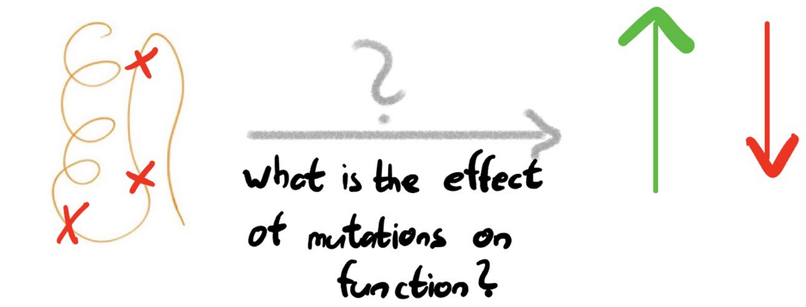
Protein Structure
Prediction
AlphaFold2

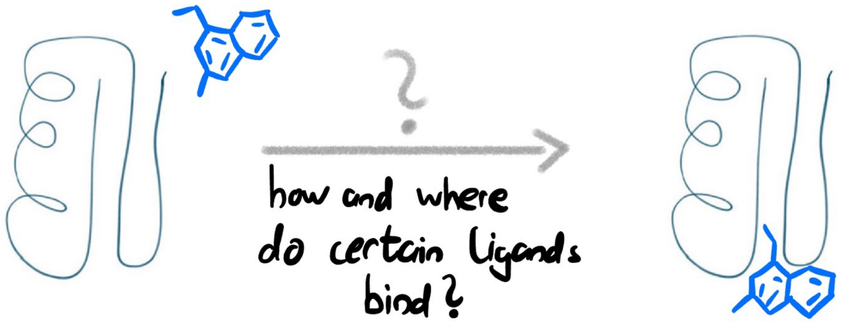
Sequence Alignments
Structure Alignments
Classification

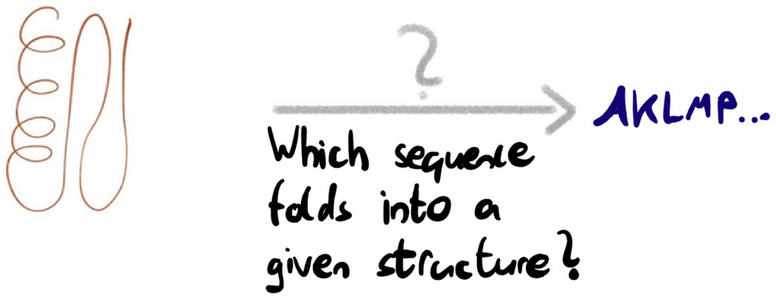
MD Simulations
Protein Design

What are the question we ask?

And how can we answer them?







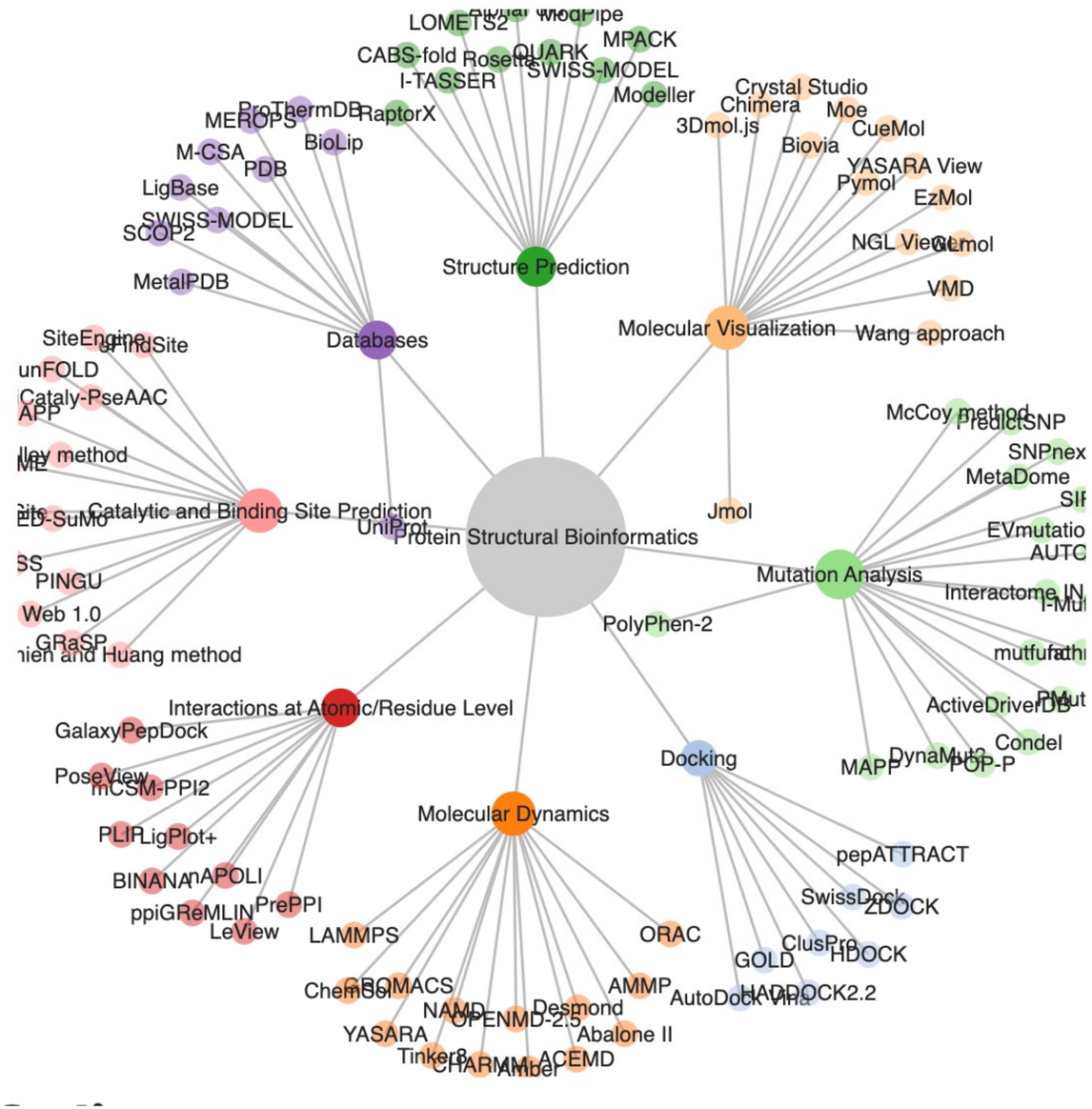
Protein Engineering

Docking
Drug Design

Inverse Folding
Protein Design

What are the tools at our disposal?

PreStO



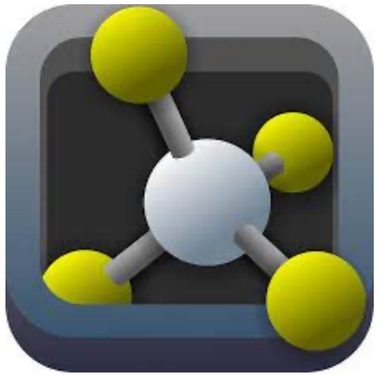
This course

What will we talk about?

1 Intro	2 ML Basics	3 ML Architectures	4 Graph ML
5 Structure Prediction	6 Generative Models	7 Protein Design	8 Simulations
9 Evolution & Bioinformatics	10 Drug Design	11 Graphics& Animations	12 Future Directions

Know your tools

Pymol – Python - Proteins



PyTorch

To-Dos for you!

1. Enter the Discord server
2. Install Pymol
3. Read the Python Post and do Google Colab Intro
4. Do the first exercises!

UNIVERSITÄT
HEIDELBERG

Math Primer 1: Linear Algebra

L1, Structural Bioinformatics

WiSe 2023/24, Heidelberg University
Kieran Didi

Overview

1. A Brief History of the Field
2. Where we are and where we are headed
3. This course
4. To-Dos for you!

Scalars

- A scalar is a single number
- Integers, real numbers, rational numbers, etc.
- We denote it with italic font:

$$a, n, x$$

Vectors

- A vector is a 1-D array of numbers:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \tag{2.1}$$

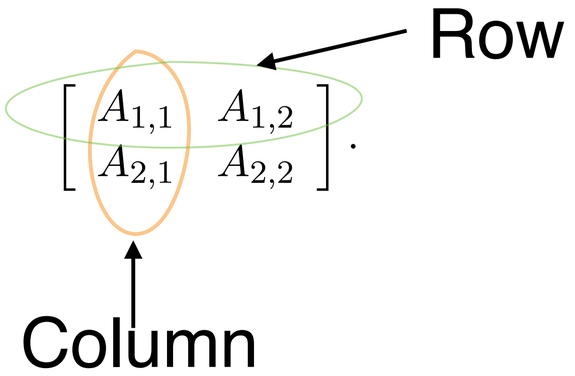
- Can be real, binary, integer, etc.

- Example notation for type and size:

$$\mathbb{R}^n$$

Matrices

- A matrix is a 2-D array of numbers:



$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}. \tag{2.2}$$

- Example notation for type and shape:

$$\boldsymbol{A} \in \mathbb{R}^{m \times n}$$

Tensors

- A tensor is an array of numbers, that may have
  - zero dimensions, and be a scalar
  - one dimension, and be a vector
  - two dimensions, and be a matrix
  - or more dimensions.

Matrix Transpose

$$(\boldsymbol{A}^\top)_{i,j} = A_{j,i}. \tag{2.3}$$

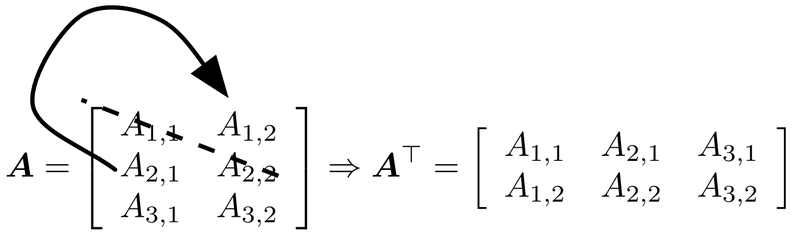


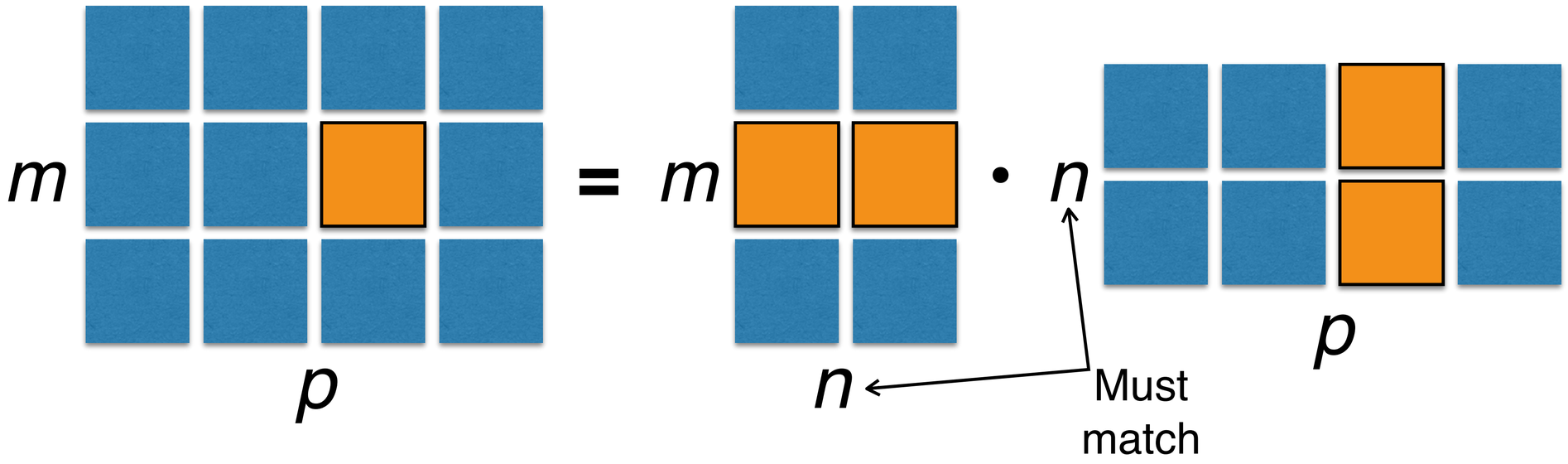
Figure 2.1: The transpose of the matrix can be thought of as a mirror image across the main diagonal.

$$(\boldsymbol{AB})^\top = \boldsymbol{B}^\top \boldsymbol{A}^\top. \tag{2.9}$$

Matrix (Dot) Product

$$\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B}. \tag{2.4}$$

$$C_{i,j} = \sum_k A_{i,k} B_{k,j}. \tag{2.5}$$



Identity Matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 2.2: *Example identity matrix*: This is $\boldsymbol{I}_3$.

$$\forall \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{I}_n \boldsymbol{x} = \boldsymbol{x}. \tag{2.20}$$

Systems of Equations

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{2.11}$$

expands to

$$\boldsymbol{A}_{1,:}\boldsymbol{x} = b_1 \tag{2.12}$$

$$\boldsymbol{A}_{2,:}\boldsymbol{x} = b_2 \tag{2.13}$$

$$\dots \tag{2.14}$$

$$\boldsymbol{A}_{m,:}\boldsymbol{x} = b_m \tag{2.15}$$

Solving Systems of Equations

- A linear system of equations can have:
  - No solution
  - Many solutions
  - Exactly one solution: this means multiplication by the matrix is an invertible function

Matrix Inversion

- Matrix inverse:

$$\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}_n. \tag{2.21}$$

- Solving a system using an inverse:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{2.22}$$

$$\boldsymbol{A}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b} \tag{2.23}$$

$$\boldsymbol{I}_n\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b} \tag{2.24}$$

- Numerically unstable, but useful for abstract analysis

Invertibility

- Matrix can't be inverted if…
  - More rows than columns
  - More columns than rows
  - Redundant rows/columns ("linearly dependent", "low rank")

Norms

- L^p norm

$$||\boldsymbol{x}||_p = \left(\sum_i |x_i|^p \right)^{\frac{1}{p}}$$

- Most popular norm: L2 norm, p=2

- L1 norm, p=1: $||\boldsymbol{x}||_1 = \sum_i |x_i|.$ (2.31)

- Max norm, infinite p: $||\boldsymbol{x}||_\infty = \max_i |x_i|.$ (2.32)

UNIVERSITÄT
HEIDELBERG

Math Primer 2: Probability

L1, Structural Bioinformatics

WiSe 2023/24, Heidelberg University
Kieran Didi

Probability Mass Function

Describing discrete event space

- The domain of P must be the set of all possible states of x.
- $\forall x \in \mathrm{x}, 0 \leq P(x) \leq 1$. An impossible event has probability 0 and no state can be less probable than that. Likewise, an event that is guaranteed to happen has probability 1, and no state can have a greater chance of occurring.
- $\sum_{x \in \mathrm{x}} P(x) = 1$. We refer to this property as being **normalized**. Without this property, we could obtain probabilities greater than one by computing the probability of one of many events occurring.

Example: uniform distribution: $P(\mathrm{x} = x_i) = \frac{1}{k}$

Probability Density Function

Describing continuous event space

- The domain of p must be the set of all possible states of x.
- $\forall x \in \mathrm{x}, p(x) \geq 0$. Note that we do not require $p(x) \leq 1$.
- $\int p(x)dx = 1$.

Example: uniform distribution: $u(x; a, b) = \frac{1}{b-a}$.

The Sum Rule of Probability

How to calculate a marginal

$$\forall x \in \mathrm{x}, P(\mathrm{x} = x) = \sum_{y} P(\mathrm{x} = x, \mathrm{y} = y). \tag{3.3}$$

$$p(x) = \int p(x, y) dy. \tag{3.4}$$

Conditional Probability

A slice through the distribution

$$P(\mathrm{y} = y \mid \mathrm{x} = x) = \frac{P(\mathrm{y} = y, \mathrm{x} = x)}{P(\mathrm{x} = x)}.$$

The Chain Rule of Probability

How to factor a joint distribution

$$P(\mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(n)}) = P(\mathrm{x}^{(1)}) \Pi_{i=2}^{n} P(\mathrm{x}^{(i)} \mid \mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(i-1)}). \tag{3.6}$$

(Conditional) Independence

When can we consider events separately?

$$\forall x \in \mathrm{x}, y \in \mathrm{y},\ p(\mathrm{x} = x, \mathrm{y} = y) = p(\mathrm{x} = x)p(\mathrm{y} = y). \tag{3.7}$$

$$\forall x \in \mathrm{x}, y \in \mathrm{y}, z \in \mathrm{z},\ p(\mathrm{x} = x, \mathrm{y} = y \mid \mathrm{z} = z) = p(\mathrm{x} = x \mid \mathrm{z} = z)p(\mathrm{y} = y \mid \mathrm{z} = z). \tag{3.8}$$

Expectation

A weighted average of all possible outcomes

$$\mathbb{E}_{\mathbf{x} \sim P}[f(x)] = \sum_x P(x) f(x), \tag{3.9}$$

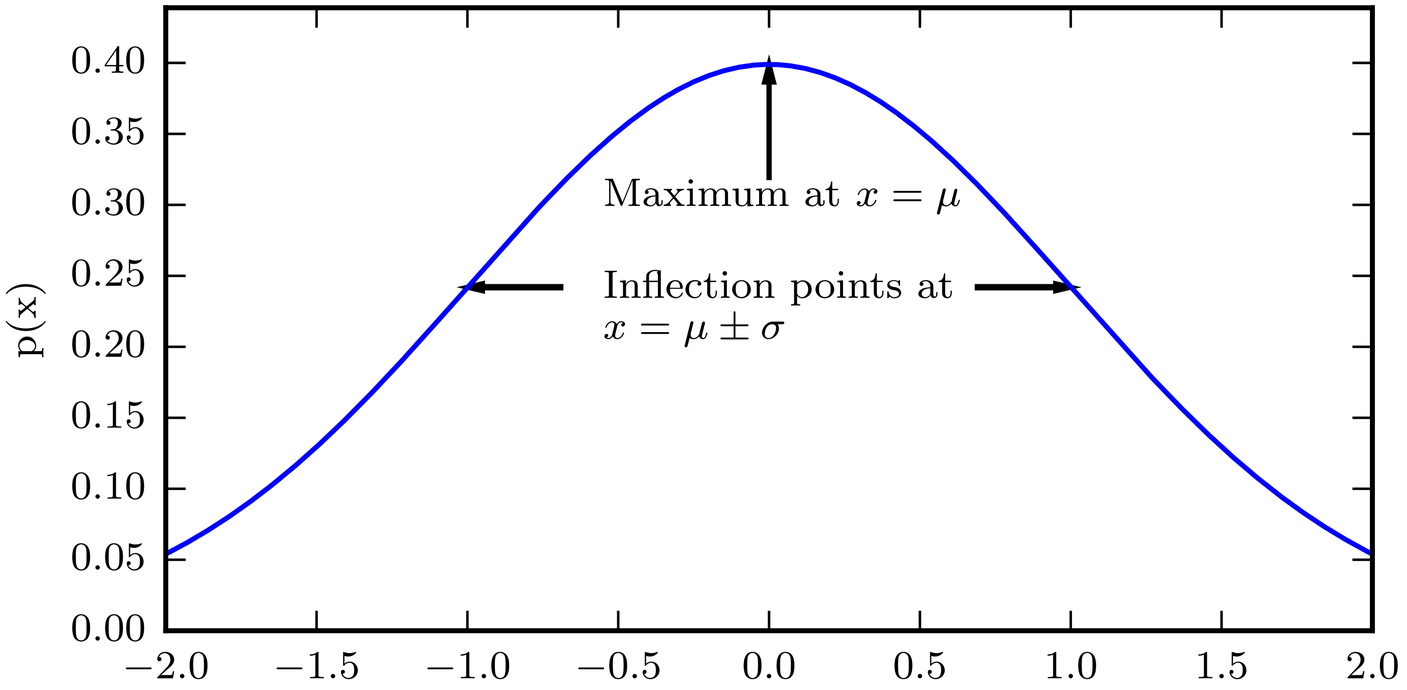
$$\mathbb{E}_{\mathbf{x} \sim p}[f(x)] = \int p(x) f(x) dx. \tag{3.10}$$

linearity of expectations:

$$\mathbb{E}_{\mathbf{x}}[\alpha f(x) + \beta g(x)] = \alpha \mathbb{E}_{\mathbf{x}}[f(x)] + \beta \mathbb{E}_{\mathbf{x}}[g(x)], \tag{3.11}$$

Gaussian Distribution

The bread-and-butter of ML



Gaussian Distribution

The bread-and-butter of ML

$$\mathcal{N}(x; \mu, \sigma^2) = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right). \tag{3.21}$$

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sqrt{\frac{1}{(2\pi)^n \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right). \tag{3.23}$$

Bayes' Rule

Incorporating prior knowledge

$$P(\mathrm{x} \mid \mathrm{y}) = \frac{P(\mathrm{x})P(\mathrm{y} \mid \mathrm{x})}{P(\mathrm{y})}. \tag{3.42}$$