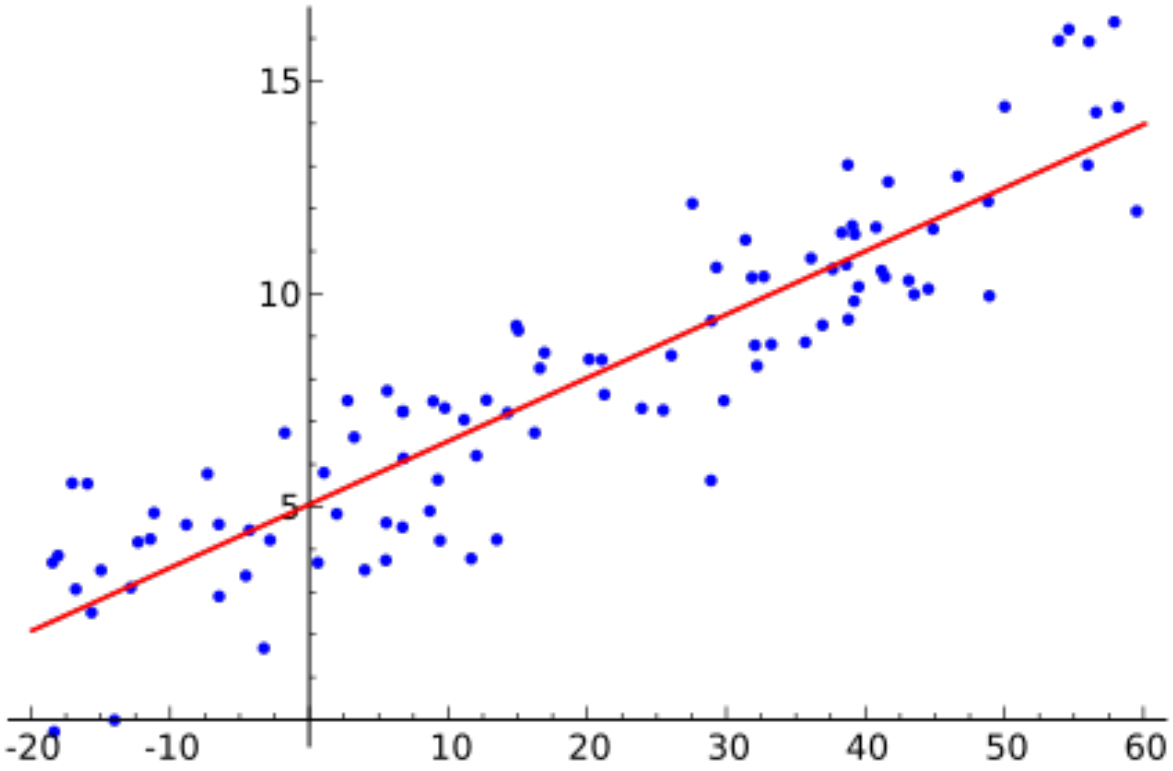
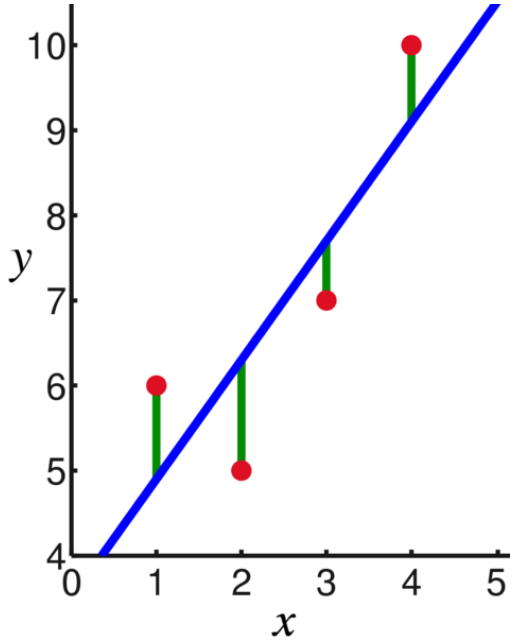
Chapter 2

Least squares problems

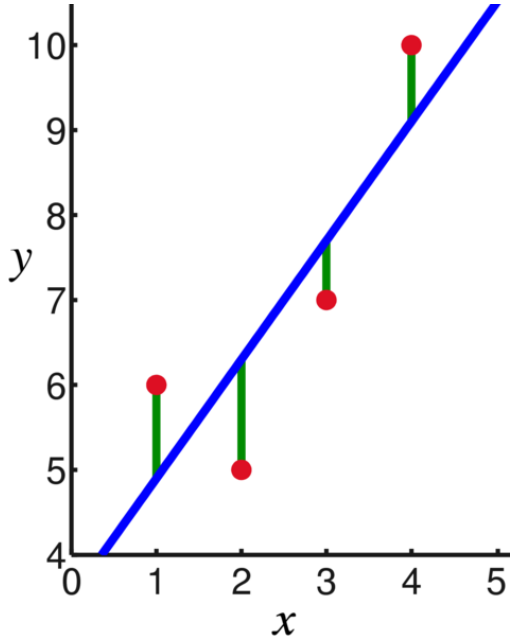


Linear curve fitting

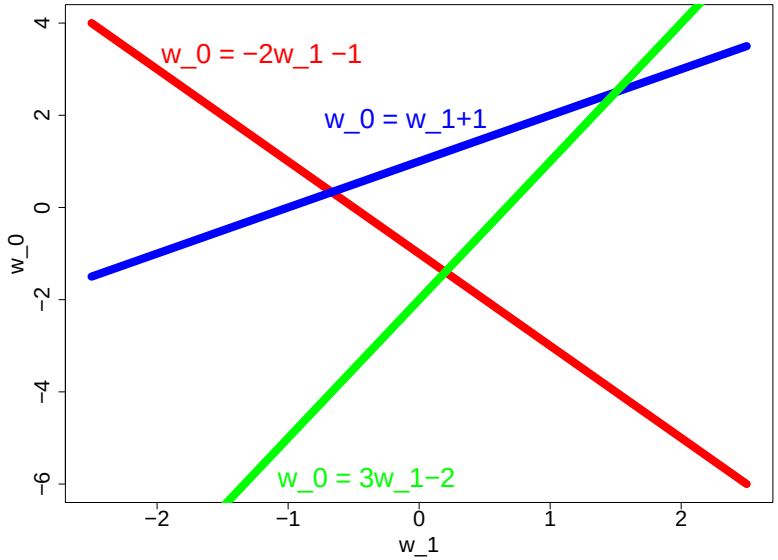


- **Notation:** n objects at locations $\boldsymbol{x}_i \in \mathbb{R}^p$.
  Every object has measurement $y_i \in \mathbb{R}$.
- **Approximate** "regression targets" y as a **parametrized function** of $\boldsymbol{x}$.
- Consider a 1-dim problem initially.
- Start with n data points $(x_i, y_i),\ i = 1, \ldots, n$.
- Choose d **basis functions** $g_0(x), g_1(x), \ldots$.
- Fitting to a **line** uses **two** basis functions
  $g_0(x) = 1$ and $g_1(x) = x$. In most cases $n \gg d$.
- **Fit function = linear combination of basis functions:**
  $f(x; \boldsymbol{w}) = \sum_j w_j g_j(x) = w_0 + w_1 x$.
- $f(x_i) = y_i$ exactly is (usually) **not possible**, so approximate $f(x_i) \approx y_i$
- n **residuals** are defined by $r_i = y_i - f(x_i) = y_i - (w_0 + w_1 x_i)$.

Calculus or algebra?



- **Quality of fit** can be measured by **residual sum of squares**
  $RSS = \sum_i r_i^2 = \sum_i [y_i - (w_0 + w_1 x_i)]^2$.
- Minimizing RSS with respect to w_1 and w_0 provides the **least-squares fit.**
- To solve the **least squares problem** we can
  1. set the derivative of RSS to zero
     $\rightsquigarrow$ **calculus**, or
  2. solve an **over-determined system**
     $\rightsquigarrow$ **algebra**: $w_0 + w_1 x_i = y_i,\ i = 1, \ldots, n$
- The results you get are...
  - **mathematically the same**, but
  - have **different numerical properties**.



Matrix-vector form

- Write $f(x) \approx y$ in matrix-vector form for n observed points as

$$\underbrace{\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} w_0 \\ w_1 \end{bmatrix}}_{\boldsymbol{w}} \approx \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{\boldsymbol{y}}$$

- We minimize the sum of squared errors, which is the squared norm of the residual vector $\boldsymbol{r} = \boldsymbol{y} - X\boldsymbol{w}$:

$$RSS = \sum_{i=1}^{n} (y_i - (X\boldsymbol{w})_i)^2 = \|\boldsymbol{y} - X\boldsymbol{w}\|^2 = \|\boldsymbol{r}\|^2 = \boldsymbol{r}^t\boldsymbol{r}.$$

- $RSS = 0$ only possible **if all the data points lie on a line**.

Basis functions

X has as many columns as there are basis functions. Examples:

- **High-dimensional linear functions**
  $\boldsymbol{x} \in \mathbb{R}^p$, $g_0(\boldsymbol{x}) = 1$ and $g_1(\boldsymbol{x}) = x_1, g_2(\boldsymbol{x}) = x_2, \ldots, g_p(\boldsymbol{x}) = x_p$.

$$X_{i\bullet} = \boldsymbol{g}^t(\boldsymbol{x}_i) = (1, — \, \boldsymbol{x}_i^t \, —), \quad (i\text{-th row of } X)$$

$$f(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{w}^t \boldsymbol{g} = w_0 + w_1 x_1 + \cdots + w_p x_p.$$

- **Document analysis:** Assume a fixed collection of words:

$$\begin{aligned} \boldsymbol{x} &= \text{ text document} \\ g_0(\boldsymbol{x}) &= 1 \\ g_i(\boldsymbol{x}) &= \#(\text{occurences of } i\text{-th word in document}) \\ f(\boldsymbol{x}; \boldsymbol{w}) &= \boldsymbol{w}^t \boldsymbol{g} = w_0 + \sum_{i \in \text{words}} w_i g_i(\boldsymbol{x}). \end{aligned}$$

Solution by Calculus

$$\begin{aligned} RSS &= \boldsymbol{r}^t\boldsymbol{r} = (\boldsymbol{y} - X\boldsymbol{w})^t(\boldsymbol{y} - X\boldsymbol{w}) \\ &= \boldsymbol{y}^t\boldsymbol{y} - \boldsymbol{y}^tX\boldsymbol{w} - \boldsymbol{w}^tX^t\boldsymbol{y} + \boldsymbol{w}^tX^tX\boldsymbol{w} \\ &= \boldsymbol{y}^t\boldsymbol{y} - 2\boldsymbol{y}^tX\boldsymbol{w} + \boldsymbol{w}^tX^tX\boldsymbol{w}. \end{aligned}$$

Minimization: set the gradient (vector of partial derivatives) to zero:

$$\nabla_{\boldsymbol{w}} RSS = \frac{\partial RSS}{\partial \boldsymbol{w}} \stackrel{!}{=} \mathbf{0}.$$

We need some properties of vector derivatives:

$$\begin{aligned} \partial(A\boldsymbol{x})/\partial\boldsymbol{x} &= A^t \\ \partial(\boldsymbol{x}^tA)/\partial\boldsymbol{x} &= A \\ \partial(\boldsymbol{x}^tA\boldsymbol{x})/\partial\boldsymbol{x} &= A\boldsymbol{x} + A^t\boldsymbol{x} \quad \text{(if } A \text{ is square)} \end{aligned}$$

Normal Equations

$$\begin{aligned}\frac{\partial RSS}{\partial \boldsymbol{w}} &= \frac{\partial}{\partial \boldsymbol{w}}\left[\boldsymbol{y}^t\boldsymbol{y} - 2\boldsymbol{y}^t X\boldsymbol{w} + \boldsymbol{w}^t X^t X\boldsymbol{w}\right] \\ &= -2X^t\boldsymbol{y} + \left[X^t X\boldsymbol{w} + (X^t X)^t\boldsymbol{w}\right] \\ &= -2X^t\boldsymbol{y} + 2X^t X\boldsymbol{w} = \boldsymbol{0}\end{aligned}$$

Normal equations: $X^t X\boldsymbol{w} = X^t\boldsymbol{y}$**.**

Could solve this system. **But:** All solution methods based on normal equations are **inherently susceptible to roundoff errors:**

$$\begin{aligned}k(X) &= \sigma_{\max}/\sigma_{\min}, \text{ where } X^t X\boldsymbol{v}_i = \sigma_i^2\boldsymbol{v}_i \\ k(X^t X) &= \mu_{\max}/\mu_{\min}, \text{ where } X^t X X^t X\boldsymbol{v}_i = \mu_i^2\boldsymbol{v}_i \\ X^t X X^t X\boldsymbol{v}_i &= X^t X\sigma_i^2\boldsymbol{v}_i = \sigma_i^4\boldsymbol{v}_i \;\Rightarrow\; \mu_i = \sigma_i^2 \\ \Rightarrow\; k(X^t X) &= k^2(X),\end{aligned}$$

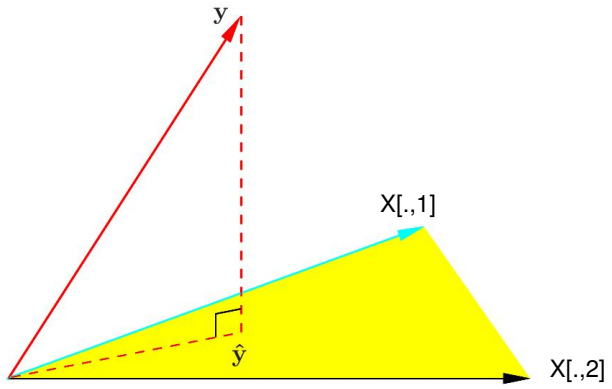
The algebraic approach will avoid this problem!

From Calculus to Algebra

$$\frac{\partial RSS(\boldsymbol{w})}{\partial \boldsymbol{w}} = -2X^t\boldsymbol{y} + 2X^tX\boldsymbol{w} \stackrel{!}{=} \mathbf{0}$$

$$\Rightarrow X^t(\boldsymbol{y} - X\hat{\boldsymbol{w}}) = X^t\boldsymbol{r} = \mathbf{0} \quad \Rightarrow \boldsymbol{r} \in N(X^t).$$

- Every $X\boldsymbol{w}$ is in column space $C(X)$,
  residual $\boldsymbol{r}$ is in the orthogonal complement $N(X^t)$ (left nullspace).
- Let $\hat{\boldsymbol{y}}$ be the **orthogonal projection** of $\boldsymbol{y}$ on $C(X)$
  $\rightsquigarrow$ $\boldsymbol{y}$ can be split into $\hat{\boldsymbol{y}} \in C(X) + \boldsymbol{r} \in N(X^t)$.



Adapted from Fig. 3.2 in (Hastie, Tibshirani, Friedman)

Algebraic interpretation

- $\boldsymbol{y} = \hat{\boldsymbol{y}} \in C(X) + \boldsymbol{r} \in N(X^t) \rightsquigarrow$ Consider over-determined systems

$$X\boldsymbol{w} = \boldsymbol{y} = \hat{\boldsymbol{y}} + \boldsymbol{r} \text{ (solution impossible, if } \boldsymbol{r} \neq \boldsymbol{0})$$

$$X\hat{\boldsymbol{w}} = \hat{\boldsymbol{y}} \text{ (solvable, since } \hat{\boldsymbol{y}} \in C(X)!)$$

- **The solution $\hat{w}$ of $Xw = \hat{y}$ makes the error as small as possible:**

$$\|X\boldsymbol{w} - \boldsymbol{y}\|^2 = \|X\boldsymbol{w} - (\hat{\boldsymbol{y}} + \boldsymbol{r})\|^2 = \|X\boldsymbol{w} - \hat{\boldsymbol{y}}\|^2 + \|\boldsymbol{r}\|^2$$

Reduce $\|X\boldsymbol{w} - \hat{\boldsymbol{y}}\|^2$ to zero by solving $X\hat{\boldsymbol{w}} = \hat{\boldsymbol{y}}$ and choosing $\boldsymbol{w} = \hat{\boldsymbol{w}}$.
Remaining error $\|\boldsymbol{r}\|^2$ cannot be avoided, since $\boldsymbol{r} \in N(X^t)$.

$$X^tX\hat{\boldsymbol{w}} = X^t\hat{\boldsymbol{y}} = X^t\boldsymbol{y} \quad \Rightarrow \quad \hat{\boldsymbol{w}} = (X^tX)^{-1}X^t\boldsymbol{y} \text{ (if } X^tX \text{ invertible).}$$

- The fitted values at the sample points are $\hat{\boldsymbol{y}} = X\hat{\boldsymbol{w}} = X(X^tX)^{-1}X^t\boldsymbol{y}$.
- $H = X(X^tX)^{-1}X^t$ is called **hat matrix** (puts a "hat" on $\boldsymbol{y} \rightsquigarrow \hat{\boldsymbol{y}}$).

Algebraic interpretation

- Left nullspace $N(X^t)$ is orthogonal complement of column space $C(X)$.
- H is **orthogonal projection** on $C(X)$:

$$HX = X(X^tX)^{-1}X^tX = X, \quad HN(X^t) = 0.$$

- $M = I - H$ is **orthogonal projection** on **nullspace** of X^t:

$$MX = (I - H)X = X - X = 0, \quad MN(X^t) = M.$$

- H and M are symmetric ($H^t = H$) and idempotent ($MM = M$)

The algebra of Least Squares:
H creates fitted values: $\hat{\boldsymbol{y}} = H\boldsymbol{y} \rightsquigarrow \hat{\boldsymbol{y}} \in C(X)$
M creates residuals: $\boldsymbol{r} = M\boldsymbol{y} \rightsquigarrow \boldsymbol{r} \in N(X^t)$

Algebraic interpretation

X^tX is invertible iff X has linearly independent columns.

Why? X^tX has the same nullspace as X:
(i) If $\boldsymbol{a} \in N(X)$, then $X\boldsymbol{a} = \mathbf{0} \Rightarrow X^tX\boldsymbol{a} = \mathbf{0} \rightsquigarrow \boldsymbol{a} \in N(X^tX)$.
(ii) If $\boldsymbol{a} \in N(X^tX)$, then $\boldsymbol{a}^tX^tX\boldsymbol{a} = 0 \Leftrightarrow \|X\boldsymbol{a}\|^2 = 0$,
so $X\boldsymbol{a}$ has length zero $\Rightarrow X\boldsymbol{a} = \mathbf{0}$.
Thus, every vector in one nullspace is also in the other one.
So if $N(X) = \{\mathbf{0}\}$, then $X^tX \in \mathbb{R}^{d\times d}$ has full rank d.

When X has independent columns, X^tX is positive definite.

Why? X^tX is clearly symmetric and invertible.
To show: All eigenvalues > 0
$X^tX\boldsymbol{v} = \lambda\boldsymbol{v} \rightsquigarrow \boldsymbol{v}^tX^tX\boldsymbol{v} = \lambda\boldsymbol{v}^t\boldsymbol{v} \rightsquigarrow \lambda = \frac{\|X\boldsymbol{v}\|^2}{\|\boldsymbol{v}\|^2} > 0.$

SVD for Least-Squares

- Goal: Avoid numerical problems for normal equations:
  $X^tX\boldsymbol{w} = X^t\boldsymbol{y}, \quad k(X^tX) = k^2(X)$.

- Idea: Apply the **SVD** directly to $X_{n\times d}$.

- The **squared norm of the residual** is

$$\begin{aligned} RSS = \|\boldsymbol{r}\|^2 &= \|X\boldsymbol{w} - \boldsymbol{y}\|^2 \\ &= \|USV^t\boldsymbol{w} - \boldsymbol{y}\|^2 \\ &= \|U(SV^t\boldsymbol{w} - U^t\boldsymbol{y})\|^2 \\ &= \|SV^t\boldsymbol{w} - U^t\boldsymbol{y}\|^2 \end{aligned}$$

  Last equation: U is orthonormal $\rightsquigarrow \|U\boldsymbol{a}\|^2 = \boldsymbol{a}^tU^tU\boldsymbol{a} = \boldsymbol{a}^t\boldsymbol{a} = \|\boldsymbol{a}\|^2$.

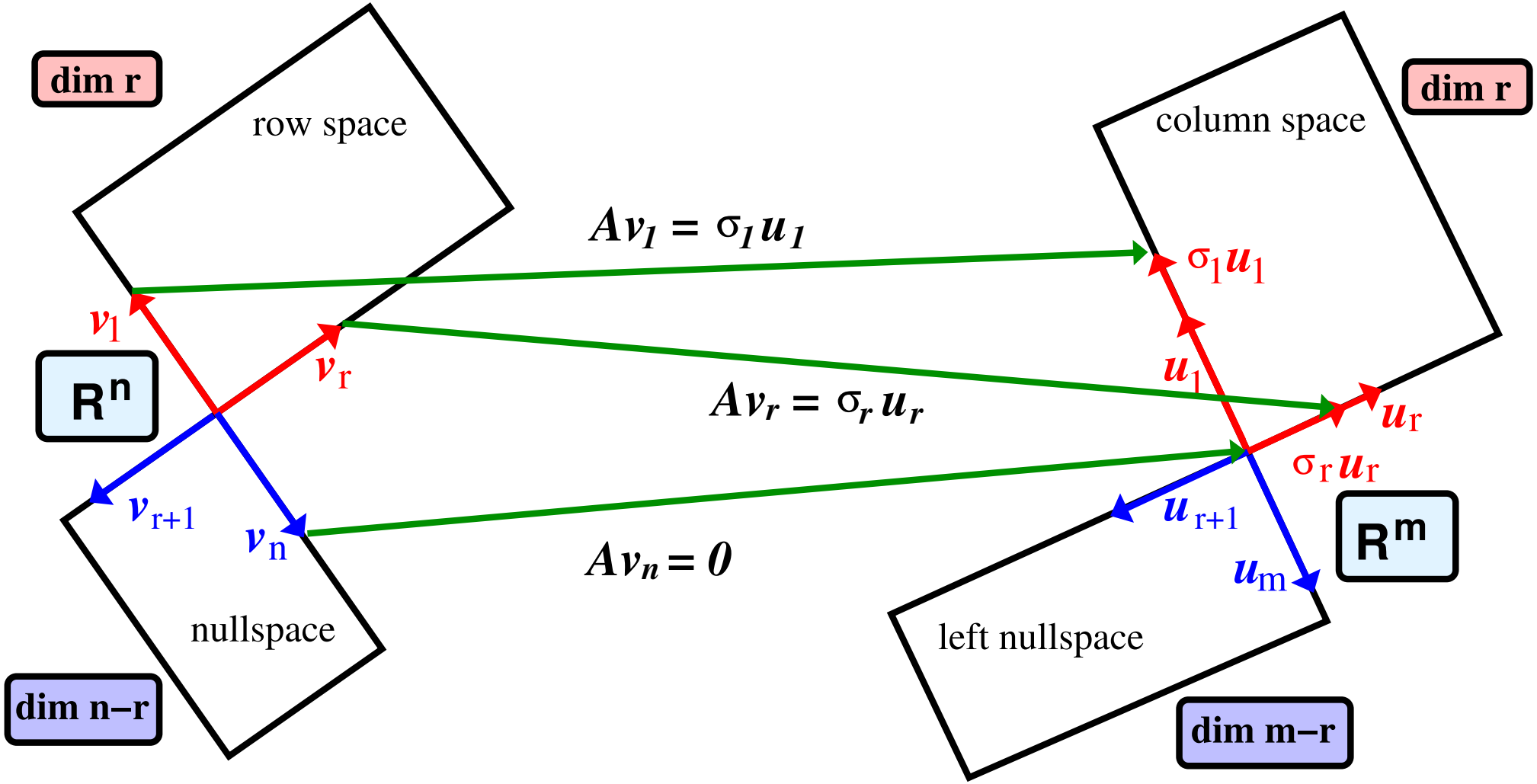
- Minimizing RSS is **equivalent** to minimizing
  $\|S\boldsymbol{z} - \boldsymbol{c}\|^2$ where $\boldsymbol{z} = V^t\boldsymbol{w}$ and $\boldsymbol{c} = U^t\boldsymbol{y}$.

SVD and LS

Recall: Columns $\boldsymbol{u}_i$ of $U_{n\times n}$ with $\sigma_i > 0$ form a **basis of** $C(X)$.
Remaining columns form **basis of** $N(X^t)$**:**

$$\boldsymbol{c} = U^t\boldsymbol{y} = \underbrace{\begin{bmatrix} - & \boldsymbol{u}_1^t & - \\ - & \boldsymbol{u}_2^t & - \\ & \vdots & \\ - & \boldsymbol{u}_d^t & - \\ 0 & 0 & 0 \\ & \vdots & \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix}}_{\begin{bmatrix} c_1 \\ \vdots \\ c_d \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in C(X)} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ & \vdots & \\ 0 & 0 & 0 \\ - & \boldsymbol{u}_{d+1}^t & - \\ - & \boldsymbol{u}_{d+2}^t & - \\ & \vdots & \\ - & \boldsymbol{u}_n^t & - \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix}}_{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ c_{d+1} \\ \vdots \\ c_n \end{bmatrix} \in N(X^t)}$$

SVD and bases for the 4 subspaces



SVD and LS

- $\|\boldsymbol{r}\|^2 = \|S\boldsymbol{z} - \boldsymbol{c}\|^2$ written in blocks:

$$\left\| \left[\begin{array}{cccc} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ 0 & 0 & \dots & \sigma_d \\ \hline 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 \end{array}\right] \left[\begin{array}{c} z_1 \\ z_2 \\ \vdots \\ z_d \end{array}\right] - \left[\begin{array}{c} c_1 \\ \vdots \\ c_d \\ c_{d+1} \\ \vdots \\ c_n \end{array}\right] \right\|^2$$

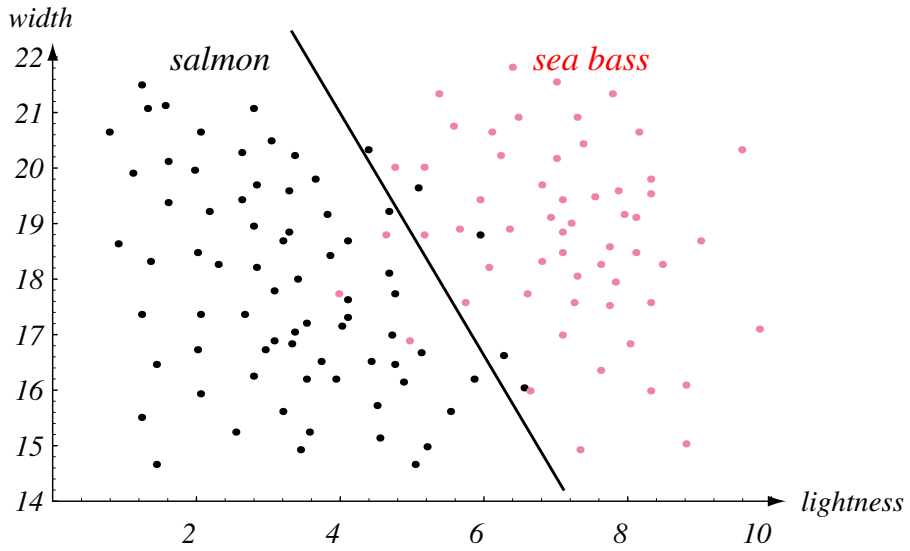
- To choose $\boldsymbol{z}$ so that $\|\boldsymbol{r}\|^2$ is minimal requires $z_i = c_i/\sigma_i, i = 1, \dots, d$
  $\rightsquigarrow r_1 = r_2 = \cdots = r_d = 0.$

- Unavoidable error: $RSS = \|\boldsymbol{r}\|^2 = c_{d+1}^2 + c_{d+2}^2 + \cdots + c_n^2.$

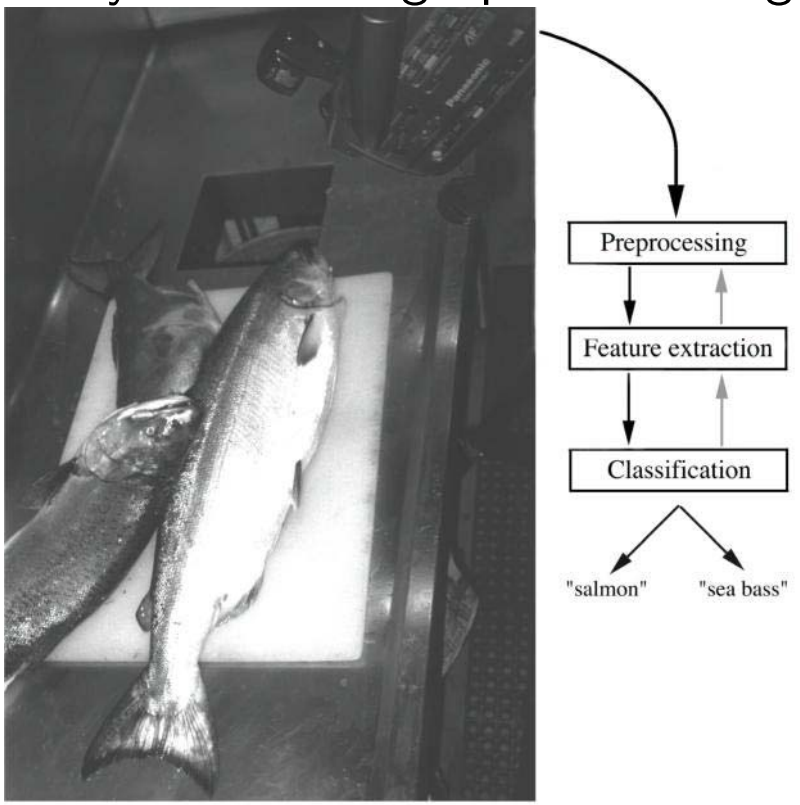
- For very small singular values, use zeroing. RSS will increase:
  One additional term (usually small): $RSS' = c_d^2 + c_{d+1}^2 + c_{d+2}^2 + \cdots + c_n^2,$
  but often significantly better precision (reduced condition number).

Classification

Classification: Find **class boundaries** based on training data $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$. Use boundaries to classify new items $\boldsymbol{x}^*$.
Here, y_i is a discrete class indicator (or "label"). Example: Fish-packing plant wants to automate the process of sorting fish on conveyor belt using optical sensing.

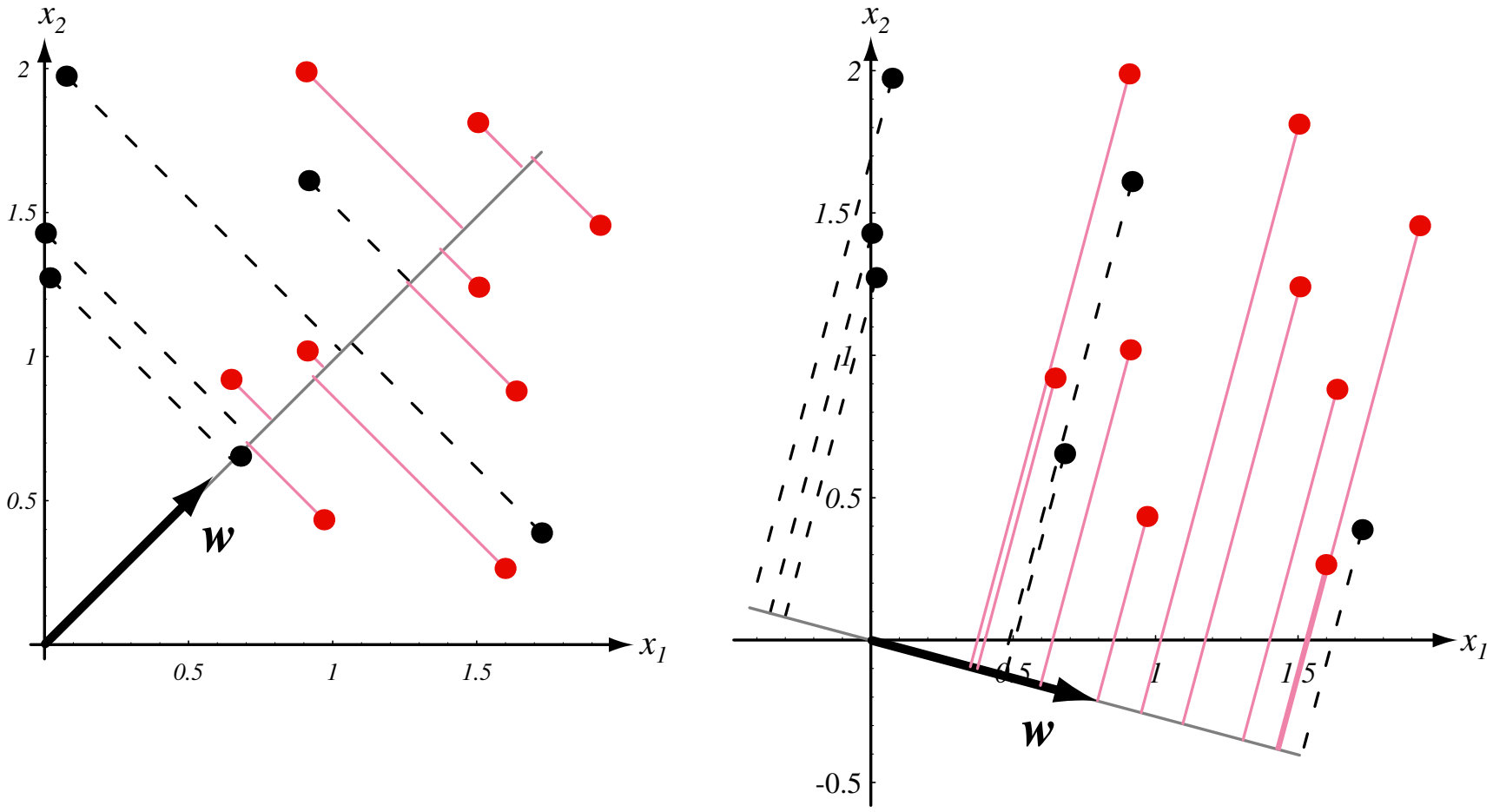


(Duda, Hart, Stork, 2001)



(Duda, Hart, Stork, 2001)

Linear Discriminant Analysis (Ronald Fisher, 1936)



(Duda, Hart, Stork, 2001)

Main Idea: Simplify the problem by projecting down to a 1-dim subspace.
Question: How should we select the **projection vector**, which optimally discriminates between the different classes?

Separation Criterion

- Let $\boldsymbol{m}_j$ an estimate of the class means $\boldsymbol{\mu}_j$:

$$\boldsymbol{m}_y = \frac{1}{n_y} \sum\nolimits_{\boldsymbol{x} \in \text{class } y} \boldsymbol{x}, \quad n_y = \#(\text{objects in class } y).$$

- Projected samples: $\boldsymbol{x}'_i = \boldsymbol{w}^t \boldsymbol{x}_i,\ i = 1, 2, \ldots, n$. Projected means:

$$\tilde{m}_y = \frac{1}{n_y} \sum\nolimits_{\boldsymbol{x} \in \text{class } y} \boldsymbol{w}^t \boldsymbol{x} = \boldsymbol{w}^t \boldsymbol{m}_y.$$

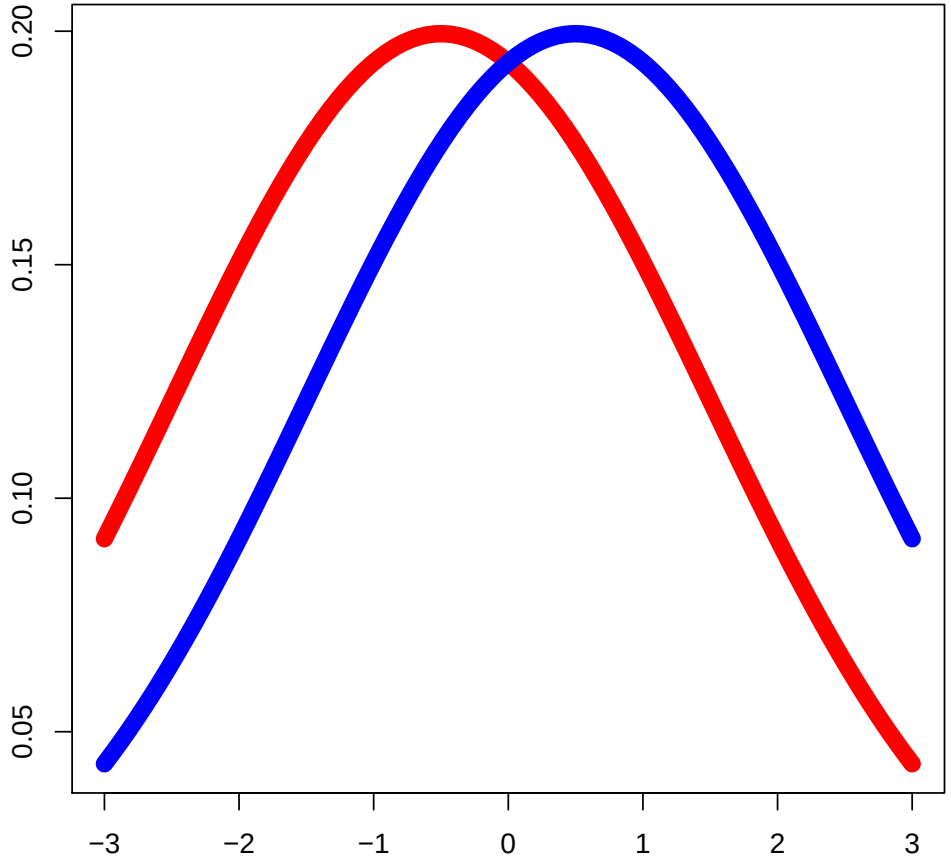
- First part of separation criterion (two-class case):

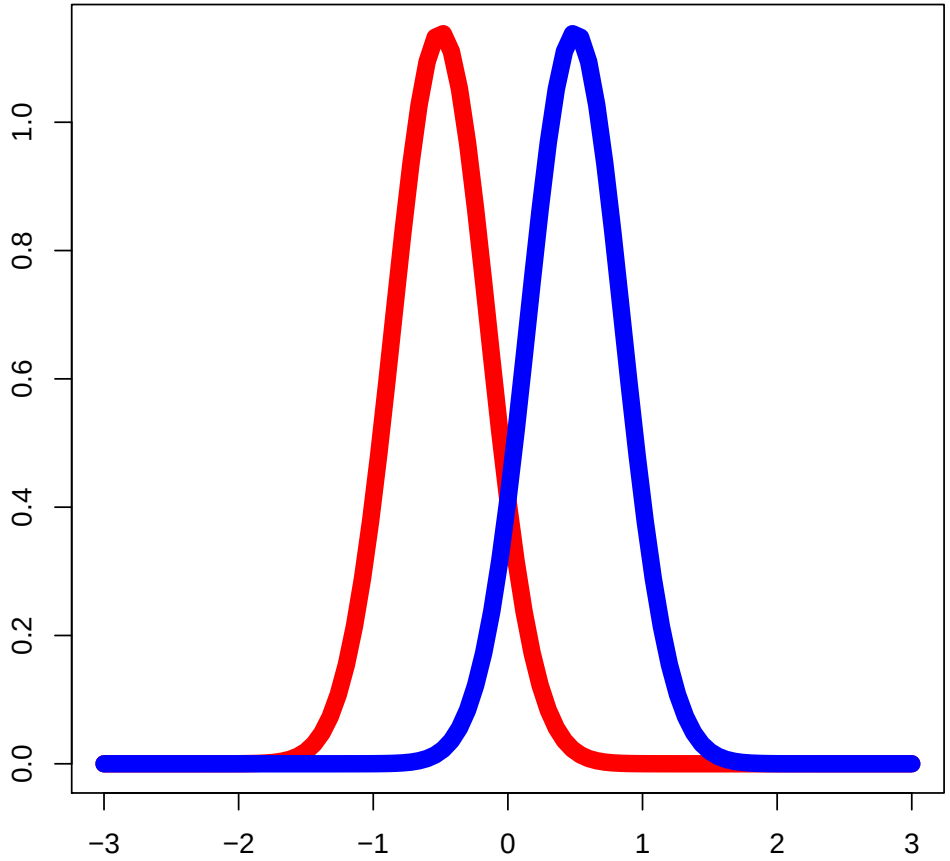
$$\max_{\boldsymbol{w}} [\boldsymbol{w}^t (\boldsymbol{m}_1 - \boldsymbol{m}_2)]^2 = \max_{\boldsymbol{w}} [\tilde{m}_1 - \tilde{m}_2]^2.$$

- There might still be considerable overlap...
  $\rightsquigarrow$ should also consider the **scatter** or **variance.**

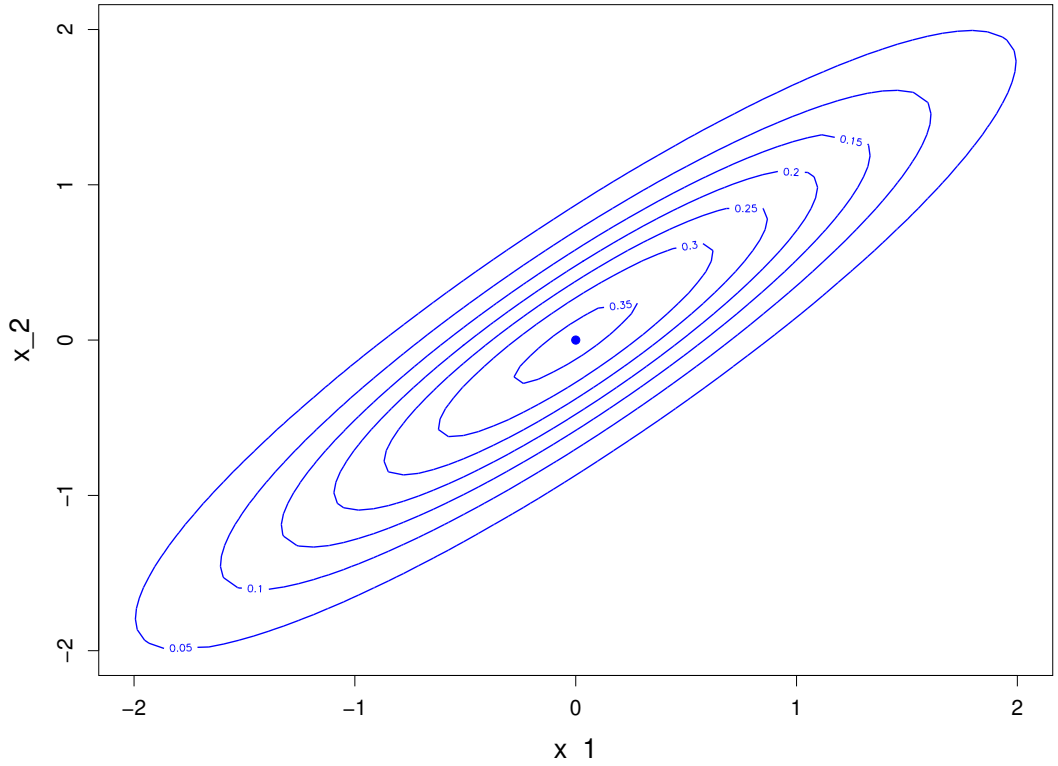
Separation Criterion

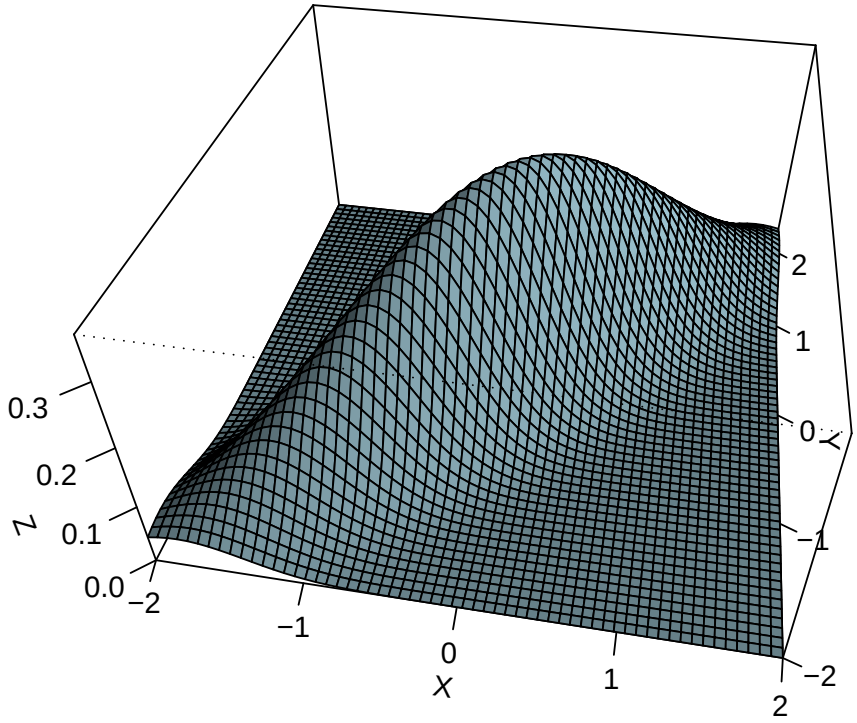
Two Gaussians with the same mean distance, but different variances:





Excursion: The multivariate Gaussian distribution



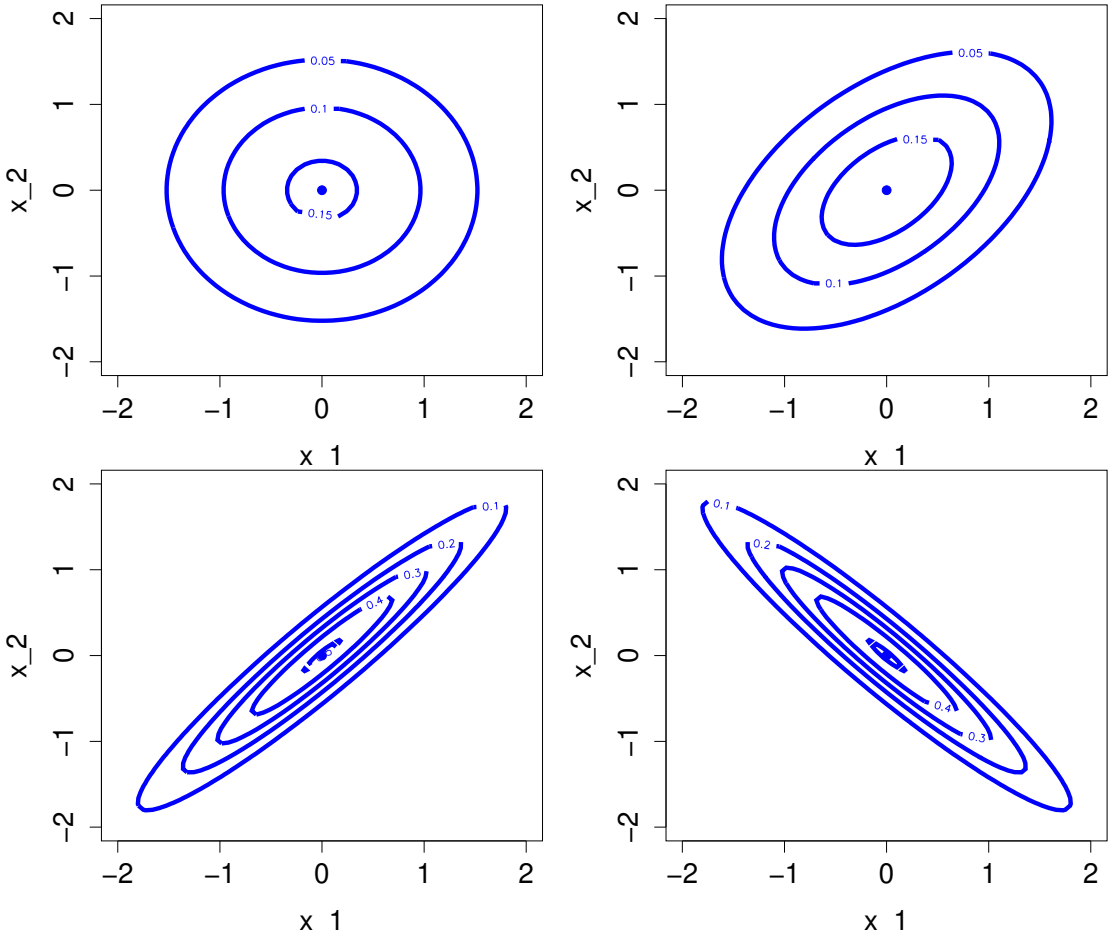


Probability density function:

$$p(\boldsymbol{x};\boldsymbol{\mu},\Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}}\exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^t\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}))$$

Excursion: The multivariate Gaussian distribution

Covariance
(also written "*co*-variance")
is a measure of how much **two random variables vary together.** Can be positive, zero, or negative.



Sample covariance matrix $\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^t$,
with sample mean $\overline{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i = \boldsymbol{m}$. If $\boldsymbol{m} = \mathbf{0} \rightsquigarrow \widehat{\boldsymbol{\Sigma}} = \frac{1}{n}X^tX$.

Separation Criterion

- Assume both classes are Gaussians with the same covariance matrix. Let Σ_W be an estimate of this **"within class" covariance matrix**:

$$\begin{aligned}\Sigma_y &= \frac{1}{n_y}\sum_{\boldsymbol{x}\in\text{class } y}(\boldsymbol{x}-\boldsymbol{m}_y)(\boldsymbol{x}-\boldsymbol{m}_y)^t,\\ \Sigma_W &= 0.5(\Sigma_1+\Sigma_2).\end{aligned}$$

- Variance of projected data:

$$\begin{aligned}\tilde{\Sigma}_y &= \frac{1}{n_y}\sum\nolimits_{\boldsymbol{x}\in\text{class } y}(\boldsymbol{w}^t\boldsymbol{x}-\tilde{m}_y)(\boldsymbol{w}^t\boldsymbol{x}-\tilde{m}_y)^t\\ &= \frac{1}{n_y}\sum\nolimits_{\boldsymbol{x}\in\text{class } y}\boldsymbol{w}^t(\boldsymbol{x}-\boldsymbol{m}_y)(\boldsymbol{x}-\boldsymbol{m}_y)^t\boldsymbol{w}=\boldsymbol{w}^t\Sigma_y\boldsymbol{w}\\ \tilde{\Sigma}_W &= 0.5(\tilde{\Sigma}_1+\tilde{\Sigma}_2)=\boldsymbol{w}^t\Sigma_W\boldsymbol{w}\in\mathbb{R}_+\end{aligned}$$

- Strategy: $\Delta^2_{\tilde{m}}=(\tilde{m}_1-\tilde{m}_2)^2$ should be large, $\tilde{\Sigma}_W$ small.

Separation Criterion

$$J(\boldsymbol{w}) \;=\; \frac{\Delta^2_{\tilde{m}}}{\tilde{\Sigma}_W} \;=\; \frac{\boldsymbol{w}^t \overbrace{(\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^t}^{=:\Sigma_B} \boldsymbol{w}}{\boldsymbol{w}^t \Sigma_W \boldsymbol{w}}.$$

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{w}} J(\boldsymbol{w}) \;&=\; \frac{\partial}{\partial \boldsymbol{w}} \frac{\boldsymbol{w}^t \Sigma_B \boldsymbol{w}}{\boldsymbol{w}^t \Sigma_W \boldsymbol{w}} \stackrel{!}{=} 0 \\
&=\; -\frac{\boldsymbol{w}^t \Sigma_B \boldsymbol{w}}{(\boldsymbol{w}^t \Sigma_W \boldsymbol{w})^2} 2\Sigma_W \boldsymbol{w} + \frac{1}{\boldsymbol{w}^t \Sigma_W \boldsymbol{w}} 2\Sigma_B \boldsymbol{w} \\
&\Rightarrow\; \frac{\boldsymbol{w}^t \Sigma_B \boldsymbol{w}}{\boldsymbol{w}^t \Sigma_W \boldsymbol{w}} (-\Sigma_W \boldsymbol{w}) + \Sigma_B \boldsymbol{w} = 0 \\
\Rightarrow\; \Sigma_B \boldsymbol{w} \;&=\; \frac{\boldsymbol{w}^t \Sigma_B \boldsymbol{w}}{\boldsymbol{w}^t \Sigma_W \boldsymbol{w}} \Sigma_W \boldsymbol{w} =: \lambda\, \Sigma_W \boldsymbol{w}
\end{aligned}$$

Separation Criterion

- Let Σ_W be non-singular:

$$\big[\Sigma_W^{-1} \underbrace{\Sigma_B\big]\boldsymbol{w}}_{\Delta_{\boldsymbol{m}}\Delta_{\boldsymbol{m}}^t\boldsymbol{w}\propto\Delta_{\boldsymbol{m}}} = \lambda\boldsymbol{w}, \quad \text{with} \quad \lambda = \frac{\boldsymbol{w}^t\Sigma_B\boldsymbol{w}}{\boldsymbol{w}^t\Sigma_W\boldsymbol{w}} = J(\boldsymbol{w}).$$

- Thus, $\boldsymbol{w}$ is an eigenvector of $\Sigma_W^{-1}\Sigma_B$, the associated eigenvalue is the objective function! **Maximum: eigenvector with largest eigenvalue.**
- Unscaled Solution: $\boxed{\hat{\boldsymbol{w}} = \Sigma_W^{-1}\Delta_{\boldsymbol{m}} = \Sigma_W^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2).}$
- This is the solution of the linear system $\Sigma_W\boldsymbol{w} = \boldsymbol{m}_1 - \boldsymbol{m}_2$.
- Σ_W is a covariance matrix $\rightsquigarrow$ there is an underlying data matrix A such that $\Sigma_W \propto A^tA \rightsquigarrow$ potential numerical problems: squared condition number compared to A...

Discriminant analysis and least squares

Theorem: The LDA vector $\hat{\boldsymbol{w}}^{\text{LDA}} = \Sigma_W^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2)$ coincides with the solution of the LS problem $\hat{\boldsymbol{w}}^{\text{LS}} = \arg\min_{\boldsymbol{w}} \|X\boldsymbol{w} - \boldsymbol{y}\|^2$ if

$$
\begin{aligned}
n_1 &= \#\text{ samples in class } \mathbf{1} \\
n_2 &= \#\text{ samples in class } \mathbf{2} \\
X &= \begin{bmatrix} - & \boldsymbol{x}_1^t & - \\ - & \boldsymbol{x}_2^t & - \\ & \vdots & \\ - & \boldsymbol{x}_n^t & - \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix},
\end{aligned}
$$

with $\dfrac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i = \boldsymbol{m} = \mathbf{0}$ (i.e. origin in sample mean),

$$
y_i = \begin{cases} +1/n_1, & \text{if } \boldsymbol{x}_i \text{ in class } \mathbf{1} \\ -1/n_2, & \text{else.} \end{cases} \qquad \Rightarrow \sum_{i=1}^{n} y_i = 0.
$$

Discriminant analysis and least squares (cont'd)

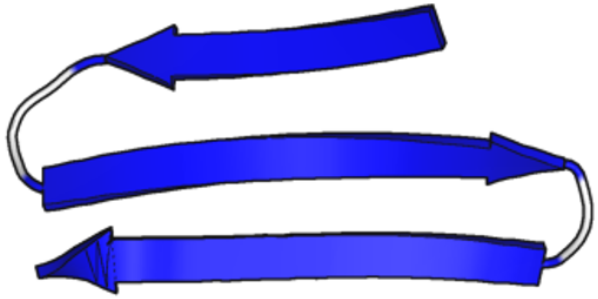
- "Within" covariance $\Sigma_W \propto \sum_{\boldsymbol{x} \in \text{class } y} (\boldsymbol{x} - \boldsymbol{m}_y)(\boldsymbol{x} - \boldsymbol{m}_y)^t$.
- "Between" covariance $\Sigma_B \propto (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^t$
- The sum of both is the "total covariance" $\Sigma_B + \Sigma_W = \Sigma_T$
  $\Sigma_T \propto \sum_i x_i \boldsymbol{x}_i^t = X^t X$.
- We know that $\boldsymbol{w}^{\text{LDA}} \propto \Sigma_W^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2) \rightsquigarrow \Sigma_W \boldsymbol{w}^{\textbf{LDA}} \propto (\boldsymbol{m}_1 - \boldsymbol{m}_2)$**.**
- Now $\Sigma_B \boldsymbol{w}^{\text{LDA}} = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^t \boldsymbol{w}^{\text{LDA}} \rightsquigarrow \Sigma_B \boldsymbol{w}^{\textbf{LDA}} \propto (\boldsymbol{m}_1 - \boldsymbol{m}_2)$**.**
- $\Sigma_T \boldsymbol{w}^{\text{LDA}} = (\Sigma_B + \Sigma_W)\boldsymbol{w}^{\text{LDA}} \rightsquigarrow \Sigma_T \boldsymbol{w}^{\textbf{LDA}} \propto (\boldsymbol{m}_1 - \boldsymbol{m}_2)$**.**
- With $X^t X = \Sigma_T$, $X^t \boldsymbol{y} = \boldsymbol{m}_1 - \boldsymbol{m}_2$, we arrive at
  $\boldsymbol{w}^{\text{LDA}} \propto \Sigma_T^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2) = \Sigma_T^{-1} X^t \boldsymbol{y} \propto (X^t X)^{-1} X^t \boldsymbol{y} = \boldsymbol{w}^{\text{LS}}$.

Chapter 2

Least squares problems

Application Example: Secondary Structure Prediction in Proteins

Secondary



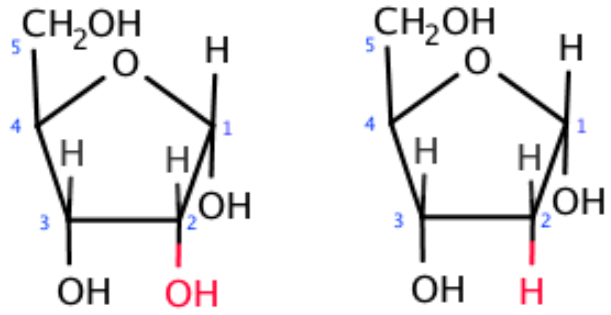
β-Sheet (3 strands)

α-helix

By Thomas Shafee, https://commons.wikimedia.org/w/index.php?curid=52821069

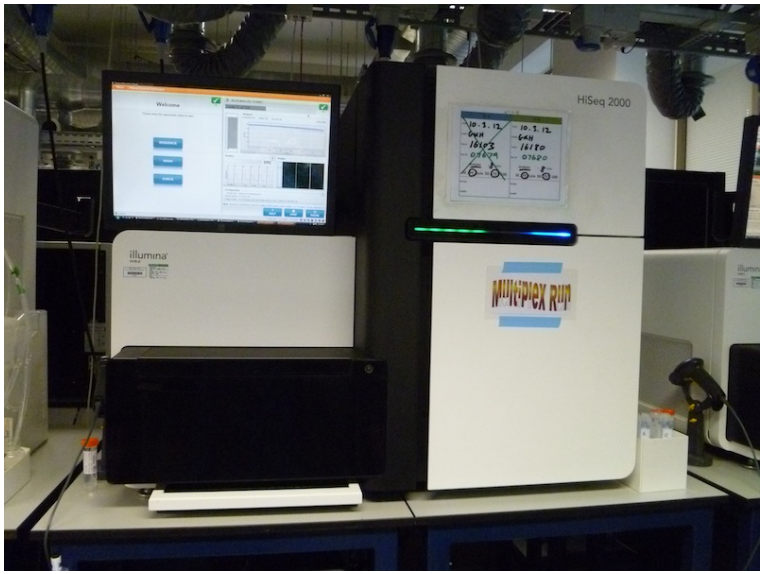
Short historical Introduction

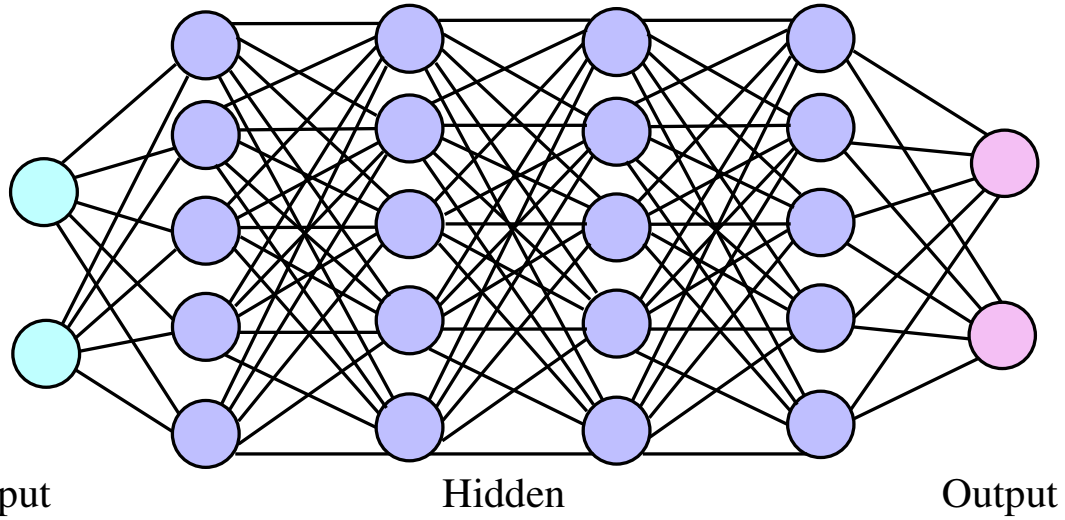
- Genetics as a natural science started in 1866: **Gregor Mendel** performed experiments that pointed to the existence of **biological elements called genes.**
- **Deoxy-ribonucleic acid (DNA)** isolated by **Friedrich Miescher** in 1869.
- 1944: Oswald Avery (and coworkers) identified DNA as the major carrier of genetic material, **responsible for inheritance.**
  Ribose: (simple) sugar molecule, deoxy-ribose $\rightsquigarrow$ loss of oxygen atom.
  Nucleic acid: overall name for DNA and RNA (large biomolecules). Named for their initial discovery in nucleus of cells, and for presence of phosphate groups (related to phosphoric acid).



Short historical Introduction

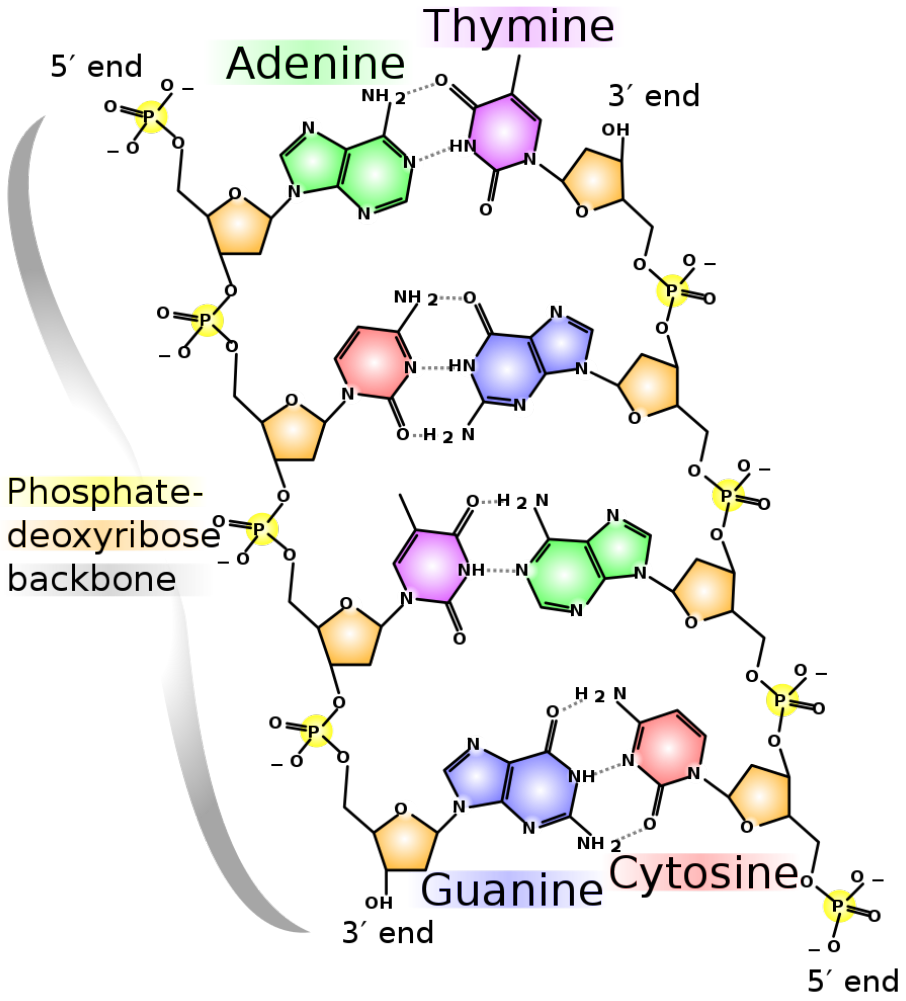
- 1953, Watson & Crick: **3-dimensional structure of DNA.** They inferred the method of **DNA replication.**
- 2001: first draft of the **human genome** published by the **Human Genome Project** and the company **Celera.**
- Many new developments, such as **Next Generation Sequencing**, **Deep learning** etc.





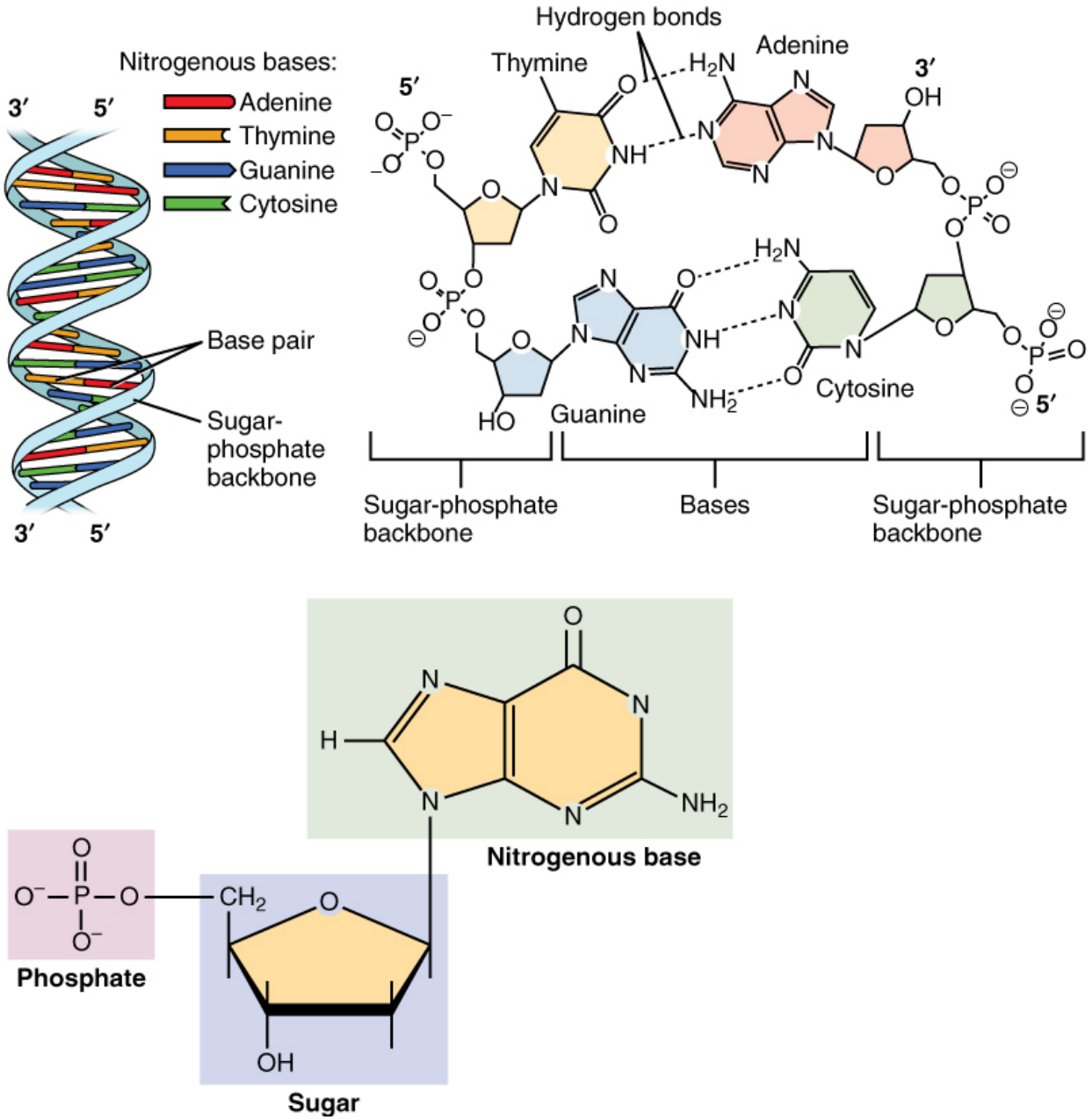
By RE73 - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=18862884

Base pairs and the DNA



By Madprime (talk Â· contribs) - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=1848174

- DNA composed of 4 basic molecules $\rightsquigarrow$ **nucleotides**.
- Nucleotides are identical up to different **nitrogen base:** organic molecule with a nitrogen atom that has the chemical properties of a base (due to free electron pair at nitrogen atom).
- Each nucleotide contains **phosphate**, **sugar** (of deoxy-ribose type), and one of the 4 bases: **Adenine, Guanine, Cytosine, Thymine** (A,G,C,T).
- **Hydrogen bonds** between base pairs: $G \equiv C$, $A = T$.

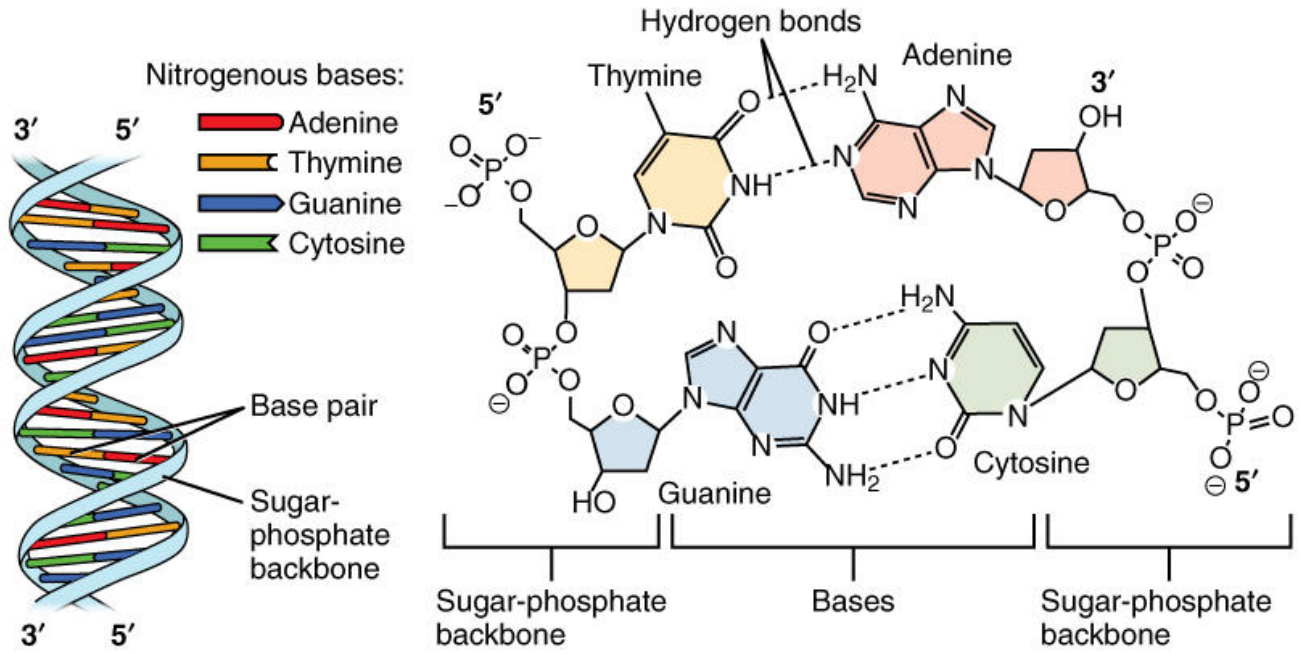


By OpenStax - https://cnx.org/contents/FPtK1zmh@8.25:fEI3C8Ot@10/Preface, CC BY 4.0, https://commons.wikimedia.org/w/index.php?curid=30131206

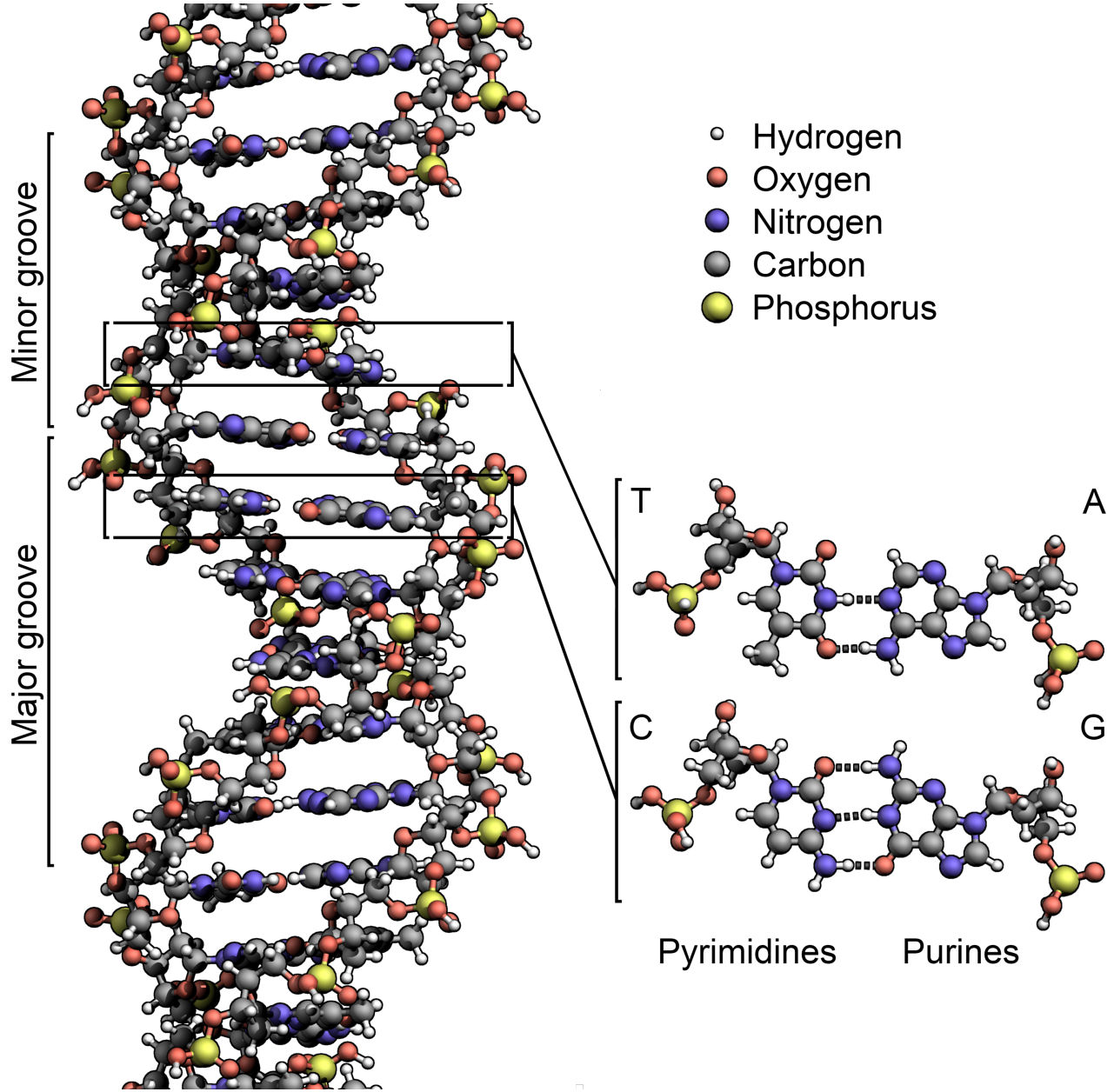
The structure of DNA

- DNA molecule is **directional** due to asymmetrical structure of the sugars which constitute the skeleton: Each sugar is connected to the strand **upstream** in its 5th carbon and to the strand **downstream** in its 3rd carbon.

- DNA strand goes from $5'$ to $3'$. The directions of the two complementary DNA strands are reversed to one another ($\rightsquigarrow$ **Reversed Complement**).



Adapted from https://commons.wikimedia.org/w/index.php?curid=30131206



By Zephyris - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=15027555

Replication of DNA

Biological process of producing two replicas of DNA from one original DNA molecule.

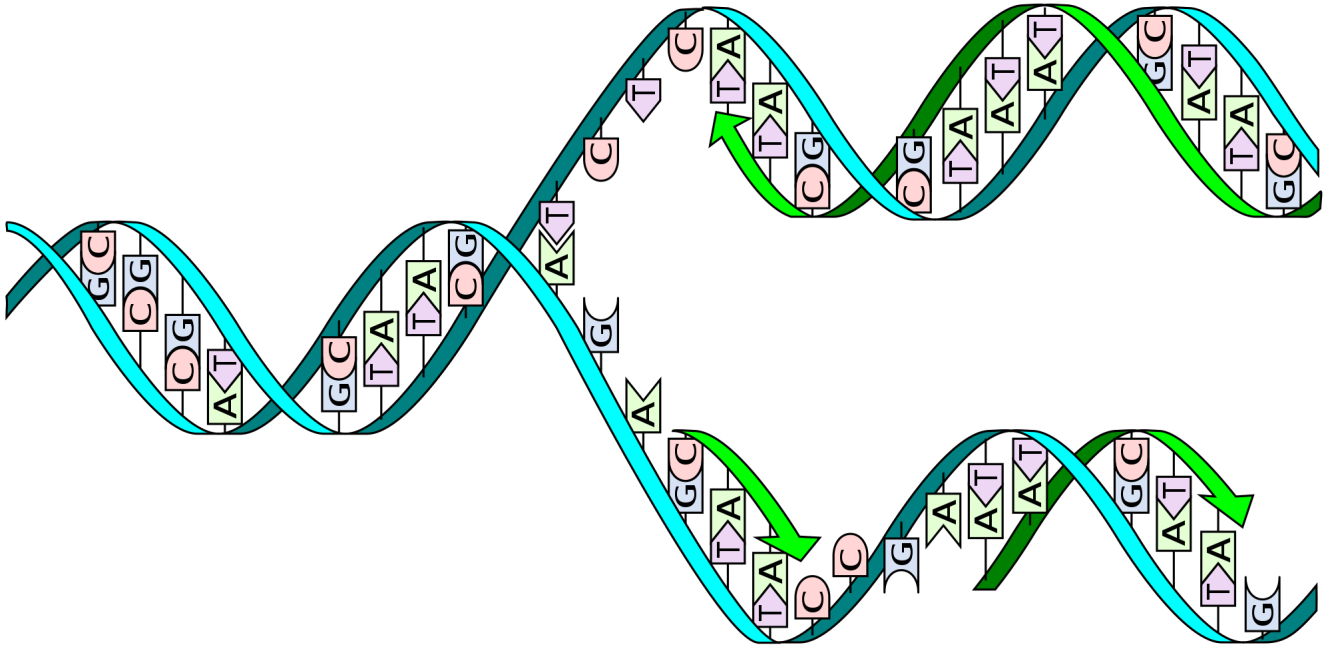
Cells have the distinctive property of division

$\rightsquigarrow$ DNA replication is most essential part for **biological inheritance.**

Unwinding $\rightsquigarrow$ single bases exposed on each strand.

Pairing requirements are **strict** $\rightsquigarrow$ single strands are templates for re-forming **identical** double helix (up to **mutations**).

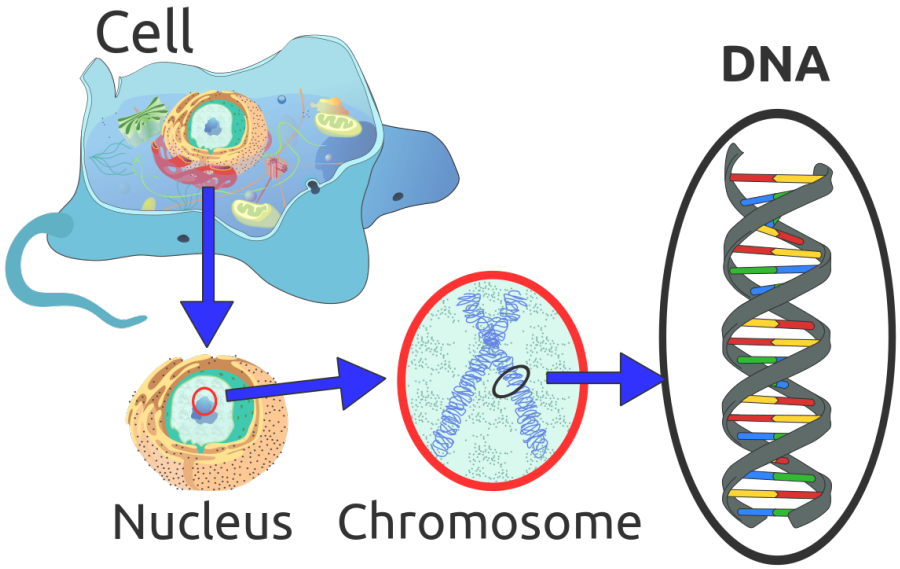
DNA polymerase: **enzyme** that catalyzes the synthesis of new DNA.



https://commons.wikimedia.org/w/index.php?curid=2497221

Genes and Chromosomes

- In higher organisms, DNA molecules are packed in a **chromosome**.
- **Genome:** total genetic information stored in the chromosomes.
- Every cell contains a **complete set** of the genome, differences are due to variable **expression** of genes.
- A **gene** is a sequence of nucleotides that encodes the synthesis of a gene product.

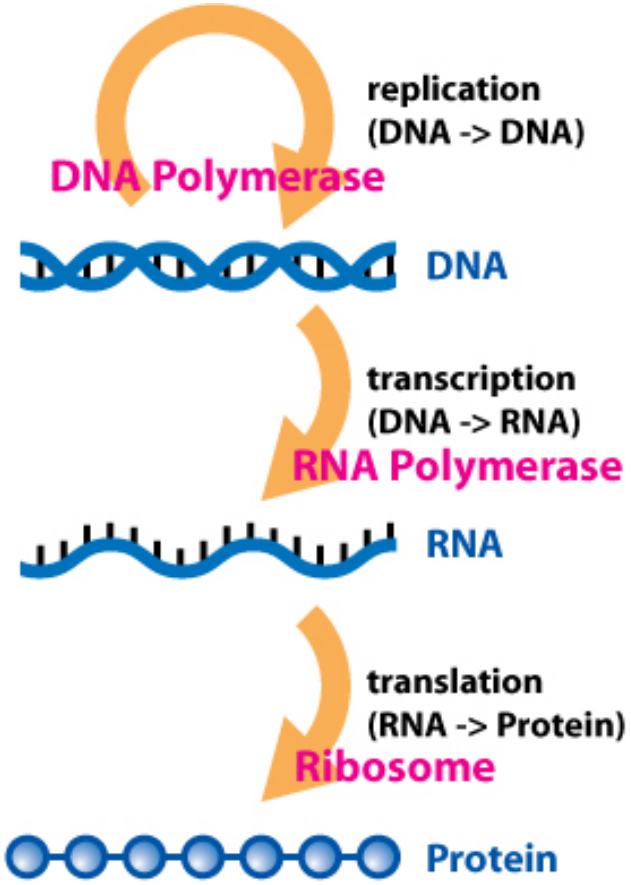


By Sponk, Tryphon, Magnus Manske,

https://commons.wikimedia.org/w/index.php?curid=20539140

- **Gene expression:** Process of synthesizing a gene product (often a protein) $\rightsquigarrow$ controls timing, location, and amount.

The Central Dogma

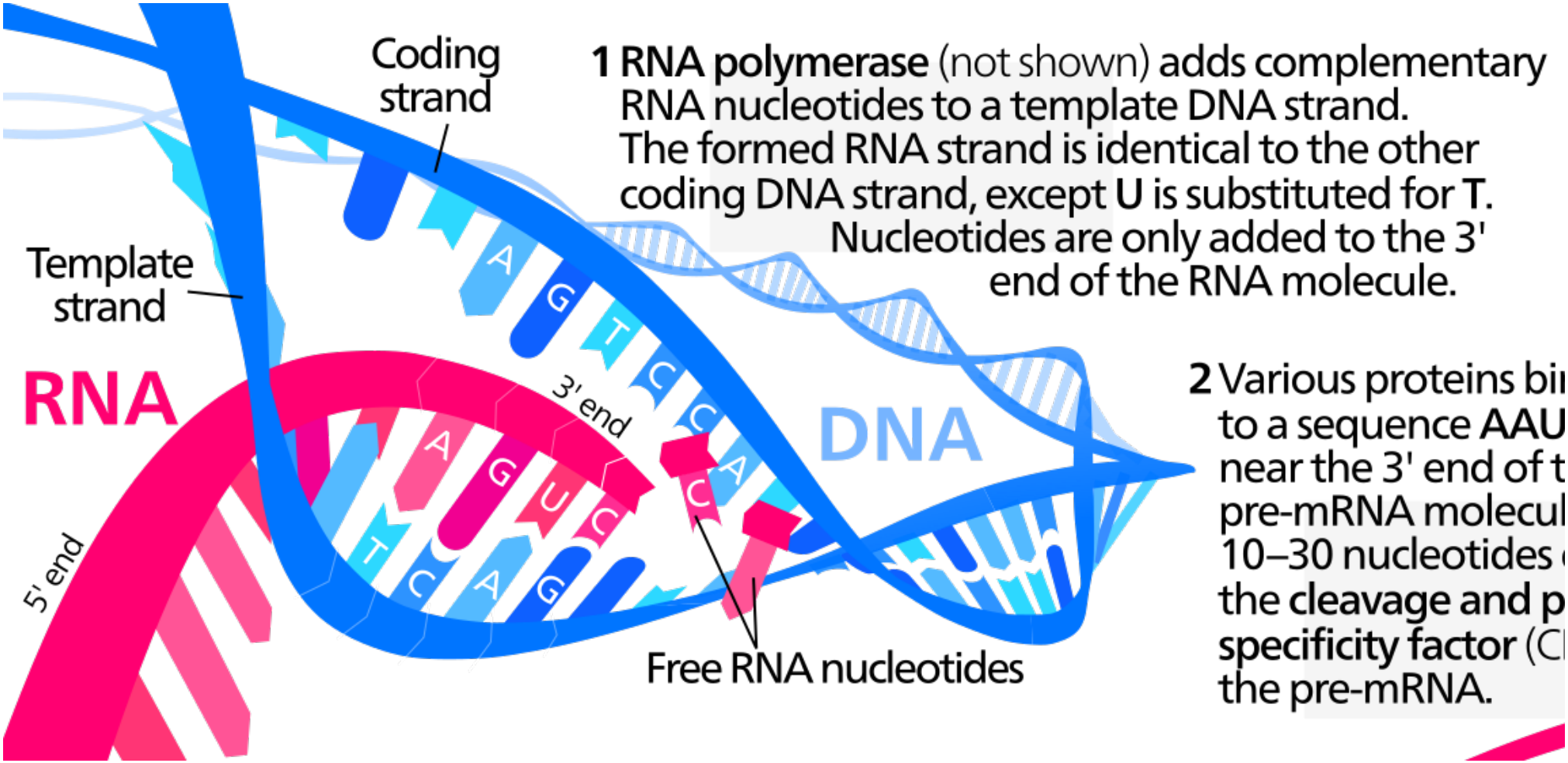


Wikipedia

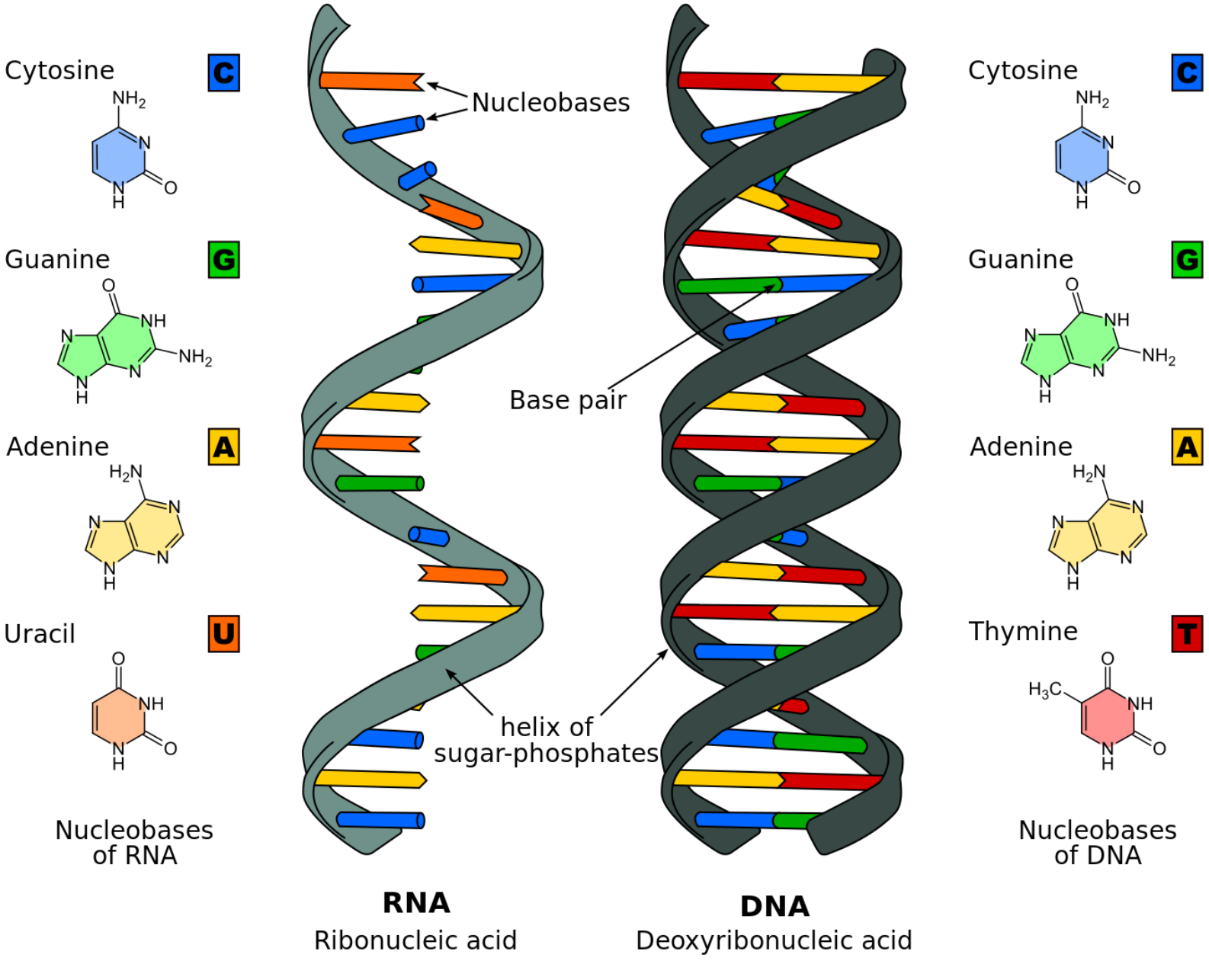
Transcription: making of an RNA molecule from DNA template.
Translation: construction of amino acid sequence from RNA.

$\Rightarrow$ Almost no exceptions ($\rightsquigarrow$ retroviruses)

Transcription



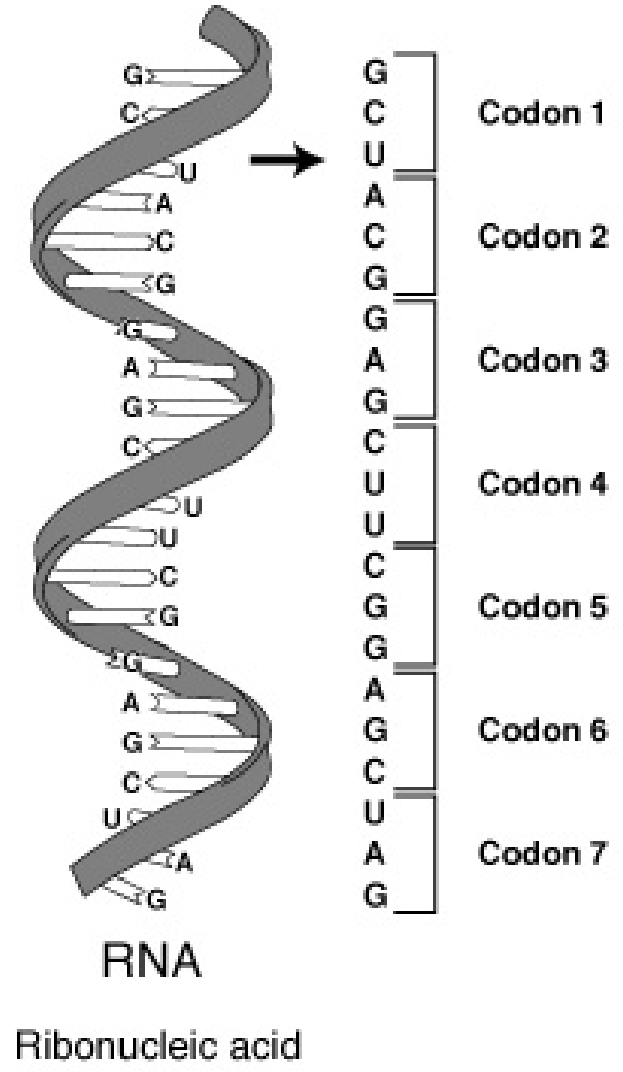
By Kelvinsong - Own work, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=23086203



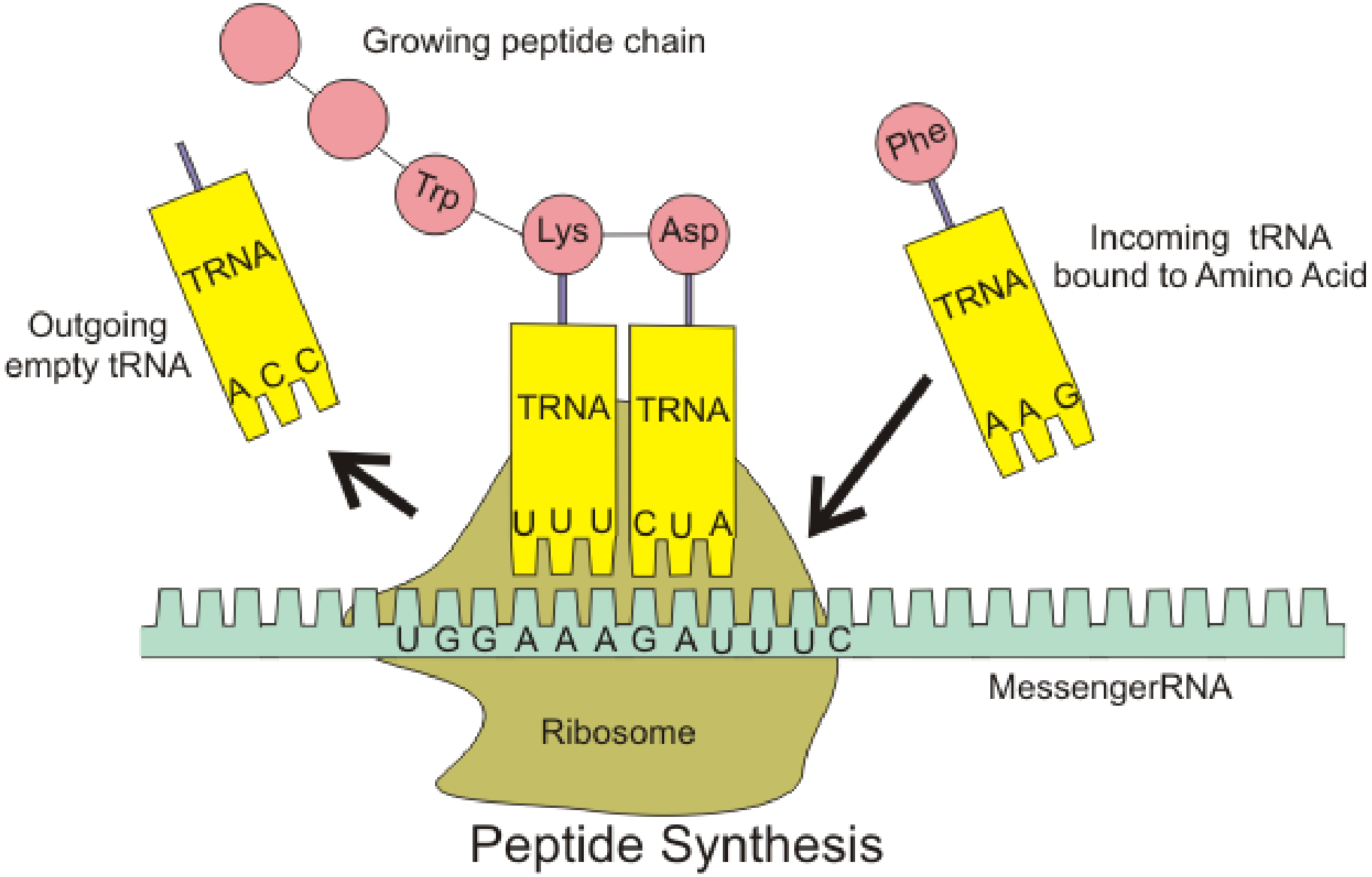
https://commons.wikimedia.org/w/index.php?curid=9810855

Translation

- mRNA molecules are translated by **ribosomes**: Enzyme that links together amino acids.
- Message is read **three bases at a time.**
- Initiated by the first AUG codon (codon = nucleotide triplet).
- Covalent bonds (=sharing of electron pairs) are made between adjacent amino acids
  ⇒ **growing chain of amino acids** ("polypeptide").
- When a **"stop" codon** (UAA, UGA, UAG) is encountered, translation stops.



Wikipedia



By Boumphreyfr - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=7200200

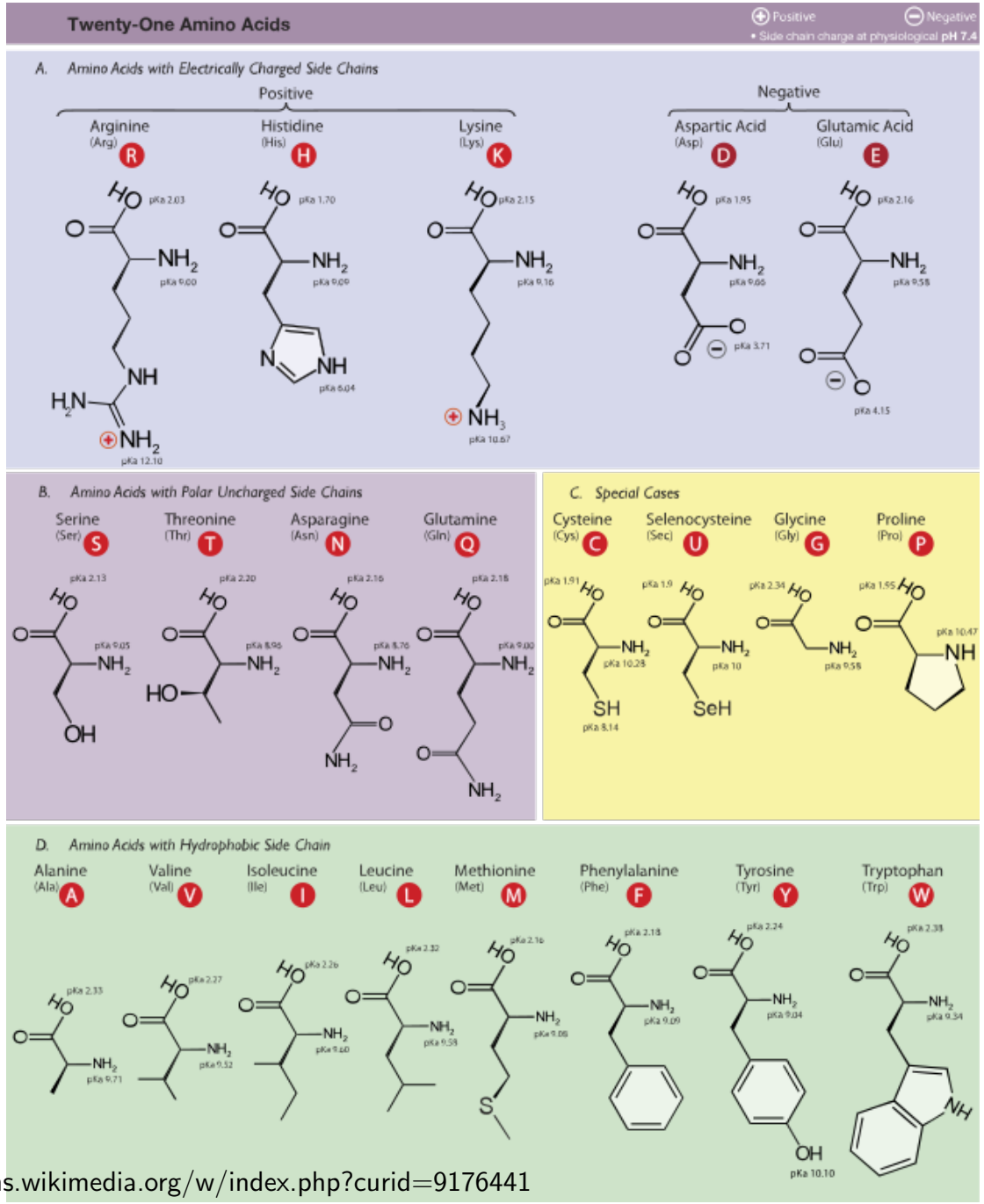
The genetic code

Standard genetic code

1st base	2nd base U		2nd base C		2nd base A		2nd base G		3rd base
U	UUU	(Phe/F) Phenylalanine	UCU	(Ser/S) Serine	UAU	(Tyr/Y) Tyrosine	UGU	(Cys/C) Cysteine	U
	UUC		UCC		UAC		UGC		C
	UUA	(Leu/L) Leucine	UCA		UAA[B]	Stop (*Ochre*)	UGA[B]	Stop (*Opal*)	A
	UUG		UCG		UAG[B]	Stop (*Amber*)	UGG	(Trp/W) Tryptophan	G
C	CUU		CCU	(Pro/P) Proline	CAU	(His/H) Histidine	CGU	(Arg/R) Arginine	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	(Gln/Q) Glutamine	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	(Ile/I) Isoleucine	ACU	(Thr/T) Threonine	AAU	(Asn/N) Asparagine	AGU	(Ser/S) Serine	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	(Lys/K) Lysine	AGA	(Arg/R) Arginine	A
	AUG[A]	(Met/M) Methionine	ACG		AAG		AGG		G
G	GUU	(Val/V) Valine	GCU	(Ala/A) Alanine	GAU	(Asp/D) Aspartic acid	GGU	(Gly/G) Glycine	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	(Glu/E) Glutamic acid	GGA		A
	GUG		GCG		GAG		GGG		G

Wikipedia

Highly redundant: only 20 (or 21) amino acids formed from $4^3 = 64$ possible combinations.



By Dancojocari. https://commons.wikimedia.org/w/index.php?curid=9176441

Proteins

- **Linear polymer of amino acids**, linked together by peptide bonds. Average size ≈ 200 amino acids, can be over 1000.

- To a large extent, **cells are made of proteins.**

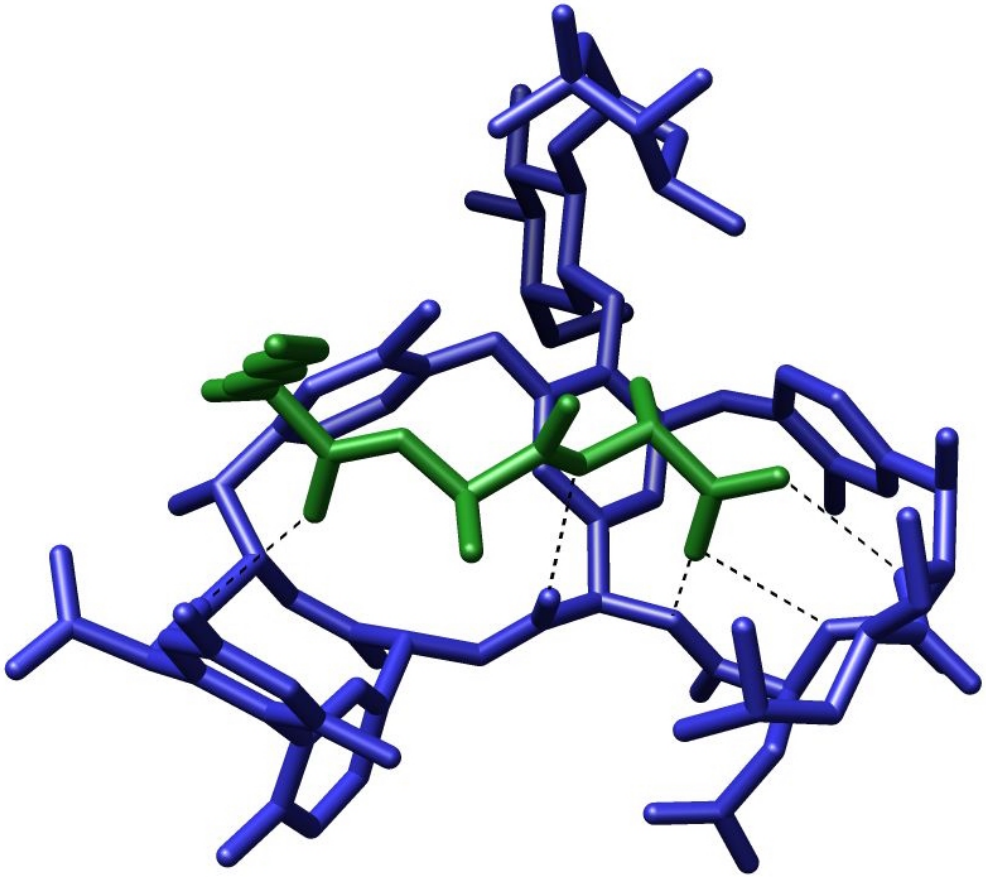
- Proteins determine **shape and structure of a cell**.
  Main instruments of **molecular recognition** and **catalysis**.

- **Complex structure** with four hierarchical levels.

  1. **Primary structure**: amino acid sequence.
  2. Different regions form locally regular **secondary structures** like α-*helices* and β-*sheets*.
  3. **Tertiary structure**: packing such structures into one or several 3D *domains*.
  4. Several domains arranged in a **quaternary structure**.

Molecular recognition

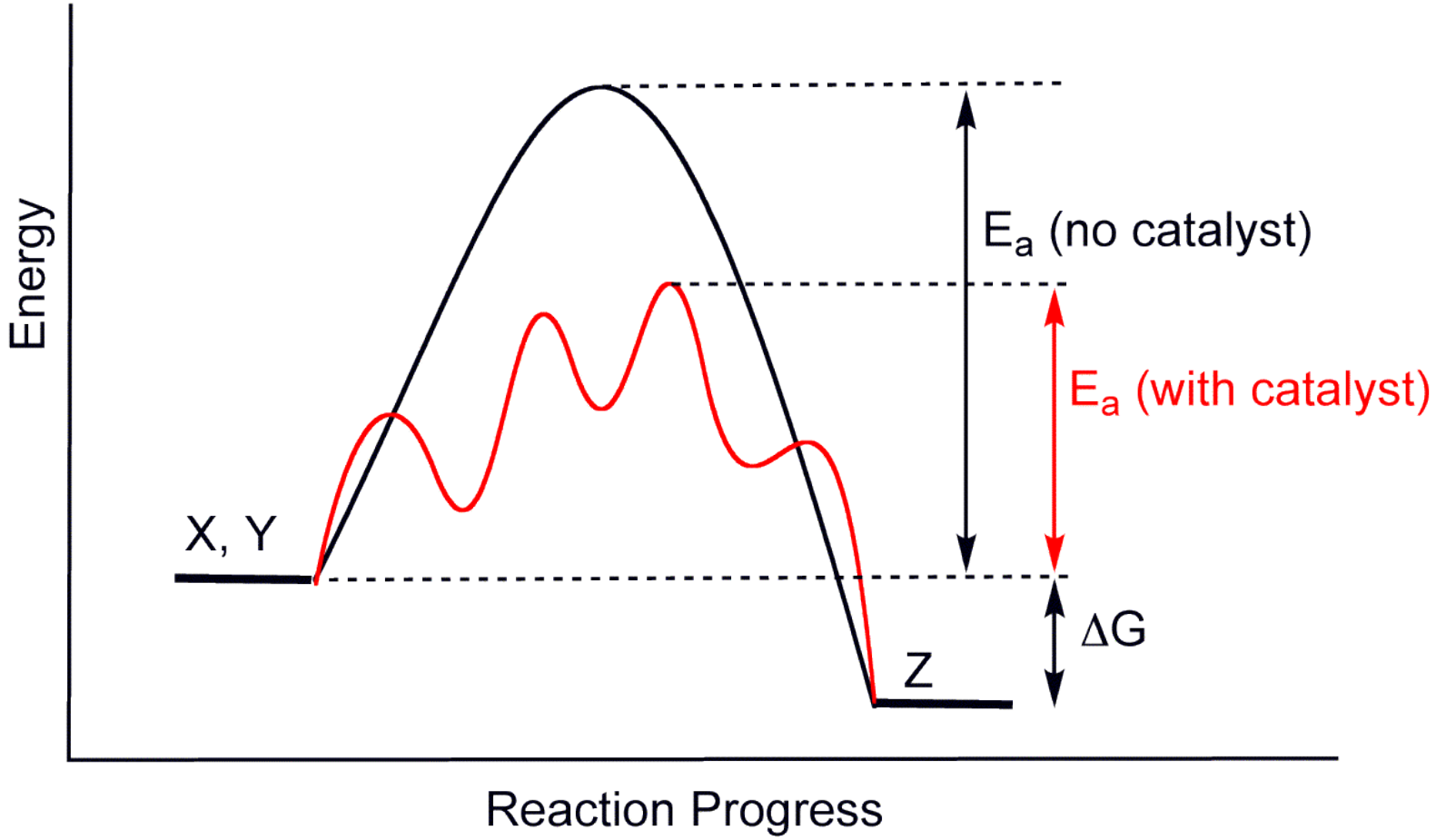
Interaction between molecules through noncovalent bonding



Crystal structure of a short peptide L-Lys-D-Ala-D-Ala (bacterial cell wall precursor) bound to the antibiotic vancomycin through hydrogen bonds. By M stone, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=2327682

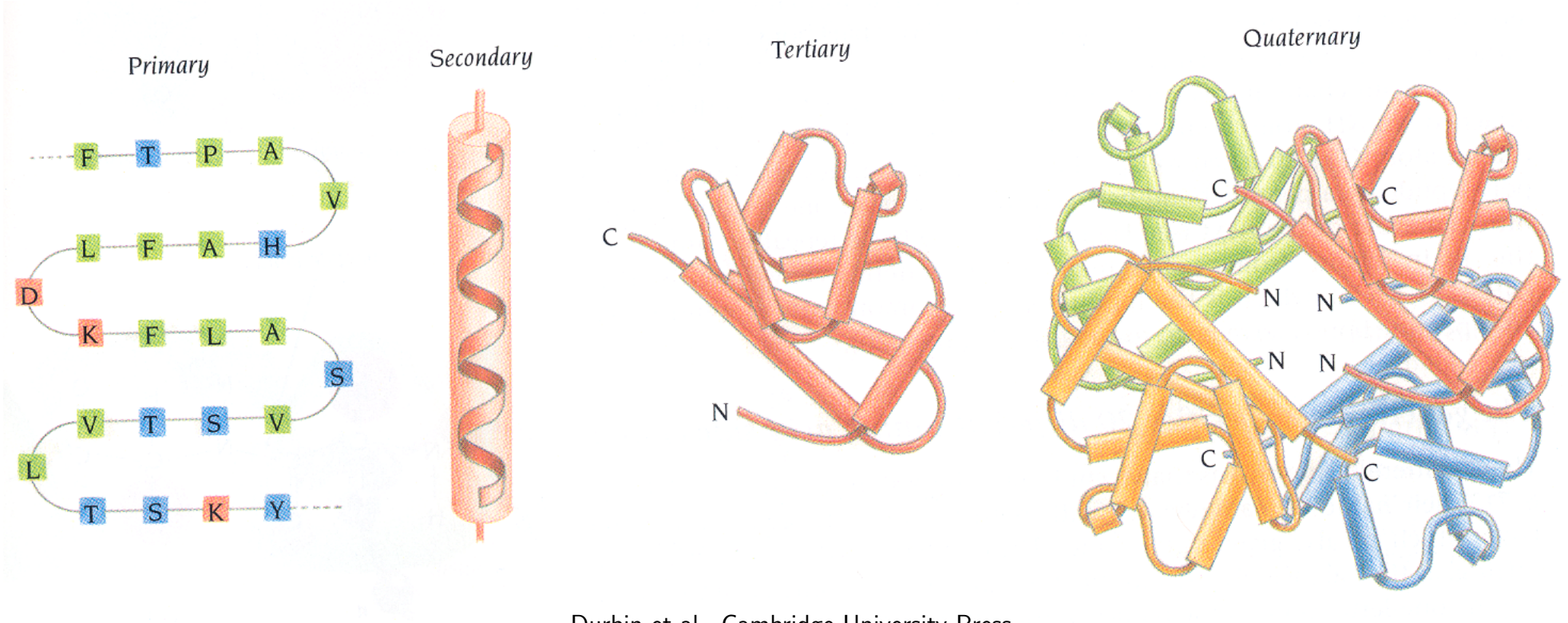
Catalysis

Increasing the rate of a chemical reaction by adding a substance $\rightsquigarrow$ catalyst.



Wikipedia

Protein Structure: primary to quaternary



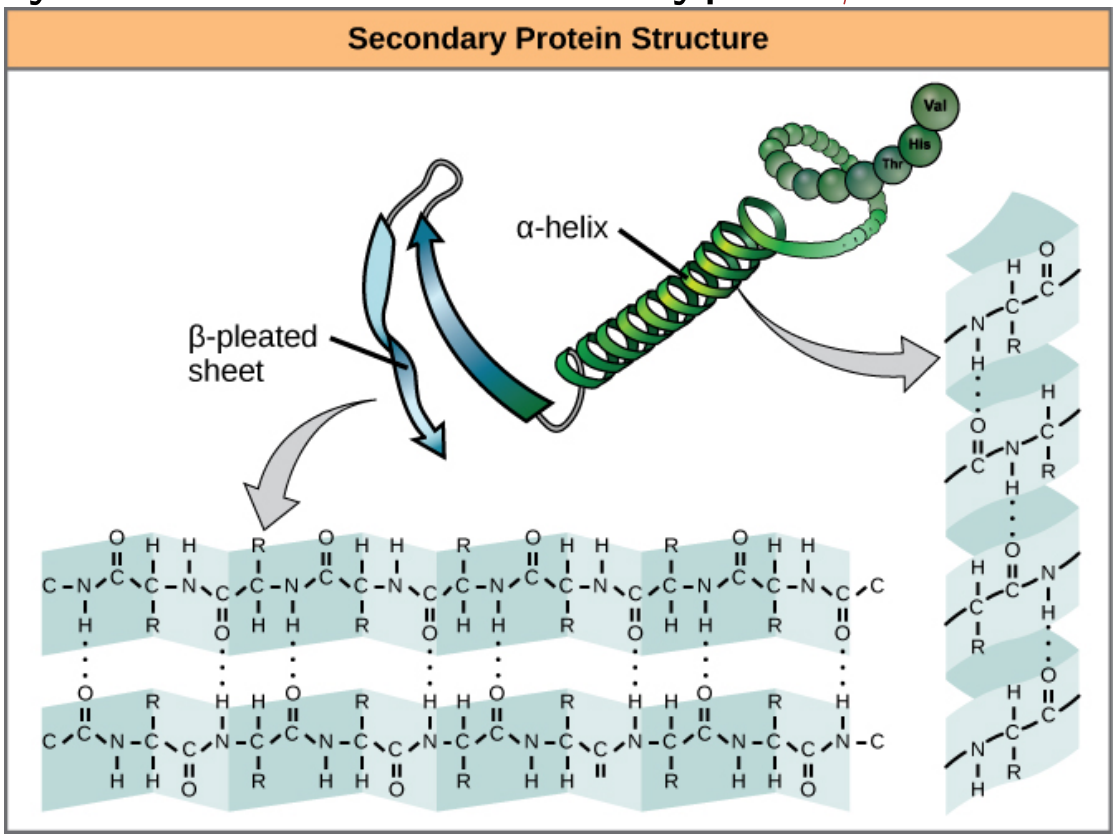
Durbin et al., Cambridge University Press

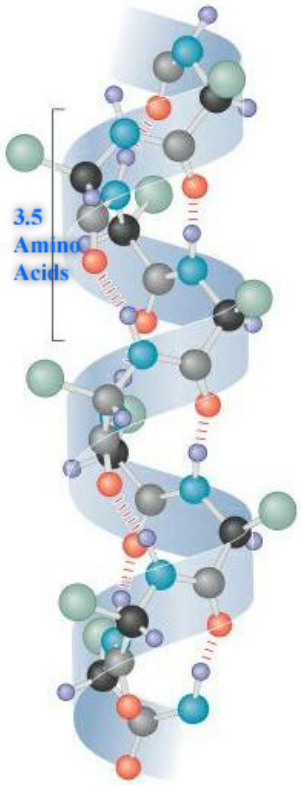
Structure is determined by the **primary sequence** and their **physico-chemical interactions** in the medium.

Structure determines functionality.

Secondary Structure

Secondary structure: two main types: *β-sheet* and *α-helix*





The School of Biomedical Sciences Wiki

Short range interactions in the AA chain are important for the secondary structure: α-helix performs a 100° turn *per amino acid* $\rightsquigarrow$ full turn after 3.6 AAs.

Formation of a helix mainly depends on interactions in a *4 AA* window.

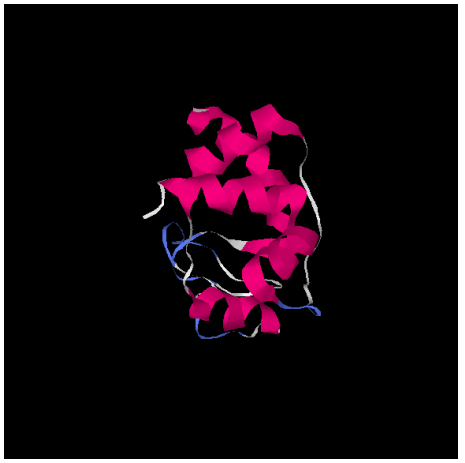
Example: Cytochrome C2 Precursor

```
Secondary structure (h=helix)
amino acid sequence

hhhhhhhhhh
MKKGFLAAGVFAAVAFASGAALAEGDAAAGEKVSKKCLACHTFDQGGANKVGPNLFGVFE
hhhhhhhh hhhhhhhhh hhhhhhhhh
NTAAHKDDYAYSESYTEMKAKGLTWTEANLAAYVKDPKAFVLEKSGDPKAKSKMTFKLTK

hhhhhhhhhhhhh

DDEIENVIAYLKTLK
```



Given: Examples of known helices and non-helices in several proteins

$\rightsquigarrow$ **training set**

Goal: Predict, mathematically, the existence and position of α-helices in **new proteins.**

Classification of Secondary Structure

Idea: Use a **sliding window** to cut the AA chain into pieces. 4 AAs are enough to capture one full turn $\rightsquigarrow$ choose window of size 5.

Decision Problem: Find function $f(\dots)$ that predicts for each substring in a window the structure:

$$f(\texttt{AADTG}) = \begin{cases} \text{"Yes"}, & \text{if the central AA belongs to an } \alpha\text{-helix,} \\ \text{"No"}, & \text{otherwise} \end{cases}$$

Problem: How should we numerically encode a string like `AADTG`?

Simple encoding scheme: Count the number of occurrences of each AA in the window. First order approximation, neglects AA's position within the window.

Example

...RAADTGGSDP...

...xxxhhhhhhx...

...xxxhhhhhhx...

...xxxhhhhhhx...

(black $\hat{=}$ structure info about central AA; green $\hat{=}$ know secondary structure; red$\hat{=}$ sliding window)

A	C	D	...	G	...	R	S	T	...	Y	Label
2	0	1	0	0	0	1	0	1	0	0	"No"
2	0	1	0	1	0	0	0	1	0	0	"Yes"
1	0	1	0	2	0	0	0	1	0	0	"Yes"
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

This is a binary classification problem

$\rightsquigarrow$ use **Linear Discriminant Analysis**

Discriminant Analysis

Consider $X_{n\times d}$, with $n = \#(\text{windows})$ and $d = \#(\text{AAs}) = 20(\text{or } 21)$, and the n-vector of class indicators $\boldsymbol{y}$

$$X = \begin{bmatrix} 2 & 0 & 1 & \dots & 0 & \dots \\ 2 & 0 & 1 & \dots & 1 & \dots \\ 1 & 0 & 1 & \dots & 2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} - & \boldsymbol{x}_1^t & - \\ - & \boldsymbol{x}_2^t & - \\ & \vdots & \\ - & \boldsymbol{x}_n^t & - \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} \text{"No"} \\ \text{"Yes"} \\ \text{"Yes"} \\ \vdots \end{bmatrix}$$

For the binary class idicators, we use some numerical encoding scheme.

Interpretation with basis functions:

$$\begin{aligned} \boldsymbol{x} &= \text{sequence of characters from alphabet } \mathcal{A} \\ g_i(\boldsymbol{x}) &= \#(\text{occurences of letter } i \text{ in sequence}) \\ f(\boldsymbol{x};\boldsymbol{w}) &= \boldsymbol{w}^t\boldsymbol{g} = \sum_{i\in\text{characters}} w_i g_i(\boldsymbol{x}) \end{aligned}$$

Discriminant analysis and least squares

Recall: The LDA vector $\hat{\boldsymbol{w}}^{\text{LDA}} = \Sigma_W^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2)$ coincides with the solution of the LS problem $\hat{\boldsymbol{w}}^{\text{LS}} = \arg\min_{\boldsymbol{w}} \|X\boldsymbol{w} - \boldsymbol{y}\|^2$ if

$$
\begin{aligned}
n_1 &= \#\text{ samples in class } \mathbf{1} \\
n_2 &= \#\text{ samples in class } \mathbf{2} \\
X &= \begin{bmatrix} - & \boldsymbol{x}_1^t & - \\ - & \boldsymbol{x}_2^t & - \\ & \vdots & \\ - & \boldsymbol{x}_n^t & - \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix},
\end{aligned}
$$

with $\dfrac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i = \boldsymbol{m} = \mathbf{0}$ (i.e. origin in sample mean),

$$
y_i = \begin{cases} +1/n_1, & \text{if } \boldsymbol{x}_i \text{ in class } \mathbf{1} \\ -1/n_2, & \text{else.} \end{cases} \qquad \Rightarrow \sum_{i=1}^{n} y_i = 0
$$

Singular Value Decomposition (SVD)

Recall: SVD for nonsquare matrix $X \in \mathbb{R}^{n \times d}$: $X = USV^t$.

Residual sum of squares:

$$RSS = \|\boldsymbol{r}\|^2 = \|X\boldsymbol{w} - \boldsymbol{y}\|^2 = \left\|USV^t\boldsymbol{w} - \boldsymbol{y}\right\|^2 = \left\|S\underbrace{V^t\boldsymbol{w}}_{\boldsymbol{z}} - \underbrace{U^t\boldsymbol{y}}_{\boldsymbol{c}}\right\|^2$$

Minimizing $\|\boldsymbol{r}\|^2$ is equivalent to minimizing $\|S\boldsymbol{z} - \boldsymbol{c}\|^2$:

$$\text{minimize } \|\boldsymbol{r}\|^2 = \left\| \left[\begin{array}{ccc} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_d \\ \hline 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{array}\right] \cdot \begin{bmatrix} z_1 \\ \vdots \\ z_d \end{bmatrix} - \begin{bmatrix} c_1 \\ \vdots \\ c_d \\ c_{d+1} \\ \vdots \\ c_n \end{bmatrix} \right\|^2$$

We now choose z_k so that $\|\boldsymbol{r}\|^2$ is minimal, i.e., for $\sigma_k > 0$:

$$z_k = \frac{c_k}{\sigma_k}$$

Iterative Algorithm

In our problem we have $d = 20$ (or 21) and $n > 10000$.
Goal: Use only $X^tX \in \mathbb{R}^{d\times d}$ and $X^t\boldsymbol{y} \in \mathbb{R}^d$.
Initialize $X^tX = 0$ (zero matrix), $X^t\boldsymbol{y} = \mathbf{0}$. **Update:** for $j = 1$ to n :

$$X^tX + \boldsymbol{x}_j\boldsymbol{x}_j^t \longrightarrow X^tX$$
$$X^t\boldsymbol{y} + \boldsymbol{x}_j y_j \longrightarrow X^t\boldsymbol{y}$$

The first update procedure is correct, since

$$\left(X^tX\right)_{ik} = \sum_{j=1}^{n} x_{ji}x_{jk}$$
$$\Rightarrow X^tX = \sum_{j=1}^{n} \begin{bmatrix} x_{j1} \\ x_{j2} \\ \vdots \\ x_{jd} \end{bmatrix} \cdot [x_{j1}, x_{j2}, \ldots, x_{jd}] = \sum_{j=1}^{n} \boldsymbol{x}_j\boldsymbol{x}_j^t$$

Iterative Algorithm

A similar calculation yields the other equation:

$$\left(X^t \boldsymbol{y}\right)_i = \sum_j x_{ji} y_j \Rightarrow X^t \boldsymbol{y} = \sum_j \begin{bmatrix} x_{j1} \\ x_{j2} \\ \vdots \\ x_{jd} \end{bmatrix} \cdot y_j = \sum_{j=1}^{n} \boldsymbol{x}_j y_j$$

One remaining problem: In LDA we assumend that X **was centered**, i.e. the column sums are all zero. Compute the column sums as:

$$\mathbf{1}^t X = [1, 1, \ldots, 1] \begin{bmatrix} - & \boldsymbol{x}_1^t & - \\ - & \boldsymbol{x}_2^t & - \\ & \vdots & \\ - & \boldsymbol{x}_n^t & - \end{bmatrix} = n \cdot [m_1, m_2, \ldots, m_d] = n \cdot \boldsymbol{m}^t$$

$\rightsquigarrow$ **"centered"** $X_c = X - \mathbf{1}\boldsymbol{m}^t = X - \frac{1}{n}\mathbf{1}\mathbf{1}^t X$

Centering

$$X_c = X - \mathbf{1}\boldsymbol{m}^t = X - \frac{1}{n}\mathbf{1}\mathbf{1}^t X$$

$$X_c^t X_c = X^t X + \frac{1}{n^2} X^t \mathbf{1} \underbrace{\mathbf{1}^t \mathbf{1}}_{=n} \mathbf{1}^t X - \frac{1}{n} X^t \mathbf{1}\mathbf{1}^t X - \frac{1}{n} X^t \mathbf{1}\mathbf{1}^t X$$

$$= X^t X - \frac{1}{n} X^t \mathbf{1}\mathbf{1}^t X$$

$$= X^t X - n \cdot \boldsymbol{m}\boldsymbol{m}^t$$

Iteratively update the vector $n \cdot \boldsymbol{m}$ for every $\boldsymbol{x}_i$ corresponding to a new window position: **Initialize $n \cdot \boldsymbol{m} = \mathbf{0}$ and update $n \cdot \boldsymbol{m} \leftarrow n \cdot \boldsymbol{m} + \boldsymbol{x}_i$**

What about $X^t \boldsymbol{y}$? We should have used

$$X_c^t \boldsymbol{y} = (X - \mathbf{1}\boldsymbol{m}^t)^t \boldsymbol{y} = (X^t - \boldsymbol{m}\,\mathbf{1}^t)\boldsymbol{y} = X^t \boldsymbol{y} - \boldsymbol{m}\,\mathbf{1}^t \boldsymbol{y}$$

But by construction, $\boldsymbol{y}$ is orthogonal to $\mathbf{1} \rightsquigarrow \mathbf{1}^t \boldsymbol{y} = 0$,
so nothing needs to be done!

Iterative Algorithm

Goal: Solution which only requires ${X_c}^t X_c \in \mathbb{R}^{d\times d}$ and ${X_c}^t \boldsymbol{y} \in \mathbb{R}^d$ alone (and does not use X_c or $\boldsymbol{y}$ explicitly).

We need:

- The matrix V (for computing $\hat{\boldsymbol{w}} = V\boldsymbol{z}$)
  Solution: columns of V are the eigenvectors of $X_c^t X_c$, corresponding eigenvalues are $\lambda_i,\ i = 1, \ldots, n \Rightarrow \sigma_i^2 = \lambda_i$
- For the nonzero SVs, we need $z_i = (U^t\boldsymbol{y})_i/\sigma_i = \sigma_i(U^t\boldsymbol{y})_i/\sigma_i^2$
  Solution:

$$X_c = USV^t \ \Rightarrow\ V^t X_c^t \boldsymbol{y} = V^t V S^t U^t \boldsymbol{y} = S^t U^t \boldsymbol{y}$$

$$\Rightarrow\ z_i = (U^t\boldsymbol{y})_i/\sigma_i = (V^t X_c^t \boldsymbol{y})_i/\sigma_i^2$$

So $\boldsymbol{z}$ and finally $\hat{\boldsymbol{w}} = V\boldsymbol{z}$ can be computed from $X_c^t X_c$ and $X_c^t \boldsymbol{y}$ alone!

Chapter 2

Least squares problems

Least-squares and dimensionality reduction

Least-squares and dimensionality reduction

Given n data points in d dimensions:

$$X = \begin{bmatrix} - & \boldsymbol{x}_1^t & - \\ - & \boldsymbol{x}_2^t & - \\ - & \vdots & - \\ - & \boldsymbol{x}_n^t & - \end{bmatrix} \in \mathbb{R}^{n \times d}$$

Want to reduce dimensionality from d to k. Choose k directions $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k$, arrange them as columns in matrix W:

$$W = \begin{bmatrix} | & | & & | \\ \boldsymbol{w}_1 & \boldsymbol{w}_2 & \ldots & \boldsymbol{w}_k \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{d \times k}$$

Project $\boldsymbol{x} \in \mathbb{R}^d$ down to $\boldsymbol{z} = W^t \boldsymbol{x} \in \mathbb{R}^k$. How to choose W?

Encoding–decoding model

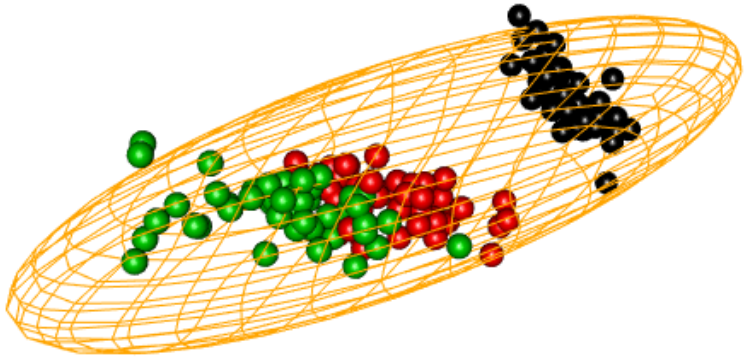
The projection matrix W serves two functions:

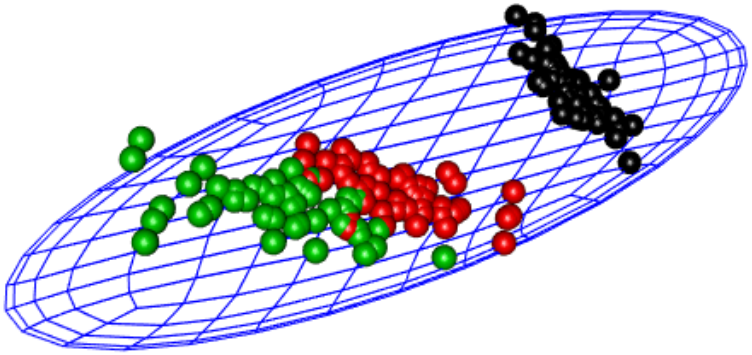
- **Encode:** $\boldsymbol{z} = W^t \boldsymbol{x}, \quad \boldsymbol{z} \in \mathbb{R}^k, \quad z_j = \boldsymbol{w}_j^t \boldsymbol{x}.$
  - The vectors $\boldsymbol{w}_j$ form a basis of the projected space.
  - We will require that this basis is orthonormal, i.e. $W^t W = I$.
- **Decode:** $\tilde{\boldsymbol{x}} = W\boldsymbol{z} = \sum_{j=1}^{k} z_j \boldsymbol{w}_j, \quad \tilde{\boldsymbol{x}} \in \mathbb{R}^d.$
  - If $k = d$, the above orthonormality condition implies $W^t = W^{-1}$, and encoding can be undone without loss of information.
  - If $k < d$, the decoded $\tilde{\boldsymbol{x}}$ can only approximate $\boldsymbol{x}$
    $\rightsquigarrow$ the reconstruction error will be nonzero.
- Note that we did not include an intercept term. Assumption: origin of coordinate system is in the sample mean, i.e. $\sum_i \boldsymbol{x}_i = 0$.

Principal Component Analysis (PCA)

We want the reconstruction error $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^2$ to be small.

Objective: minimize $\min_{W \in \mathbb{R}^{d \times k} : W^t W = I} \sum_{i=1}^{n} \|\boldsymbol{x}_i - W W^t \boldsymbol{x}_i\|^2$





Finding the principal components

Projection vectors are orthogonal $\rightsquigarrow$ can treat them separately:

$$\min_{\boldsymbol{w}:\,\|\boldsymbol{w}\|=1} \sum_{i=1}^{n} \left\|\boldsymbol{x}_i - \boldsymbol{w}\boldsymbol{w}^t\boldsymbol{x}_i\right\|^2$$

$$\begin{aligned}
\sum_i \left\|\boldsymbol{x}_i - \boldsymbol{w}\boldsymbol{w}^t\boldsymbol{x}_i\right\|^2 &= \sum_{i=1}^{n} [\boldsymbol{x}_i^t\boldsymbol{x}_i - 2\boldsymbol{x}_i^t\boldsymbol{w}\boldsymbol{w}^t\boldsymbol{x}_i + \boldsymbol{x}_i^t\boldsymbol{w}\underbrace{\boldsymbol{w}^t\boldsymbol{w}}_{=1}\boldsymbol{w}^t\boldsymbol{x}_i] \\
&= \sum_i [\boldsymbol{x}_i^t\boldsymbol{x}_i - \boldsymbol{x}_i^t\boldsymbol{w}\,\boldsymbol{w}^t\boldsymbol{x}_i] \\
&= \sum_i \boldsymbol{x}_i^t\boldsymbol{x}_i - \sum_i \boldsymbol{w}^t\boldsymbol{x}_i\,\boldsymbol{x}_i^t\boldsymbol{w} \\
&= \sum_i \boldsymbol{x}_i^t\boldsymbol{x}_i - \boldsymbol{w}^t\Big(\sum_{i=1}^{n} \boldsymbol{x}_i\boldsymbol{x}_i^t\Big)\boldsymbol{w} \\
&= \underbrace{\sum_i \boldsymbol{x}_i^t\boldsymbol{x}_i}_{\text{const.}} - \boldsymbol{w}^t X^t X\boldsymbol{w}.
\end{aligned}$$

Finding the principal components

- Want to maximize $\boldsymbol{w}^t X^t X \boldsymbol{w}$ under the constraint $\|\boldsymbol{w}\| = 1$
- Can also maximize the ratio $J(\boldsymbol{w}) = \frac{\boldsymbol{w}^t X^t X \boldsymbol{w}}{\boldsymbol{w}^t \boldsymbol{w}}$.
- Optimal projection $\boldsymbol{w}$ is the eigenvector of $X^t X$ with largest eigenvalue (compare handout on spectral matrix norm).
- We assumed $\sum_i \boldsymbol{x}_i = \boldsymbol{0}$, i.e. the columns of X sum to zero.
  $\rightsquigarrow$ compute SVD of "centered" matrix X_c
  $\rightsquigarrow$ column vectors in W are eigenvectors of $X_c^t X_c$
  $\rightsquigarrow$ they are the principal components.

Eigen-faces [Turk and Pentland, 1991]

- $d =$ number of pixels
- Each $\boldsymbol{x}_i \in \mathbb{R}^d$ is a face image
- $x_{ij} =$ intensity of the j-th pixel in image i

$$\begin{array}{ccc} \boldsymbol{x}_i & \approx & WW^t\boldsymbol{x}_i = W\boldsymbol{z}_i \\ (X^t)_{d\times n} & \approx & W_{d\times k} \qquad (Z^t)_{k\times n} \end{array}$$

$$\left(\begin{array}{ccc} \\ \end{array} \right. \quad \cdots \quad \left. \begin{array}{ccc} \\ \end{array} \right) \approx \left(\begin{array}{ccccc} \\ \end{array} \right. \left. \begin{array}{ccccc} \\ \end{array} \right) \begin{bmatrix} | & & | \\ \boldsymbol{z}_1 & \cdots & \boldsymbol{z}_n \\ | & & | \end{bmatrix}$$

Conceptual: We can lean something about the structure of face images.
Computational: Can use $\boldsymbol{z}_i$ for efficient nearest-neighbor classification: Much faster when $k \ll d$.

Information retrieval: Latent Semantic Analysis [Deerwater, 1990]

- $d =$ number of words in the vocabulary, say 10000.
- Each $\boldsymbol{x}_i \in \mathbb{R}^d$ is a vector of word counts
- $x_{ij} =$ frequency of word j in document i

$$
\underset{(X^t)_{d\times n}}{\begin{bmatrix} \text{stocks:} & 2 & \ldots\ldots & 0 \\ \text{chairman:} & 4 & \ldots\ldots & 1 \\ \text{the:} & 8 & \ldots\ldots & 7 \\ \ldots & \vdots & \ldots\ldots & \vdots \\ \text{wins:} & 0 & \ldots\ldots & 2 \\ \text{game:} & 1 & \ldots\ldots & 3 \end{bmatrix}} \approx \underset{W_{d\times k}}{\begin{bmatrix} 0.4 & \ldots & -0.001 \\ 0.8 & \ldots & 0.03 \\ 0.01 & \ldots & 0.04 \\ \vdots & \ldots & \vdots \\ 0.002 & \ldots & 2.3 \\ 0.003 & \ldots & 1.9 \end{bmatrix}} \approx \underset{(Z^t)_{k\times n}}{\begin{bmatrix} | & & | \\ \boldsymbol{z}_1 & \ldots & \boldsymbol{z}_n \\ | & & | \end{bmatrix}}
$$

How to measure similarity between two documents? Dot products $\boldsymbol{x}_i^t \boldsymbol{x}_j$

In such high-dimensional spaces most pairs of vectors are almost orthogonal $\rightsquigarrow$ scalar products tend to be "noisy".

If $k \ll d$, $\boldsymbol{z}_i^t \boldsymbol{z}_j$ is probably a better similarity measure than $\boldsymbol{x}_i^t \boldsymbol{x}_j$.