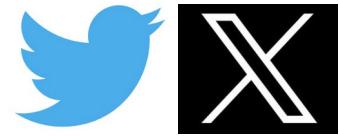


Machine Learning for Computational Biology

Nikolay Oskolkov, Lund University, NBIS SciLifeLab, Sweden
AFI course, 16.02.2026



@NikolayOskolkov



Brief introduction: who am I

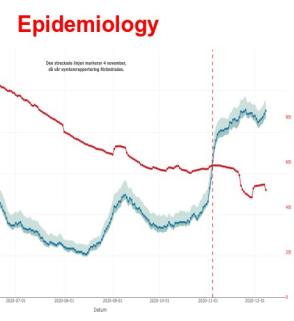
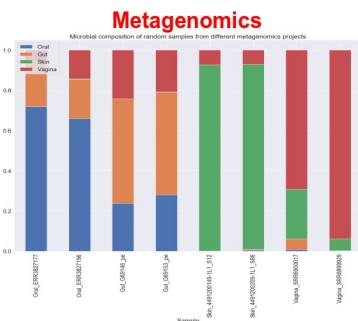
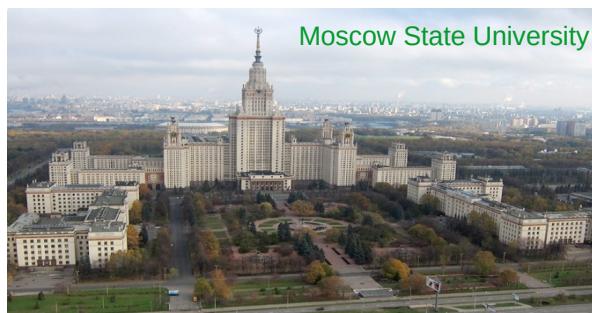
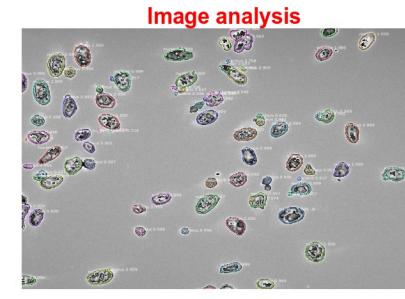
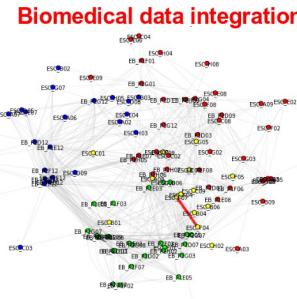
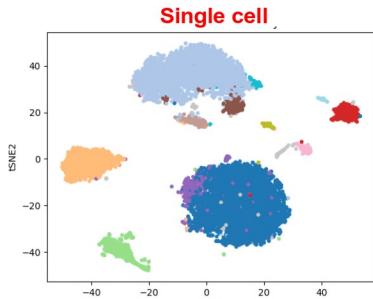
2007 PhD in theoretical physics in Moscow, Russia

2011 medical genomics at Lund University, Sweden

2016 bioinformatician at NBIS SciLifeLab, Sweden

2025 Metabolic Research Group leader, LIOS, Latvia

50+ publications; h-index = 25; 4,000+ citations



NIRI=LIOS+BMC



Metabolic Research Group at LIOS



Publications

Conferences

Metabolic Research Group

Metabolic Research Group

The Metabolic Research Group (MRG) focuses on advancing computational methods to identify and validate novel drug targets for metabolic diseases. Our research profile centers on the development and application of machine learning approaches, combined with statistical modeling, to extract biological knowledge from complex datasets. A key expertise of the group is the integration of diverse multiOmic data—including genomics, transcriptomics, proteomics, metabolomics, and metagenomics—enabling a systems-level understanding of metabolic processes and disease mechanisms. Through this integrative and data-driven approach, we aim to contribute to precision medicine by supporting the discovery of innovative therapeutic strategies within the TARGETWISE project.

1 more postdoctoral fellow and 1 PhD student to be hired

**If you know anyone who might be interested,
please contact me!**



PhD. Nikolay Oskolkov
Group Leader (PI) of the Metabolic
Research Group



Kristina Grausa, PhD student
in Metabolic Research Group



Daniel Rivas, MD, PhD in AI,
postdoctoral fellow
in Metabolic Research Group

Plan for the course:

Day 1: Introduction to Frequentist and Bayesian statistics (challenge of biological high-dimensional data, curse of dimensionality, feature selection via LASSO, Bayes theorem and priors, Markov Chain Monte Carlo)

Day 2: From statistics to machine learning (differences between statistics and machine learning, cross-validation, supervised and unsupervised machine learning, k-means clustering, underfitting and overfitting, ROC-curves)

Day 3: From linear models to neural networks and Random Forest (universal approximation theorem, gradient descent, learning rate, forward and backward propagation, decision tree, Gini index, optimal split and new data for split, applications of Deep Learning to the Life Sciences)

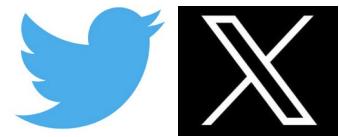
- To ask question please **raise your hand, unmute yourself and ask**. You can also ask questions in the zoom chat or email nikolay.oskolkov@osi.lv
- Please **keep your camera** on as much as possible for better contact and communication. Your face will not be recorded.
- The seminar is **focused on lectures**, however every session (except first one) will be accompanied by a ~20 mins practical (the notebooks are provided), where I will use html-versions of Rmarkdown and Jupyter notebooks to go through command lines with my explanations.
- The material is based on data and problems from **computational biology**, however the concepts discussed are general and can be applied for other types of data.

Machine Learning for Computational Biology

Nikolay Oskolkov, Lund University, NBIS SciLifeLab, Sweden

AFI course, 16.02.2026

Session 1: Introduction to Statistical Analysis in Computational Biology



@NikolayOskolkov



GitHub

github.com/NikolayOskolkov

Topics we'll cover in this session:

- 1) High-dimensional biological data and curse of dimensionality
- 2) Data-driven choice of statistical analysis
- 3) Basics of Frequentist statistics: pros and cons of p-value
- 4) LASSO regularization to overcome the curse of dimensionality
- 5) Basics of Bayesian statistics: priors as regularizations

Biological data are high dimensional

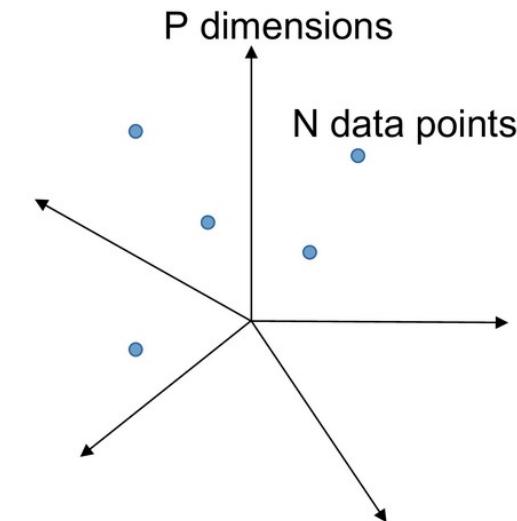
Statistical observations:
e.g. samples, cells etc.

Features: genes, proteins,
microbes, metabolites etc.

N

P

0	3	1	0	2	3	8	1	1	3
1	1	0	0	7	1	2	2	3	3
1	2	2	0	0	6	7	1	2	2
1	2	3	10	0	4	6	1	0	5
3	2	2	1	4	3	2	1	6	0
7	4	4	5	3	9	6	1	6	1
7	1	1	5	2	8	9	1	3	6
5	0	1	6	2	0	0	0	1	5
1	6	3	3	4	6	2	0	1	1
1	2	2	4	1	1	3	0	8	2



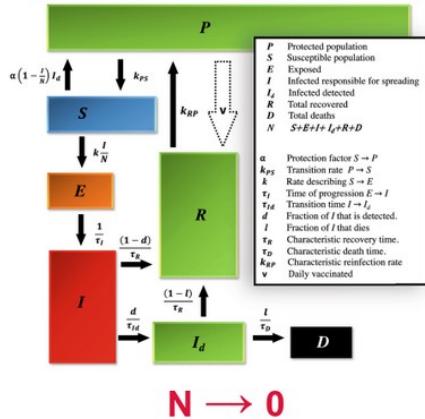
High Dimensional Data:
 $P \gg N$

For a robust statistical analysis, one should properly “sample” the P-dimensional space, hence large sample size is required, $N \gg P$

Types of data analysis

Biology / Biomedicine

Mathematical modeling

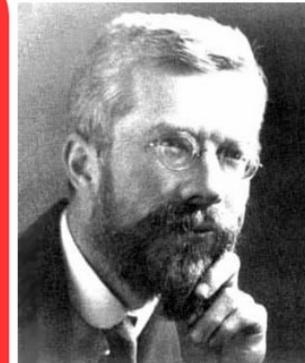


Bayesianism



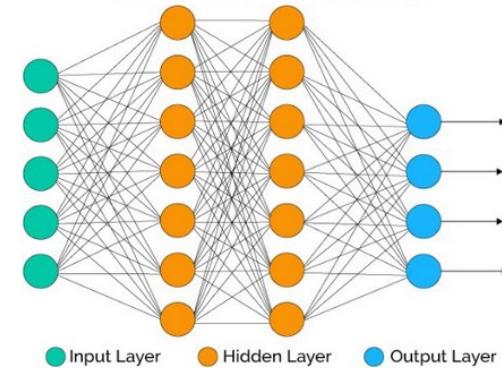
$N \ll P$

Frequentism



$N \approx P$

Machine Learning



$N >> P$

Hypothesis-driven

Data-driven

Amount of Data

The Curse of Dimensionality

Ex.1

$$Y = \alpha + \beta X$$

$$\beta = (X^T X)^{-1} X^T Y$$

$$(X^T X)^{-1} \sim \frac{1}{\det(X^T X)} \dots \rightarrow \infty, \quad n \ll p$$

$$\text{Ex.2} \quad E[\hat{\sigma}^2] = \frac{n-p}{n} \sigma^2$$

Biased Maximum Likelihood variance estimator at $n \ll p$

Some peculiarities of Frequentist statistics

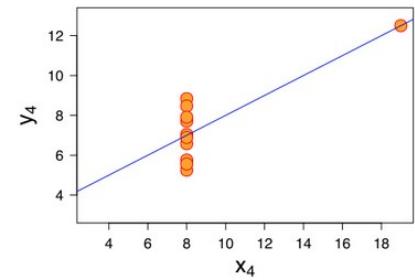
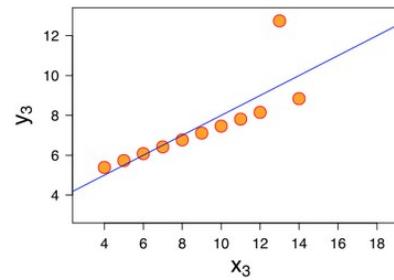
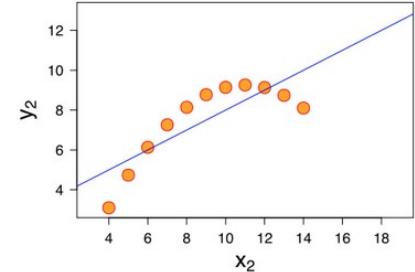
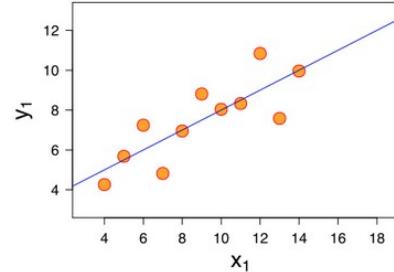
- based on Maximum Likelihood principle
- focus too much on summary statistics

$$L(x_i | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{\sum_{i=1}^N (x_i - \mu)^2}{2\sigma^2}}$$

$$\frac{\partial L(x_i | \mu, \sigma^2)}{\partial \mu} = 0; \quad \frac{\partial L(x_i | \mu, \sigma^2)}{\partial \sigma^2} = 0$$

$$\mu = \frac{1}{N} \sum_{i=0}^N x_i - \text{mean estimator}$$

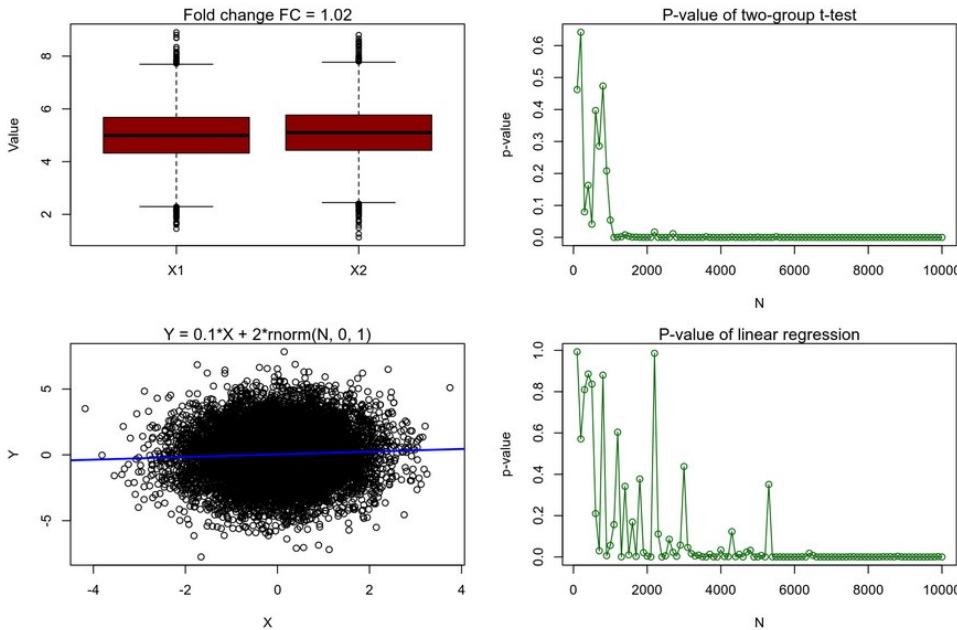
$$\sigma^2 = \frac{1}{N} \sum_{i=0}^N (x_i - \mu)^2 - \text{variance estimator}$$



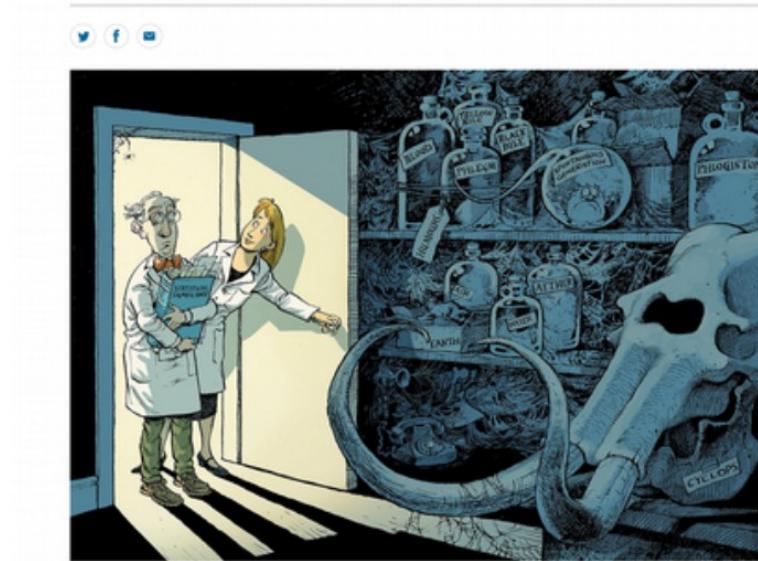
Summary statistics do not always reasonably describe data (example: Anscombes quartet)

Frequentist statistics: focus too much on p-values

```
1 FC<-1.02; x_mean<-5; x_sd<-1; N_vector<-seq(from=100,to=10000,by=100); pvalue_t<-vector(); pvalue_lm<-vector()
2 for(N in N_vector)
3 {
4   x1 <- rnorm(N, x_mean, x_sd); x2 <- rnorm(N, x_mean+FC, x_sd)
5   t_test_res<-t.test(x1, x2); pvalue_t <- append(pvalue_t, t_test_res$p.value)
6 
7   x <- rnorm(N, 0, 1); y <- 0.1*x+2*rnorm(N, 0, 1)
8   lm_res <- summary(lm(y~x)); pvalue_lm <- append(pvalue_lm, lm_res$coefficients[2,4])
9 }
10 par(mfrow=c(2,2)); par(mar = c(5, 5, 1, 1))
11 boxplot(x1, x2, names=c("X1", "X2"), ylab="Value", col="darkred"); mtext("Fold change FC = 1.02")
12 plot(pvalue_t~N_vector,type='o',xlab="N",ylab="p-value",col="darkgreen"); mtext("P-value of two-group t-test")
13 plot(y~x, xlab="X", ylab="Y"); abline(lm(y~x), col="blue", lwd=2); mtext("Y = 0.1*X + 2*rnorm(N, 0, 1)")
14 plot(pvalue_lm~N_vector,type='o',xlab="N",ylab="p-value",col="darkgreen"); mtext("P-value of linear regression")
```



A screenshot of a Nature journal article. The title is 'Scientists rise up against statistical significance'. It features a sub-headline: 'Valentin Amrhein, Sander Greenland, Blake McShane and more than 800 signatories call for an end to hyped claims and the dismissal of possibly crucial effects.' Below the article is a cartoon illustration of a scientist in a lab coat looking at a book while another scientist stands behind a counter filled with various bottles and containers.



Questionable whether p-value is a best metric for ranking features (biomarkers)

Frequentist statistics struggles with high-dimensional data

```
1 n <- 20 # number of samples
2 p <- 2 # number of features / dimensions
3 Y <- rnorm(n)
4 X <- matrix(rnorm(n * p), n, p)
5 summary(lm(Y ~ X))
```

Call:
lm(formula = Y ~ X)

Residuals:

	Min	1Q	Median	3Q	Max
	-2.0522	-0.6380	0.1451	0.3911	1.8829

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.14950	0.22949	0.651	0.523
X1	-0.09405	0.28245	-0.333	0.743
X2	-0.11919	0.24486	-0.487	0.633

Residual standard error: 1.017 on 17 degrees of freedom
Multiple R-squared: 0.02204, Adjusted R-squared: -0.09301
F-statistic: 0.1916 on 2 and 17 DF, p-value: 0.8274

```
1 n <- 20 # number of samples
2 p <- 10 # number of features / dimensions
3 Y <- rnorm(n)
4 X <- matrix(rnorm(n * p), n, p)
5 summary(lm(Y ~ X))
```

Call:
lm(formula = Y ~ X)

Residuals:

	Min	1Q	Median	3Q	Max
	-1.0255	-0.4320	0.1056	0.4493	1.0617

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.54916	0.26472	2.075	0.0679 .
X1	0.30013	0.21690	1.384	0.1998
X2	0.68053	0.27693	2.457	0.0363 *
X3	-0.10675	0.26010	-0.410	0.6911
X4	-0.21367	0.33690	-0.634	0.5417
X5	-0.19123	0.31881	-0.600	0.5634
X6	0.81074	0.25221	3.214	0.0106 *
X7	0.09634	0.24143	0.399	0.6992
X8	-0.29864	0.19004	-1.571	0.1505
X9	-0.78175	0.35408	-2.208	0.0546 .
X10	0.83736	0.36936	2.267	0.0496 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8692 on 9 degrees of freedom
Multiple R-squared: 0.6592, Adjusted R-squared: 0.2805
F-statistic: 1.741 on 10 and 9 DF, p-value: 0.2089

```
1 n <- 20 # number of samples
2 p <- 20 # number of features / dimensions
3 Y <- rnorm(n)
4 X <- matrix(rnorm(n * p), n, p)
5 summary(lm(Y ~ X))
```

Call:
lm(formula = Y ~ X)

Residuals:
ALL 20 residuals are 0: no residual degrees of freedom!

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.34889	NaN	NaN	NaN
X1	0.66218	NaN	NaN	NaN
X2	0.76212	NaN	NaN	NaN
X3	-1.35033	NaN	NaN	NaN
X4	-0.57487	NaN	NaN	NaN
X5	0.02142	NaN	NaN	NaN
X6	0.46290	NaN	NaN	NaN
X7	0.03313	NaN	NaN	NaN
X8	-0.31983	NaN	NaN	NaN
X9	-0.92833	NaN	NaN	NaN
X10	0.18091	NaN	NaN	NaN
X11	-1.37618	NaN	NaN	NaN
X12	2.11438	NaN	NaN	NaN
X13	-1.75103	NaN	NaN	NaN
X14	-1.55073	NaN	NaN	NaN
X15	0.01112	NaN	NaN	NaN
X16	-0.50943	NaN	NaN	NaN
X17	-0.47576	NaN	NaN	NaN
X18	0.31793	NaN	NaN	NaN
X19	1.43615	NaN	NaN	NaN
X20	NA	NA	NA	NA

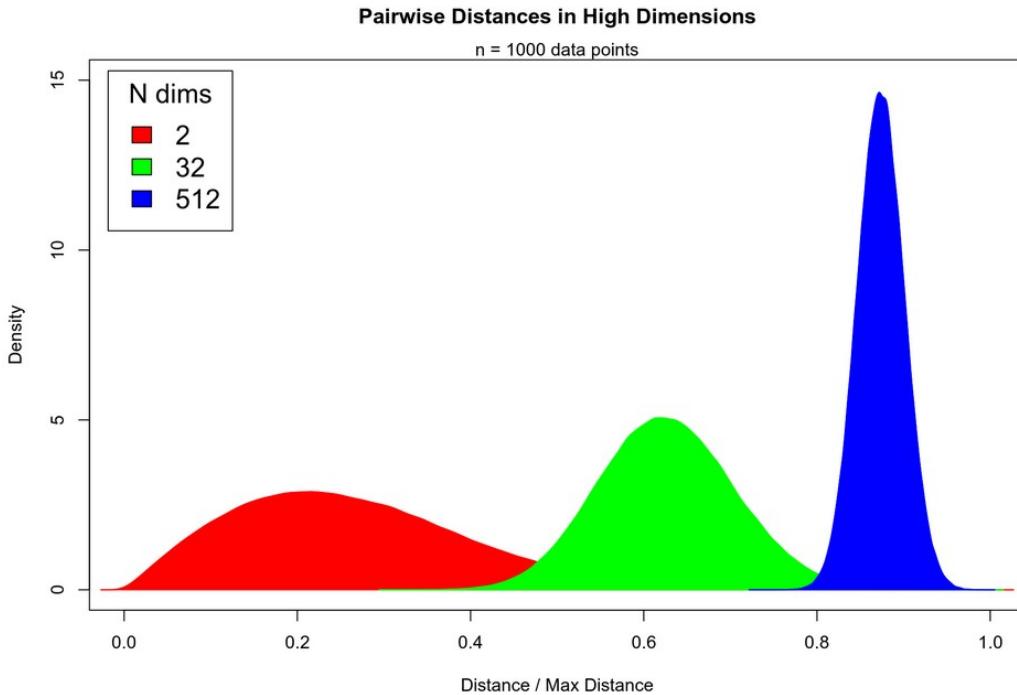
Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: NaN
F-statistic: NaN on 19 and 0 DF, p-value: NA

Going to higher dimensions →

Going to even higher dimensions →

Equidistant points in high dimensions

```
1 n <- 1000; p <- c(2, 32, 512); pair_dist <- list()
2 for(i in 1:length(p)) {
3   X <- matrix(rnorm(n * p[i]), n, p[i])
4   pair_dist[[i]] <- as.vector(dist(X));
5   pair_dist[[i]] <- pair_dist[[i]] / max(pair_dist[[i]])
6 }
```



- Data points in high dimensions:
 - move away from each other
 - become equidistant and similar
- Impossible to see differences between cases and controls

Regularizations: LASSO

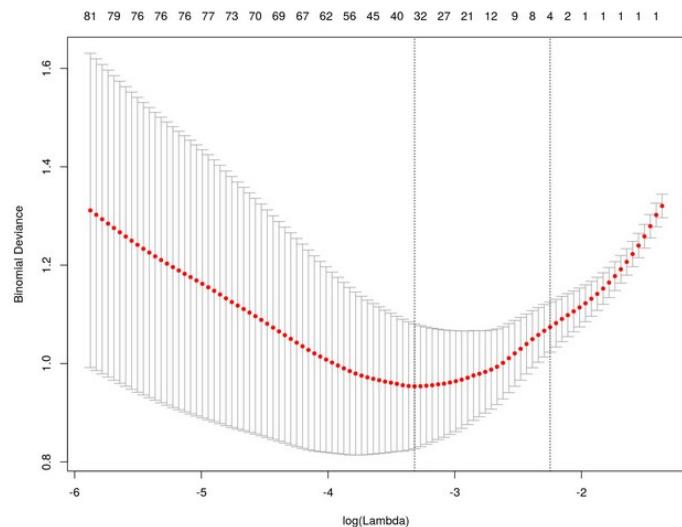
$$Y = \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

$$\text{OLS} = (Y - \beta_1 X_1 - \beta_2 X_2)^2$$

$$\text{Penalized OLS} = (Y - \beta_1 X_1 - \beta_2 X_2)^2 + \lambda(|\beta_1| + |\beta_2|)$$



Cross-validation is a standard way to tune model hyperparameters such as λ in LASSO



Regularizations are priors in Bayesian statistics

$$Y = \beta_1 X_1 + \beta_2 X_2 + \epsilon; \quad Y \sim N(\beta_1 X_1 + \beta_2 X_2, \sigma^2) \equiv L(Y | \beta_1, \beta_2)$$

- Maximum Likelihood principle: maximize probability to observe data given parameters:

$$L(Y | \beta_1, \beta_2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(Y - \beta_1 X_1 - \beta_2 X_2)^2}{2\sigma^2}}$$

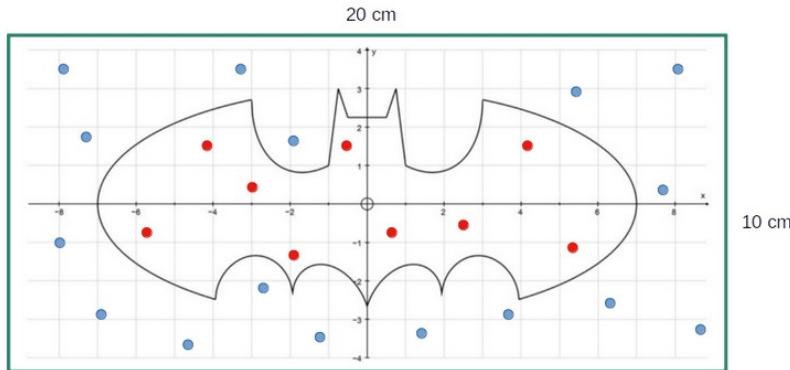
- Bayes theorem: maximize posterior probability of observing parameters given data:

$$\text{Posterior}(\text{params} | \text{data}) = \frac{L(\text{data} | \text{params}) * \text{Prior}(\text{params})}{\int L(\text{data} | \text{params}) * \text{Prior}(\text{params}) d(\text{params})}$$

$$\begin{aligned} \text{Posterior}(\beta_1, \beta_2 | Y) &\sim L(Y | \beta_1, \beta_2) * \text{Prior}(\beta_1, \beta_2) \sim \exp^{-\frac{(Y - \beta_1 X_1 - \beta_2 X_2)^2}{2\sigma^2}} * \exp^{-\lambda(|\beta_1| + |\beta_2|)} \\ -\log [\text{Posterior}(\beta_1, \beta_2 | Y)] &\sim (Y - \beta_1 X_1 - \beta_2 X_2)^2 + \lambda(|\beta_1| + |\beta_2|) \end{aligned}$$

Markov Chain Monte Carlo (MCMC): introduction

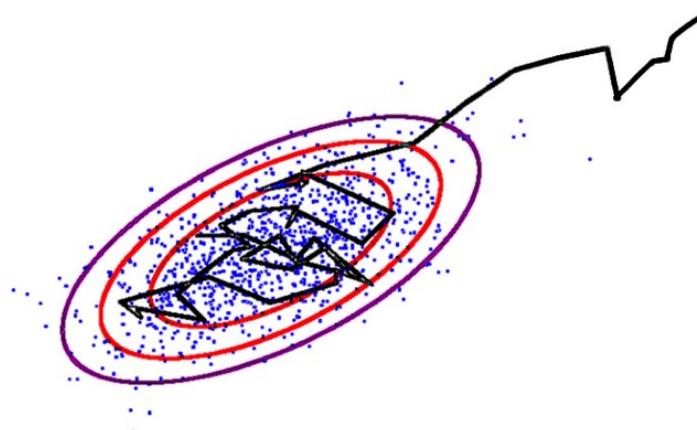
- Integration via Monte Carlo sampling
- Markov Chain Monte Carlo (MCMC)



$$I = 2 \int_{-2}^4 x dx = 2 \frac{x^2}{2} \Big|_2^4 = 16 - 4 = 12$$

```
1 f <- function(x){return(2*x)}; a <- -2; b <- 4; N <- 10000; count <- 0
2 x <- seq(from = a, to = b, by = (b-a) / N); y_max <- max(f(x))
3 for(i in 1:N)
4 {
5   x_sample <- runif(1, a, b); y_sample <- runif(1, 0, y_max)
6   if(y_sample <= f(x_sample)){count <- count + 1}
7 }
8 paste0("Integral by Monte Carlo: I = ", (count / N) * (b - a) * y_max)
```

```
[1] "Integral by Monte Carlo: I = 11.9248"
```



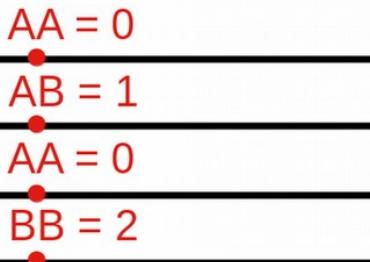
$$\text{Hastings ratio} = \frac{\text{Posterior}(\text{params}_{\text{next}} | \text{data})}{\text{Posterior}(\text{params}_{\text{previous}} | \text{data})}$$

- If Hastings ratio $> u [0, 1]$, then accept, else reject
- Hastings ratio does not contain the intractable integral from Bayes theorem

Markov Chain Monte Carlo (MCMC) from scratch in R

- Example from population genetics

SNP = A / B



f – frequency of allele B; $g = 0, 1, 2$

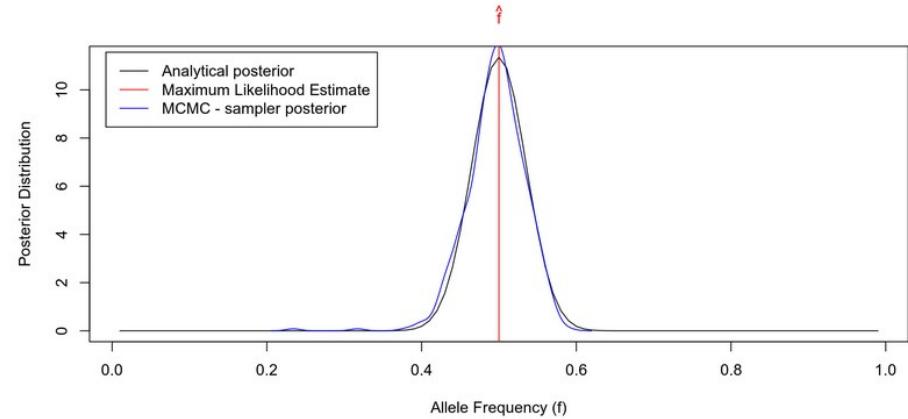
$n_g = n_0, n_1, n_2$ – genotype carriers

$$L(n | f) = \prod_g \left[\binom{2}{g} f^g (1-f)^{2-g} \right]^{n_g}$$

$$\frac{\partial \log[L(n|f)]}{\partial f} = 0 \Rightarrow \hat{f} = \frac{n_1 + 2n_2}{2(n_0 + n_1 + n_2)}$$

$$\text{Prior}(f, \alpha, \beta) = \frac{1}{B(\alpha, \beta)} f^{\alpha-1} (1-f)^{\beta-1}$$

```
1 N <- 100; n <- c(25, 50, 25) # Observed genotype data for N individuals
2 f_MLE <- (n[2] + 2*n[3]) / (2 * sum(n)) # MLE of allele frequency
3
4 # Define log-likelihood function (log-binomial distribution)
5 LL <- function(n, f){return((n[2] + 2*n[3])*log(f) + (n[2] + 2*n[1])*log(1-f))}
6 # Define log-prior function (log-beta distribution)
7 LP <- function(f, alpha, beta){return(dbeta(f, alpha, beta, log = TRUE))}
8
9 # Run MCMC Metropolis - Hastings sampler
10 f_poster <- vector(); alpha <- 0.5; beta <- 0.5; f_cur <- 0.1 # initialization
11 for(i in 1:1000)
12 {
13   f_next <- abs(rnorm(1, f_cur, 0.1)) # make random step for allele frequency
14
15   LL_cur <- LL(n, f_cur); LL_next <- LL(n, f_next)
16   LP_cur <- LP(f_cur, alpha, beta); LP_next <- LP(f_next, alpha, beta)
17   hastings_ratio <- LL_next + LP_next - LL_cur - LP_cur
18
19   if(hastings_ratio > log(runif(1))){f_cur <- f_next}; f_poster[i] <- f_cur
20 }
```



Take home messages of the session:

- 1) Biological data are high dimensional and this should be taken into account when performing statistical analyses
- 2) The choice of analysis (e.g. Frequentist, Bayesian or machine learning) is driven by particular data type
- 3) LASSO and Bayesian statistics are recommended for analysis of high-dimensional biological data
- 4) Penalized regression can be viewed as a bridge from traditional statistics to machine learning



*Knut och Alice
Wallenbergs
Stiftelse*



**LUNDS
UNIVERSITET**