V3 – missing values + batch effect correction

- How can one deal with missing values?
- What are Batch Effects?
- **ComBat** tool
- **BEclear** tool applies latent factor model to predict missing values and to remove batch effects
  - DNA microarray
  - DNA methylation
- Functional Normalization (**FunNorm**) tool
- Some Review of Probability Theory Basics

In this lecture, we deal with the issue of reconstructing missing values in our data set and with the problem of batch effects in the data set.

We will discuss the principles of two tools, ComBat and FunNorm that are widely used for removing batch effects.

Then we will also look at the tool BEclear from our group.

At the end I have summarized some basics from probability theory that are worth browsing over.

Process *S. aureus* microarray data – part II

StaphyType Test Report

Operator	
Sample ID	2192119
Experiment ID	2192119 - {4083AD2C-7D42-4FB9-82D5-E50CC0FD6206}
Date of Result	Thu Apr 14 10:46:01 2011
Assay Name	StaphyType
Assay ID	10248
Well Position	01 (01-A)
Software Version	2009-07-09
Device	04a0022

Internal Controls

Data Quality	passed

Genetic markers for S. aureus / MRSA / PVL

Taxonomy	Species Marker (*S.aureus*) positive
MRSA (mecA)	positive
PVL	negative

Resistance Genotype

Hybridisation (Gene)	Result	Expected Resistance
mecA	positive	Methicillin, Oxacillin and all Beta-Lactams, defining MRSA
blaZ	negative	Beta-Laktamase
ermA	positive	Macrolide, Lincosamide, Streptogramin
ermB	negative	Macrolide, Lincosamide, Streptogramin
ermC	negative	Macrolide, Lincosamide, Streptogramin
linA	negative	Lincosamides

	11	46	10	33	28
MRSA (mecA)	0	0	0	0	0
PVL	0	0	0	0	0
23S-rRNA	1	1	1	1	1
gapA	1	1	1	1	1
katA	1	1	1	1	1
coA	1	0	1	1	1
Protein A	1	1	1	1	1
sbi	1	1	1	1	1
nuc	1	1	1	1	1
fnbA	1	1	1	1	1
vraS	1	1	1	1	1
sarA	1	1	1	1	1
eno	1	1	1	1	1
saeS	1	1	1	1	1
mecA	0	0	0	0	0
blaZ	0	1	0	0	0
blaI	0	1	0	0	0
blaR	0	1	0	0	0
ermA	0	0	0	0	0
ermB	0	0	0	0	0
ermC	0	0	0	0	0
linA	0	0	0	0	0

Compute Euclidian distance between samples

$$\|a - b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$$

First, we will look again at the microarray data set that we discussed in the first lecture.

Ambiguous values

In the *S. aureus* genotyping test report, individual markers can be "**positive**" or "**negative**" and also "**ambiguous**".

Such ambiguous classifications can be caused by:

- poor sample quality, or
- poor signal quality, or
- by the presence of plasmids in low copy numbers.

www.alere-technologies.com

The image-reader device generates 3 sorts of output „positive" (dark circle), „negative" (white field), and „ambiguous" for fields that cannot be precisely determined.

There are various possible reasons why about 5% of the microarray tests yielded ambiguous densities.

Re-Assign ambiguous values in DNA microarray

Task – predict ambiguous values.

Simple idea: **baseline prediction** using average values

total average

$$\mu = \frac{1}{N} \sum_{(i,j)\in\Omega} D_{ij}$$

sample average

$$b_i = \frac{1}{N_i} \sum_{(i,j)\in\Omega} D_{ij} - \mu$$

gene average

$$b_j = \frac{1}{N_j} \sum_{(i,j)\in\Omega} D_{ij} - \mu$$

$$b_{prediction} = \frac{1}{3}(\mu + b_i + b_j)$$

replace small fraction of known values by (thresholded) baseline values -> ~85% correct predictions

Better results are obtained with:

Latent Factor Model (LFM)

~95% correct predictions

In the large scale project discussed in the first lecture, ambiguous values are disturbing the process of data analysis. They need to be cleaned up and replaced by either „positive“ or „negative“ values.

A simple approach would be to replace them either by the average signal of the data points for this particular gene probe („gene average“), or by the average of the data points in this particular sample („sample average“) or by the average value of the full data matrix.

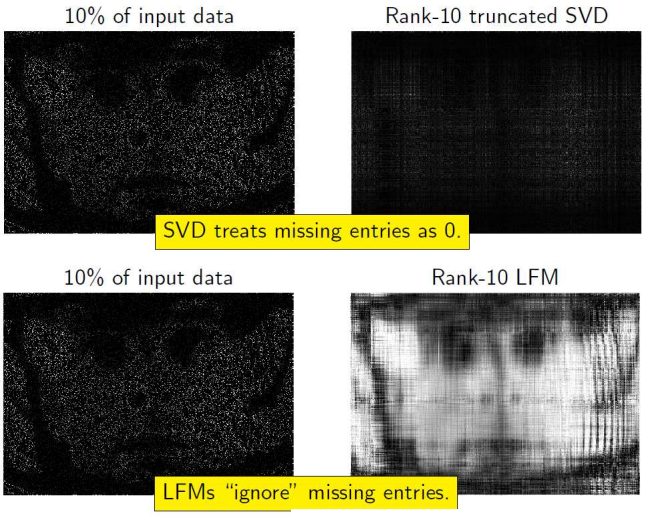
One could even compute the average of these 3 averages –> b_prediction.

Because we can only deal with 0 or 1 entries, the computed averages need to be thresholded by a suitable value, e.g. 0.5. Averages below 0.5 would be set to 0, those above 0.5 to 1.

We tested how well this works for some randomly selected data points. If we regenerate their entries and compare them to the correct values, this gives an agreement of 85% which is much better than random (50%).

We will now introduce a method that uses latent factor models that even generates predictions that are about 95% correct.

Latent Factor Models in image reconstruction



DMM course by R. Gemulla and P. Miettinen

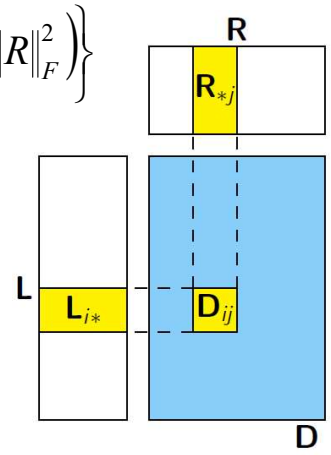
Latent Factor Models are very successful in image reconstruction.

If we delete 90% of the data points, the upper row shows that SVD is not useful for reconstructing the missing values.

However, a LFM can recover enough contrast so that we can recognize the face in this picture.

LFM: mathematical background

$$\min_{l^*, r^*} \left\{ \sum_{(i,j)\in\Omega} \left(D_{ij} - [LR]_{ij}\right)^2 + \lambda\left(\|L\|_F^2 + \|R\|_F^2\right) \right\}$$



L ($m \times r$) and R ($r \times n$) are sought matrices of rank r

D ($m \times n$) is the given matrix

Approach is termed **regularized least squares**: regularization limits the size of coefficients in the least squares method.

Idea: construct L and R from known data; use them to reconstruct the missing data.

This slide illustrates the principles of LFM.

The idea is to represent the data matrix D_{ij} as the product of two matrices L and R. Once L and R are found, they can be used to compute all missing data points.

The algorithm iteratively refines guesses for L and R so that the squared difference of their product from the known data points is minimal.

Since this problem is usually underdetermined, there would be many different equally good solutions.

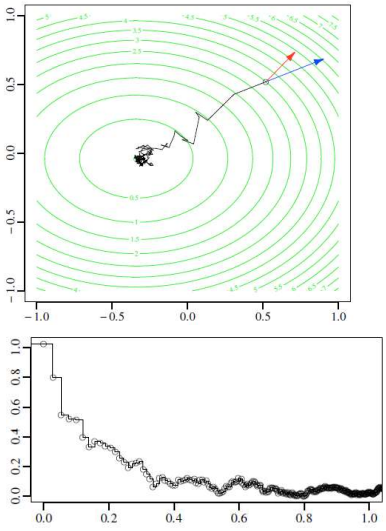
Therefore, one also applies the principle of regularization meaning that the algorithm constructs L and R in a way so that their norm is minimal.

A parameter lambda controls the balance between the two terms.

LFM: solve by stochastic gradient descent

- Pick a random entry;
- Compute approximate gradient;
- Update parameters *L* and *R*
- Repeat *N* times.

We implemented LFM-completion of missing values in the Bioconductor package **BEclear**.



Akulenko, R., Merl, M., Helms, V. (2016) PLoS ONE, 11:e0159921

Koren, Yehuda, Robert Bell, and Chris Volinsky. 2009. "Matrix Factorization Techniques for Recommender Systems." Computer 42 (8):30–37.

BEclear implements a stochastic gradient descent algorithm following a classic paper by Koren et al. This paper has been cited more than 9000 times.

It mentions that there are two popular approaches to solve the minimization task, stochastic gradient descent and alternating least squares (ALS).

Gradient descent is a well-known algorithm for optimization. An initial guess is iteratively refined by taking small steps along the direction of steepest descent which is the direction where the negative of the first derivative of the objective function is largest.

In **stochastic gradient descent**, the actual gradient that is calculated from the entire data set is replaced by an estimate of the gradient that is calculated from a randomly selected subset of the data.

MA assignment to clonal complexes + LFM predictions confirmed by WGS

154 *S. aureus* isolates (182 target genes) from Germany-vs-Africa study

Table 1A

Result Category		Result caused by		Functional Category of genes: Identification	Regulation	Resistance	Virulence	Total	% Total
Concordant	Positive	Microarray and WGS (*de novo*)		829	990	1,060	8,495	11,374	40.6%
n=27,119	Negative	Microarray and WGS (*de novo*)		0	1,159	8,100	6,486	15,745	56.2%
(96.8 %)									
Discrepant	False Positive	Microarray	Mishybridizations	0	78	21	103	202	0.7%
n=909 (3.2 %)		LFM	Misprediction	0	17	2	9	28	0.1%
	False Negative	Microarray	Polymorphisms	0	3	14	140	157	0.6%
		LFM	Misprediction	0	0	0	5	5	< 0.1%
		WGS	Assembly error	88	42	16	164	310	1.1%
			Cropped contig	1	12	15	28	56	0.2%
			Not sequenced or aberrant allele	6	9	8	100	123	0.4%
	Unknown			0	0	4	24	28	0.1%
		Total number of typing results		924	2,310	9,235	15,554	28,028	100%

Very few errors due to LFM mis-predictions.

Strauss et al. J Clin Microbiol (2016)

In this comparison that was already shown in the first lecture, we were using data for 334 probe IDs from 154 isolates.

Out of this data, n = 2,788 or 5.4% of the hybridization signals were assigned as ambiguous value.

As just described, ambiguous were replaced by 1 or 0 values according to an LFM prediction based on the entries in neighboring fields of the involved columns and rows.

First, the accuracy of this approach was tested by a bootstrap approach as follows: 5% of randomly selected entries that were known to be positive or negative were removed from the dataset. This fraction corresponds to the typical number of targets typed as ambiguous in the microarray experiments. Then, these missing entries were predicted using LFM and were compared to the original values. As a result, LFM yielded an accuracy of 97% against the original values. Thus, the error rate of predicted values can be estimated as about 3%.

By comparison to the WGS data, LFM predictions were actually only wrong in 33 or about 0.1 % of the cases.

Batch effects

Batch effects are:

Subgroups of measurements that show **qualitatively different behavior** across conditions and are unrelated to the biological or scientific variables in a study.

For a **microarray experiment**, batch effects may occur due to:

- Chip type/lot/platform
- Different laboratories may have different **standard operating procedures**
- Sample/preservation **protocols** (procedures of drawing biological samples may vary from center to center and over time within center, relevant to retrospective studies)
- Storage/shipment conditions
- RNA isolation (different laboratories may use different extraction procedures or kits, and different lots of reagents may perform differently)
- cRNA/cDNA synthesis
- Amplification/labeling/hybridization protocol (different reagents or lots may be used)
- Wash conditions (temperature, ionic strength, fluidics modules/stations; cleaning schedules)
- Ambient conditions during sample preparation/handling, such as room temperature and ozone levels
- Scanner (types, settings, calibration drift over prolonged studies; scheduled maintenance)

Luo et al. Pharmacogenomics J. (2010) 10: 278–291.

Now we come to the detection of batch effects.

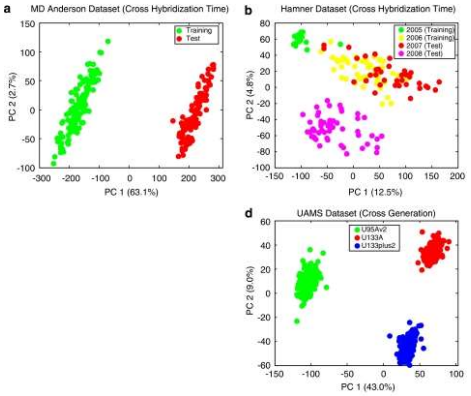
A batch effect describes a case when a subgroup of measurements in the data set shows a qualitatively different behavior from the rest of the data.

Listed here are possible reasons why batch effects may occur.

This is a link to this paper: https://www.nature.com/articles/tpj201057

Another typical **error source** is mislabeling of samples either as healthy or as tumor. But this introduces random noise, not systematic deviations.

Example: batch effects in public MA data sets



Score plot of the first 2 principal components.

Batches (groups) are indicated by colors.

(a) MD Anderson breast cancer data set. 230 samples from stage I–III breast cancers were split into training/test according to hybridization dates. The first 130 samples assayed were used as "training" set and the remaining 100 samples as "test" set.

(b) Hamner lung carcinogen data set. 2 batches in training set hybridized in 2005 and 2006, and 2 batches in test set hybridized in 2007 and 2008.

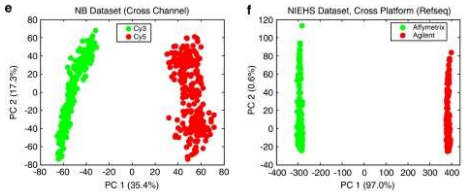
(d) UAMS multiple myeloma data set. Here, the 3 batches represent 3 generations of Affymetrix chips for human genes.

Luo et al. Pharmacogenomics J. (2010) 10: 278–291.

In these examples taken from the literature, significant batch effects can be seen by the perfect separation of different batches on the PCA score plots.

For the Hamner data set (panel B), batch effects exist with overlaps between several batches.

Batch effects in public MA data sets



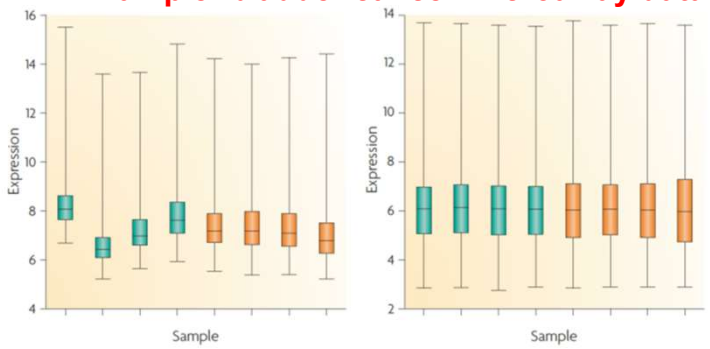
(**e**) Cologne neuroblastoma data set. The 2 batches represent the 2 channels of Agilent arrays. Cy3 and Cy5 are two different fluorescent dye molecules.

(**f**) NIEHS data set (cross-platform): the two groups represent Affymetrix and Agilent microarray platforms.

Luo et al. Pharmacogenomics J. (2010) 10: 278–291.

These two plots show batch effects due to using different fluorescent dyes and due to using different microarray platforms.

Example: bladder cancer microarray data



Raw data for normal samples taken from a bladder cancer microarray data set (Affymetrix chip).

Green and orange represent two different processing dates. Box plot of raw gene expression data ($\log_2$ values)

Same data after processing with RMA, a widely used preprocessing algorithm for Affymetrix data.

RMA applies **quantile normalization** — a technique that forces the distribution of the raw signal intensities from the microarray data to be the same in all samples.

Leek et al. Nature Rev. Genet. 11, 733 (2010)

The left plot shows a box plot of microarray data. Each line represents the expression of all genes in one sample.

Obviously, the medians are very different. The left sample is highest.

The right plot shows the same data after RMA normalization. This algorithm uses quantile normalization.

Now, the distributions are very similar to each other.

Quantile normalisation: adjusts multiple distributions

Given: 3 measurements of 4 variables A – D.

Aim: all measurements should get identical distributions of values

Original data

A	5	4	3
B	2	1	4
C	3	4	6
D	4	2	8

→

Determine in each column the rank of each value

A	iv	iii	i
B	i	i	ii
C	ii	iii	iii
D	iii	ii	iv

Sort columns by magnitude

A	2	1	3
B	3	2	4
C	4	4	6
D	5	4	8

→

Compute mean of each row

A	2	Rank i
B	3	Rank ii
C	4.67	Rank iii
D	5.67	Rank iv

A	5.67	4.67	2
B	2	2	3
C	3	4.67	4.67
D	4.67	3	5.67

Replace original values by mean values according to the rank of the data field.

Now all columns contain the same values (except of duplicates) so that they can be easily compared.

This slide reviews the quantile normalization method.

All data points are replaced by row averages so that the distributions become identical (except of duplicates).

Methods to correct batch effects

Available batch effect removal methods can be classified in 2 main approaches: location-scale methods and matrix factorization methods.

The **location-scale methods** assume a model for the data distribution within batches, and adjust the data within each batch to fit this model.

This approach is the most straight-forward one and many methods have been proposed: ratio-based methods, ComBat, quantile based methods, mean or median centering etc.

The **matrix factorization based methods** assume that the gene-by-sample expression matrix can be represented by a small set of rank-one components which can be estimated by means of matrix factorization.

The components that correlate with the batch number are then removed to obtain the normalized dataset

Emilie Renard, P.-A. Absil *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1511-1518, 2017

This is an overview over existing methods for removing batch effects.

Global methods to correct batch effects

Mean-centering : after the transformation, the mean of each feature across all the samples within each batch is set to zero.

Standardization: Beyond mean-centering, this approach normalizes the standard deviation of all features across samples within each batch to unity.

The combination of these 2 yields the z-score of feature *i*: $\frac{x_{ij}-\mu_i}{\sigma_i}$

Ratio-based: All samples are scaled by a reference array.

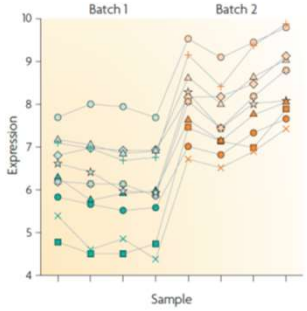
This can be the average of multiple reference arrays, such as the measurement of universal human reference RNA samples for clinical data and vehicle control samples for toxicogenomics data.

Such global **normalization** methods do not remove batch effects if these affect specific subsets of genes so that different genes are affected in different ways.

Luo et al. Pharmacogenomics J. (2010) 10: 278–291.

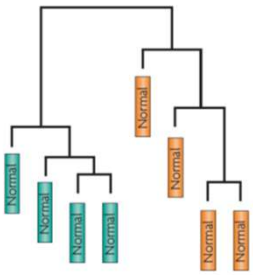
Listed here are three global methods that correct all data entries.

Example: same bladder cancer microarray data



10 particular genes that are susceptible to batch effects even after RMA normalization.

Hundreds of other genes show similar behavior but, for clarity, are not shown.



Clustering of samples after normalization.

The samples perfectly cluster by processing date.

→ clear evidence of **batch effect**

Processing date is likely a "**surrogate**" for other variations (laboratory temperature, quality of reagents etc.).

Leek et al. Nature Rev. Genet. 11, 733 (2010)

This is again the microarray data set that was normalized by RMA.

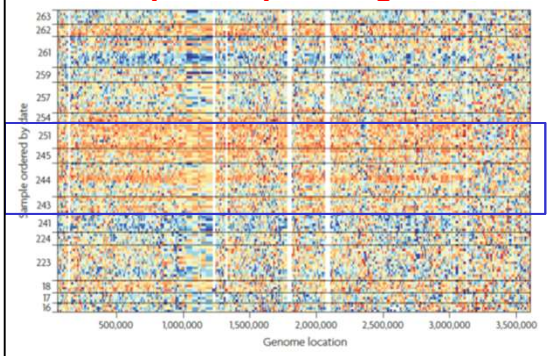
Although the overall distributions of the samples have been homogenized, there are hundreds of genes left that show clear batch effects.

Note that this plot shows the expression of individual genes.

If one clusters this normalized data (see right plot), the samples cluster according to processing date (green and orange represent two different dates).

This indicates that **RMA did not manage to remove the batch effect** for these genes.

Example: sequencing data from 1000 Genomes project



Coverage data (number of mapped reads in 10 kb windows) were standardized across samples: **blue** represents three standard deviations below average and **orange** represents three standard deviations above average.

Each row is a different sample processed in the same facility with the same platform. The samples are ordered by processing date with horizontal lines dividing the different dates. Shown is a 3.5 Mb region from chromosome 16.

Various batch effects of the read coverage can be observed. The largest one occurs between days 243 and 251 (the large orange horizontal streak).

Leek et al. Nature Rev. Genet. 11, 733 (2010)

This is another example for a large-scale batch effect in a famous genomic project.

For some reason, sequencing in the 1000 genome project generated higher read coverage during days 243 and 251.

ComBat

A widely used location-scale method is ComBat.

Here, the expression value of gene *i* for sample *j* in batch *b* is modeled as

$$X_{bij} = \alpha_i + \beta_i C_j + \gamma_{bi} + \delta_{bi}\varepsilon_{bij}$$

where α_i is the overall gene expression, and C_j is the vector of known covariates representing the sample conditions (such as batch membership).

The error term ε_{bij} is assumed to follow a normal distribution $N(0,\sigma^2_i)$.

Additive and multiplicative batch effects are represented by parameters γ_{bi} and δ_{bi}.

ComBat uses a Bayesian approach to model the different parameters, and then removes the batch effects from the data to obtain the clean data:

$$X^*_{bij} = \hat{\alpha}_i + \hat{\beta}_i C_j + \hat{\varepsilon}_{bij}$$

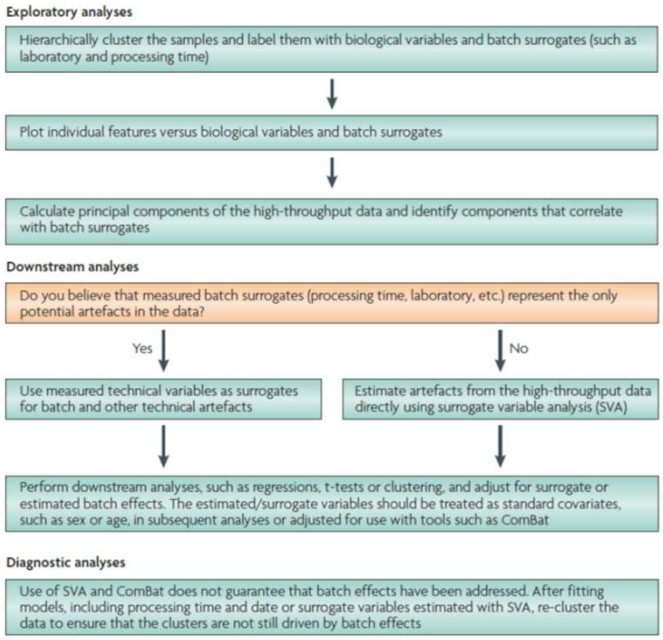
Emilie Renard, P.-A. Absil *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1511-1518, 2017

See also discussion of Combat in Y.Zhang et al. BMC Bioinformatics 19, 262 (2018)

A widely used tool for removing batch effects is ComBat. It is a location-scale method. The slide explains the basic principles of ComBat.

Although it is widely used, experience has shown that ComBat also has caveats. The listed Zhang paper (https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2263-6) discusses some of them. For example, ComBat removes batch effects impacting both the means and variances of each gene across the batches. However, in some cases, the data might require a less (or more) extreme batch adjustment. Also, ComBat suffers from sample 'set bias', meaning that if samples or batches are added to or removed from the set of samples on hand, the batch adjustment must be reapplied, and the adjusted values will be different–even for the samples that remained in the dataset in all scenarios.

Workflow to identify and adjust batch effects



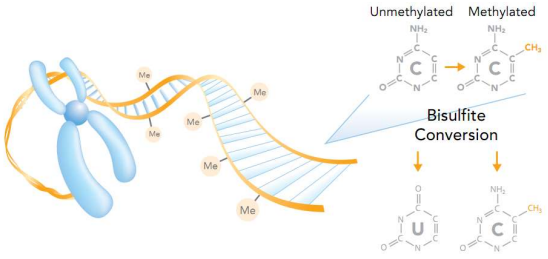
Leek et al. Nature Rev. Genet. 11, 733 (2010)

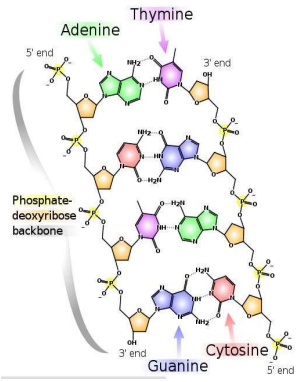
After a high-throughput study has been performed, the statistical approach for dealing with batch effects consists of two key steps.

Exploratory analyses must be carried out to identify the existence of batch effects and quantify their effect, as well as the effect of other technical artefacts in the data.

Downstream statistical analyses must then be adjusted to account for these unwanted effects.

Correcting batch effects in DNA methylation data





Infinium HumanMethylation27, RevB BeadChip Kits

BEclear: Akulenko, Merl, Helms (2016) PLOS ONE, v11, e0159921.

Breast cancer DNA methylation data: Akulenko and Helms (2013) Human Molecular Genetics, 22, 3016-3022

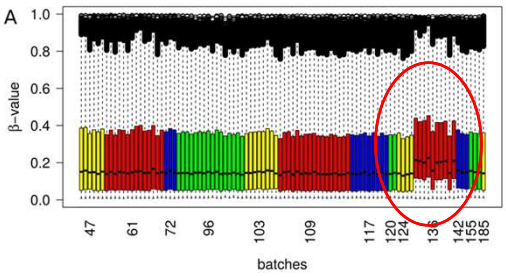
In late 2011, we started working with DNA methylation data from the TCGA breast cancer study.

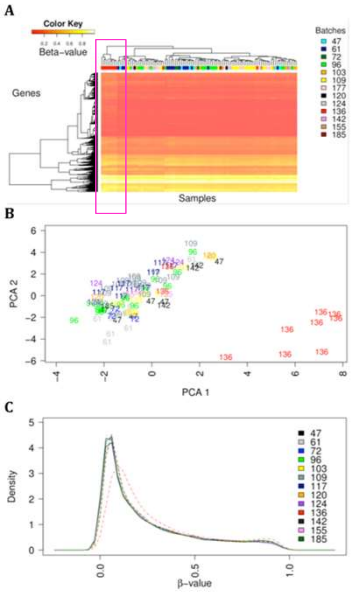
Soon we detected a severe batch effect that only affected segments on the Illumina chips.

In our 2013 publication, we omitted all affected genes (ca. 25% of the data).

Later, we developed a method termed BEclear (stands for clearing of batch effects) and published that tool in 2016.

Original DNA methylation data for breast cancer (TCGA)





β: fraction of methylated cytosines in CpG

Clear batch effect in batch 136

Left: box-plot

Right/top: hierarchical clustering

Right/middle: PCA

Right/bottom: density distribution

This is an example of exploratory analysis. The top left panel shows a boxplot of the DNA methylation data in different batches of the TCGA data set.

The top right panel shows hierarchical clustering of the same data. The middle right panel shows a PCA of the same data. The bottom right panel shows a density distribution plot.

All plots illustrate clearly that, in batch 136, the distribution of β-values of genes is shifted to larger values than in the other batches.

The per sample plot (top left) shows that the difference in batch 136 is not due to only one sample but exists in all but two samples from this batch.

Beclear: Identify batch effected genes

(1) Compare the distribution of every gene in one batch to its distribution in all other batches using the nonparametric Kolmogorov-Smirnov (KS) test.

P-values are corrected by False Discovery Rate.

(2) To consider only biologically relevant differences in methylation levels, identify the absolute difference between the median of all β-values within a batch for a specific gene and the respective median of the same gene in all other batches.

Beta-values range between 0 and 1. The exp. error was estimated as 5%.

-> Smaller variations are not considered meaningful.

Therefore, only those genes that have a FDR-corrected significance p-value below 0.01 (KS-test) AND a median difference larger than 0.05 are considered as batch effected (BE) genes in a specific batch or sample.

We suspected that the batch effect of the analyzed data affected various genes on the chip in different ways.

Therefore, we first had to identify which genes contain data points that differ largely from the remaining data points.

Beclear: score the severeness of batch effect for each batch

(3) Score severeness of batch effect in single batches by a heuristic weighting-scheme :

$$BEscore = \frac{\sum_{i \in mdif_{cat}} (N_{BEgenes_i} \cdot w_i)}{N}$$

N : total number of genes in a current batch,
$mdif_{cat}$: category of median differences $\in [0,1]$
$N_{BEgenes\_i}$: # BE-genes in *mdif* category *i*
w_i : weight of *mdif* category *i*
Weight categories:
if *mdif* < 0.05, then weight = 0;
if 0.05 ≤ *mdif* < 0.1 weight = 1;
if mdif $\in$ $[m \times 0.1 \leq$ mdif $< (m+1) \times 0.1]$, $m \in N$, $m \leq 9$
weight = $2 \times m$

Scoring scheme considers number of BE-genes in the batch and magnitude of deviation of the medians of BE-genes in one batch compared to all other batches.

Based on the BE-scores of all batches, identify using the Dixon test which batches have BE-scores that deviate significantly from the BE-scores of the other batches.

All BE-gene entries in these affected batches are **replaced** by **LFM predictions (see p.6 in V3)**.

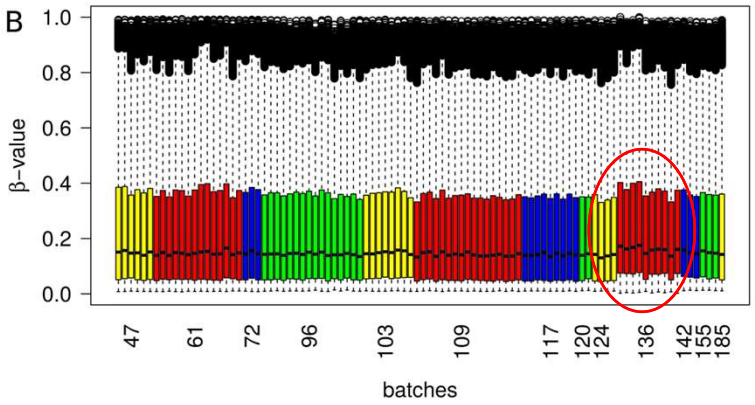
Each batch is assigned a BEscore value that considers the number of BE genes in that batch and the magnitude of their batch effects.

The question was now which values should be replaced: only the individual data points of BE genes in this batch or all of them?

We reasoned that if a sample (or batch) has a BEscore that is significantly larger than the other BEscore values, all values of that sample (or batch) should be replaced by LFM predictions.

Comparison of BEscores is done using the tabulated Dixon test. This test considers the absolute difference (gap) between the outlier in question and the closest value to it relative to the range of values (max – min).

TCGA data for breast cancer – batch affected entries predicted by LFM/BEclear

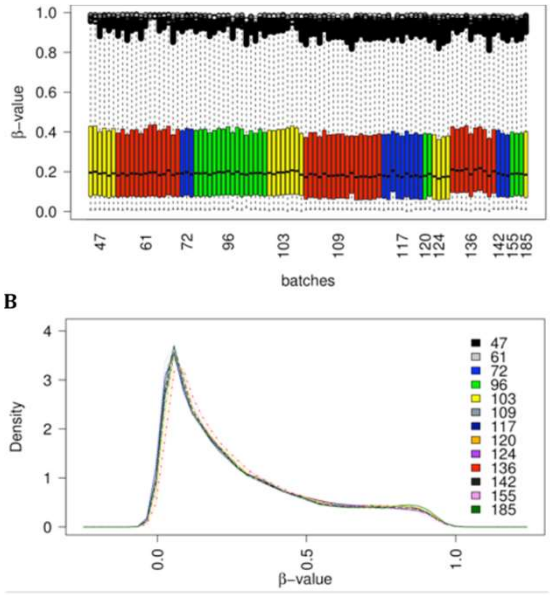


Batch 136 has still slightly larger values than other batches,

but the deviation is no longer statistically significant.

This figure shows the outcome of BEclear for the tumor data.

TCGA data for breast cancer – data corrected by FunNorm



B

A. Per sample boxplot

B. Density plot.

Functional normalization was able to adjust the batch effect equally well as BEclear

This figure shows the normalization result by the tool FunNorm, another tool.

Functional Normalization

Functional normalization uses information from 848 **control probes** on 450k array.

The method extends the idea of **quantile normalization** by adjusting for known covariates measuring unwanted variation.

Consider $\mathbf{Y}_1,\ldots,\mathbf{Y}_n$ high-dimensional vectors each associated with a set of scalar **covariates** $Z_{i,j}$ whereby $i = 1,\ldots,n$ is the index of the samples and $j = 1,\ldots,m$ is the index of the covariates.

Ideally these known covariates are associated with unwanted variation and unassociated with biological variation.

Functional normalization attempts to remove their influence.

Fortin et al. Genome Biol. 15, 503 (2014)

FunNorm builds on the idea of quantile normalization. This is a link to the paper: https://genomebiology.biomedcentral.com/articles/10.1186/s13059-014-0503-2

It is particularly tailored to the Illumina 450k chip that detects methylation levels for 450.000 CpG sites in the human genome.

It also contains close to 1000 control probes that do not measure CpG methylation of the sample, but are used to test the correctness of the biochemical processing steps carried out.

Functional Normalization

For each high-dimensional observation $\mathbf{Y}_i$, we form the empirical **quantile function** $r \in [0,1]$ for its marginal distribution, and denote it by q_i^{emp} .

What is a **quantile function**?

The k-th **percentile** of a set of values divides them so that k % of the values lie below and (100−k)% of the values lie above.
- The 25th percentile is known as the lower quartile.
- The 50th percentile is known as the median.
- The 75th percentile is known as the upper quartile.

It is more common in statistics to refer to **quantiles**.
These are the same as percentiles, but are indexed by sample fractions rather than by sample percentages.

The data sets Y to be analyzed are transformed into their quantile functions. Here, we review what quantiles of a data set are.

Functional Normalization

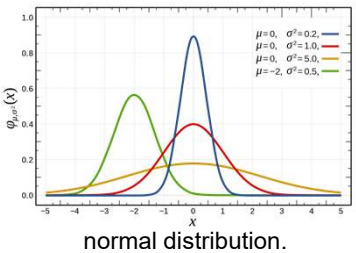
The **quantile function**, associated with a probability distribution *F* of a random variable, specifies the value of the random variable such that the probability of the variable being less than or equal to that value equals the given probability.

It is also called the percent-point function or inverse cumulative distribution function.

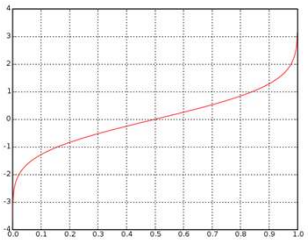
$$Q(p) = \inf\{x \in \Re : p \le F(x)\}$$

for a probability $0 < p < 1$.

www.wikipedia.org



normal distribution.



quantile function of the normal distribution.

Let us look at the quantile function of the standard normal distribution (blue curve in the upper plot).

Its quantile function is shown below.

For p=0.5, the variable has a 50% chance to be smaller than 0 in the normal distribution. Thus 0 is plotted on the y-axis for p = 0.5.

For p=0.1, the variable has a 10% chance to be smaller than (about) -1.3 in the normal distribution. Thus -1.3 is plotted on the y-axis for p = 0.1

For p=0.05 (the normal significance threshold), the value is -1.7. It is not -2 as we are used to (deviations of at least two standard deviations yield a p-value of 0.05) because we are only looking at one tail of the distribution.

Functional Normalization

We assume the following **model for the quantile function** over the interval $r \in [0,1]$

$$q_i^{\text{emp}}(r) = \alpha(r) + \sum_{j=1}^{m} Z_{i,j}\beta_j(r) + \epsilon_i(r)$$

α : mean of the quantile functions across all samples i,

β_j : coefficient functions associated with the covariates j and

ε_i : error functions, which are assumed to be independent and centered around 0.

In this model, the term $\sum_{j=1}^{m} Z_{i,j}\beta_j$

represents variation in the quantile functions explained by the covariates.

Functional normalization removes unwanted variation by regressing out this term.

FunNorm considers the quantile functions of the methylation value in all samples and takes its mean. This is termed alpha.

Then FunNorm assumes that the quantile function of a particlular sample i shows variation due to the covariates and some error term.

Functional Normalization

$\hat{\beta}_j$ for j = 1, …., m
are estimated using regression from the values observed for the **control probes**.

Assuming we have obtained estimates $\hat{\beta}_j$ for $j = 1, \ldots, m$, we form the functional normalized quantiles by

$$q_i^{\text{Funnorm}}(r) = q_i^{\text{emp}}(r) - \sum_{j=1}^{m} Z_{i,j}\hat{\beta}_j(r)$$

We then transform $\mathbf{Y}_i$ into the functional normalized quantity $\tilde{\mathbf{Y}}_i$ using the formula $\tilde{\mathbf{Y}}_i = q_i^{\text{Funnorm}}\left(\left(q_i^{\text{emp}}\right)^{-1}(\mathbf{Y}_i)\right)$

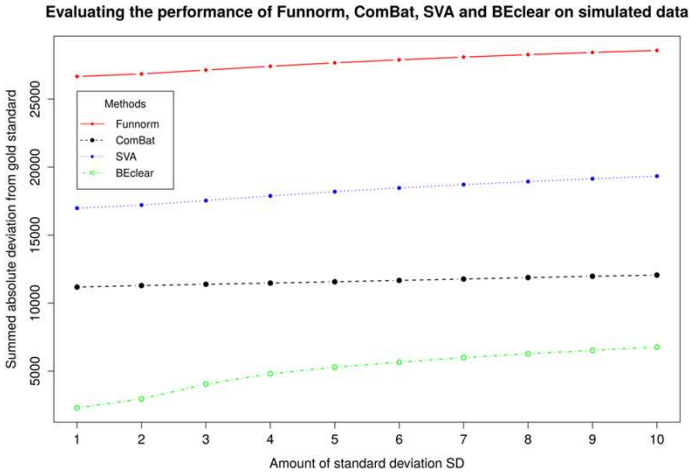
This ensures that the marginal distribution of $\tilde{\mathbf{Y}}_i$ has q_i^{Funnorm} as its quantile function.

The aim is to subtract the variation due to the covariates Z. The coefficients are estimated based on the values observed for the control probes.

The control probes are explained in the supplementary material of the FunNorm paper: “For “Bisulfite Conversion I” probes, 3 probes (C1,C2,C3) are expected to have high signal in the green channel in case the bisulfite conversion reaction was successful, and similarly 3 additional probes (C4,C5,C6) are expected to have high signal in the red channel. We therefore consider these 6 intensities and take the mean as a single summary value.“

So these probes do not detect methylation levels of CpG sites in the sample, but rather are a quality measure for the performed experiments.

Benchmarking BEclear



Funnorm, ComBat and SVA scale all values
-> large total deviation

BEclear corrects only affected entries

Here, we generated synthetic data sets with “known” batch effects.

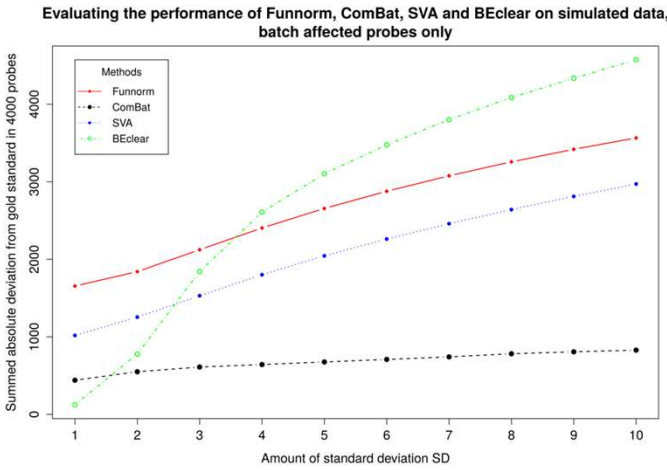
First, we determined the standard deviation of the methylation value of each promoter probe in level 1 adjacent normal samples (samples belonging to batch 136 were excluded due to the existing batch effect).

Then we randomly selected 8000 promoter probes (approximately 10% of all promoter probes present on the chip) and increased the methylation values of 4000 of these promoter probes by a specified multiple of their specific standard deviation plus a noise term. The original probe values before introducing the synthetic batch effect were considered as our gold standard.

Because the methods Funnorm, ComBat and SVA adjust all values, the summed deviation of the corrected values from the original values (y-axis) is quite large.

In contrast, BEclear modifies only the values that are affected by batch effects. Therefore, the summed deviations are much smaller.

Effect on corrected entries only



Even for affected entries,
BEclear predicts smallest changes for batch effects
up to 2 stand.dev.

which is a typical magnitude of batch effects.

Maybe the previous analysis was a bit unfair to the other methods.

Therefore, we now only inspect the deviation of the batch effected data points.

For small batch effects of 2 standard deviations or less (which is a typical magnitude), BEclear still produces the smallest deviations.

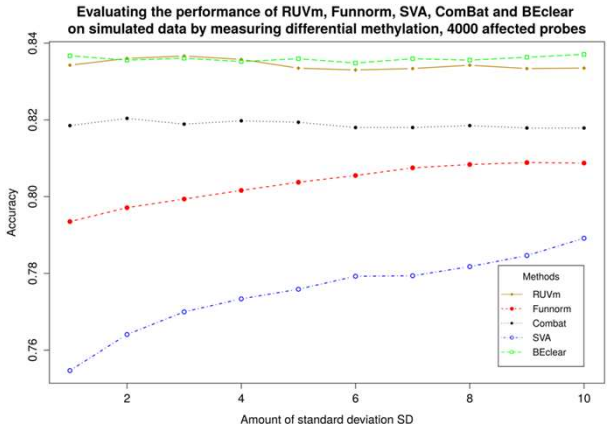
Only for larger deviations, BEclear-adjusted values differ more strongly from the original data that with the other methods.

Accuracy of differential methylation analysis

Identify differentially methylated CpG probes (tumor vs. normal) in original data

Then introduce synthetic batch effect (*n* x st.dev.) + noise term

Identify differentially methylated CpG probes again + compare to reference



Then, we considered the identities of differentially methylated genes in breast tumor samples vs. normal samples.

As gold standard reference, we used the list of differentially methylated probes identified in the unaffected data using the limma package.

Then, we designed a synthetic batch effect in a similar fashion as before and applied BEclear, RUVm, FunNorm, ComBat, and SVA to this data.

Then, again we identified differentially methylated genes in this BE-adjusted data with limma and compared the results to the original data.

Shown here is the accuracy defined as (TP + TN) / (TP + TN + FP + FN) for the different BE-adjustment methods.

BEclear yielded a similar accuracy as the RUVm method that is not explained.

Both methods were more accurate compared to all other methods.

Conclusions

Predicting **missing values or batch-effected values** by **Latent Factor Model** (**BEClear software**):

- Accuracy of MA hybridization prediction confirmed by WGS (97%), low LFM error
- Superior accuracy of predicting DNA methylation levels by LFM confirmed in benchmark against SVA, Combat, FunNorm tools.

Today, we started the lecture by discussing various approaches to reconstruct missing data points.

Then, we met the important problem of batch effects in the raw data. If one does not care about batch effects, the downstream analysis may be heavily corrupted.

Therefore, as a bioinformatician, it is your job to check for possible batch effects.

We discussed different approaches that are implemented in software tools for removing unwanted batch effects. In our view, there is no „best“ tool.

Certain approaches will offer advantages in certain situations and will give mediocre results in other cases. Identifying the best suited tool depends on the data to be analyzed.

Review: Foundations of Probability Theory

„**Probability**“ : degree of confidence that an event of an uncertain nature will occur.

„Events“ : we will assume that there is
an agreed upon **space Ω** of possible outcomes („events“).

E.g. a normal die (*dt. Würfel*) has a space $\Omega = \{1,2,3,4,5,6\}$

Also we assume that there is a set of **measurable events S**
to which we are willing to assign probabilities.

In the die example, the event $\{6\}$ is the case where the die shows 6.

The event $\{1,3,5\}$ represents the case of an **odd** outcome.

Here, I have compiled some basics from probability theory. Some of this will be considered as known to you in the following lectures.

Probably you know most of this already.

Quickly browsing over these slides will help you fresh up these things.

Foundations of Probability Theory

Probability theory requires that the **event space** satisfies 3 basic properties:

- It contains the **empty event** $\varnothing$ and the **trivial event** Ω.
- It is **closed under union** → if $\alpha, \beta \in S$, then so is $\alpha \cup \beta \in S$,
- It is **closed under complementation** → if $\alpha \in S$, then so is $\Omega - \alpha \in S$

The requirement that the event space is closed under union and complementation implies that it is also closed under other Boolean operations, such as intersection and set difference.

These are the 3 basic properties that every event space needs to fulfill.

Probability distributions

A **probability distribution** P *over* (Ω, S) is a mapping from events in S to real values. The mapping must satisfy the following conditions:

(1) $P(\alpha) \geq 0$ for all $\alpha \in$ S → *Probabilities are never negative*

(2) $P(\Omega) = 1$ → *The probability of the trivial event which allows all possible outcomes has the maximal possible probability of 1.*

(3) If $\alpha, \beta \in$ S and $\alpha \cap \beta = 0$ then $P(\alpha \cup \beta) = P(\alpha) + P(\beta)$

These are the 3 basic conditions that any probability distribution must obey.

Conditional probability

The **conditional probability** of β given α is defined as

$$P(\beta|\alpha) = \frac{P(\alpha \cap \beta)}{P(\alpha)}$$

In other words: The probability that β is true given that we know α is the relative proportion of outcomes satisfying β among those that satisfy α.

From this we see that

$$P(\alpha \cap \beta) = P(\alpha)P(\beta|\alpha)$$

This equality is know as the **chain rule** of conditional probabilities.

More generally, if α_1, α_2, … α_k are events, we can write

$$P(\alpha_1 \cap \alpha_2 \cap \cdots \cap \alpha_k) = P(\alpha_1)P(\alpha_2|\alpha_1) \ldots P(\alpha_k|\alpha_1 \cap \cdots \cap \alpha_{k-1})$$

Bayes rule

Another immediate consequence of the definition of conditional probability is **Bayes' rule**.

Due to symmetry, we can swap the 2 variables α and β in the definition $P(\beta|\alpha) = \frac{P(\alpha \cap \beta)}{P(\alpha)}$ and get the equivalent expression $P(\alpha|\beta) = \frac{P(\beta \cap \alpha)}{P(\beta)}$

If we rearrange, we get Bayes' rule $P(\beta|\alpha)P(\alpha) = P(\alpha|\beta)P(\beta)$ or

$$P(\alpha|\beta) = \frac{P(\beta|\alpha)P(\alpha)}{P(\beta)}$$

A more general conditional version of Bayes' rule where all probabilities are conditioned on some background event γ also holds:

$$P(\alpha|\beta \cap \gamma) = \frac{P(\beta|\alpha \cap \gamma)P(\alpha|\gamma)}{P(\beta|\gamma)}$$

Example 1 for Bayes rule

Consider a student population.
Let `Smart` denote smart students and `GradeA` denote students who got grade A.

Assume we believe that P(`GradeA|Smart`) = 0.6, and that we get to know that a particular student received grade A.

Suppose that P (`Smart`) = 0.3 and P (`GradeA`) = 0.2

Then we have P(`Smart|GradeA`) = 0.6 × 0.3 / 0.2 = 0.9

In this model, an A grade strongly suggests that the student is smart.

On the other hand, if the test was easier and high grades were more common, e.g. P (`GradeA`) = 0.4, then we would get

P(`Smart|GradeA`) = 0.6 × 0.3 / 0.4 = 0.45 which is much less conclusive.

Example 2 for Bayes rule

Suppose that a tuberculosis skin test is 95% percent accurate.

That is, if the patient is TB-infected, then the test will be positive with probability 0.95 and if the patient is not infected, the test will be negative with probability 0.95.

Now suppose that a person gets a positive test result.

What is the probability that the person is infected?

Naive reasoning suggests that if the test result is wrong 5% of the time, then the probability that the subject is infected is 0.95.

That would mean that 95% of subjects with positive results have TB.

Example 2 for Bayes rule

If we consider the problem by applying Bayes' rule, we need to consider the prior probability of TB infection, and the probability of getting a positive test result.

Suppose that 1 in 1000 of the subjects who get tested is infected → P(TB) = 0.001

We see that 0.001×0.95 infected subjects get a positive result
and 0.999×0.05 uninfected subjects get a positive result.

Thus $P(\text{Positive}) = 0.001 \times 0.95 + 0.999 \times 0.05 = 0.0509$

Applying Bayes' rule, we get $P(\text{TB}|\text{Positive}) = P(\text{TB}) \times P(\text{Positive}|\text{TB}) / P(\text{Positive})$
$= 0.001 \times 0.95 / 0.0509 \cong 0.0187$

Thus, although a subject with a positive test is much more probable to be TB-infected than a random subject, fewer than 2% of these subjects are TB-infected.

Random Variables

A **random variable** is defined by a function
that associates a value with each outcome in Ω.

For students in a class, this could be a function f_{grade} that maps
each student in the class (in Ω) to his or her grade (1, …, 5).

The event `grade = A` is a shorthand for the event $\{\omega \in \Omega: f_{grade}(\omega) = A\}$.

There exist **categorical (or discrete) random values** that take on
one of a few values, e.g. intelligence could be „high“ or „low“.

There also exist **integer or real random variable** that can take on
an infinite number of continuous values, e.g. the height of students.

By Val(X) we denote the set of values that a random variable X can take.

Random Variables

In the following, we will either consider categorical random variables or random variables that take real values.

We will use capital letters X, Y, Z to denote random variables.

Lowercase values will refer to the values of random variables.

E.g. $P(X = x) \geq 0 \text{ for all } x \in Val(X)$

When we discuss categorical random numbers, we will denote the *i-th* value as x^i.

Bold capital letters are used for sets of random variables: **X, Y, Z**.

Marginal Distributions

Once we define a random variable X, we can consider the **marginal distribution** *P(X)* over events that can be described using X.

E.g. let us take the two random variables `Intelligence` and `Grade` and their marginal distributions `P(Intelligence)` and `P(Grade)`

Let us suppose that

$$P(\text{Intelligence=high}) = 0.3$$
$$P(\text{Intelligence=low}) = 0.7$$

$$P(\text{Grade=A}) = 0.25$$
$$P(\text{Grade=B}) = 0.37$$
$$P(\text{Grade=C}) = 0.38$$

These marginal distributions are probability distributions satisfying the 3 properties.

Joint Distributions

Often we are interested in questions that
involve the values of several random variables.

E.g. we might be interested in the event „`Intelligence = high` and `Grade = A`".

In that case we need to consider the **joint distribution** $P(X_1, \dots, X_n)$
over these two random variables.

The joint distribution of 2 random variables has to be consistent
with the marginal distribution in that $P(x) = \sum_y P(x, y)$.

		Intelligence		
		low	high	Σ
Grade	A	0.07	0.18	0.25
	B	0.28	0.09	0.37
	C	0.35	0.03	0.38
	Σ	0.7	0.3	1

Conditional Probability

The notion of conditional probability extends to induced distributions over random variables.

$P(\text{Intelligence}|\text{Grade}=\text{A})$ denotes the conditional distribution over the events describable by `Intelligence` given the knowledge that the student's grade is A.

Note that the conditional probability $P(\text{Intelligence}=\text{high}|\text{Grade}=\text{A}) = \frac{0.18}{0.25} = 0.72$ is quite different from the marginal distribution $P(\text{Intelligence}=\text{high}) = 0.3$.

We will use the notation $P(X|Y)$ to present a set of conditional probability distributions.

Bayes' rule in terms of conditional probability distributions reads

$$P(X|Y) = \frac{P(X)P(Y|X)}{P(Y)}$$

Probability Density Functions

A function $p: \mathbb{R} \to \mathbb{R}$
is a **probability density function** (PDF) for X

if it is a nonnegative integrable function so that $\int_{Va\ (X)} p(x)dx = 1$

The function $\mathrm{P}(X \leq a) = \int_{-\infty}^{a} p(x)dx$ is the **cumulative distribution** for X.

By using the density function we can evaluate the probability of other events. E.g.

$$\mathrm{P}(a \leq X \leq b) = \int_{a}^{b} p(x)dx$$

Uniform distribution

The simplest PDF is the **uniform distribution**

Definition: A variable X has a uniform distribution over [a,b] denoted X ~ Unif[a,b] if it has the PDF

$$p(x) = \begin{cases} \frac{1}{b-a} & b \geq x \geq a \\ 0 & \text{otherwise} \end{cases}$$

Thus the probability of any subinterval of [*a,b*] is proportional to its size relative to the size of [*a,b*].

If $b - a < 1$, the density can be greater than 1.

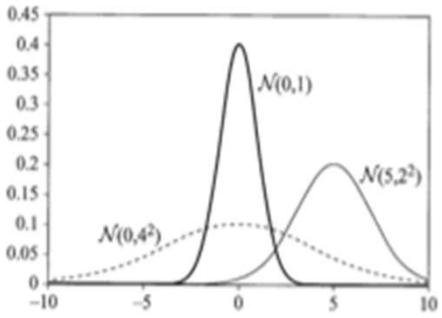
We only have to satisfy the constraint that the total area under the PDF is 1.

Gaussian distribution

A random variable X has a Gaussian distribution with mean μ and variance σ^2 , denoted $X \sim N(\mu;\sigma^2)$ if it has the PDF

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

A standard Gaussian has mean 0 and variance 1.



Expectation

Let X be a discrete random variable that takes numerical values.

Then, the **expectation** of X under the distribution P is

$$\mathbf{E}_P[X] = \sum_x x \cdot P(x)$$

If X is a continuous variable,
then we use the density function

$$\mathbf{E}_P[X] = \int x \cdot p(x)dx$$

E.g. if we consider X to be the outcome of rolling a good die with probability 1/6 for each outcome, then **E**[X] = 1 · 1/6 + 2 · 1/6 + … + 6 · 1/6 = 3.5

Properties of the expectation of a random variable

$\mathbf{E}[a \cdot X + b] = a\,\mathbf{E}[X] + b$

Let X and Y be two random variables
$\mathbf{E}[X + Y] = \mathbf{E}[X] + \mathbf{E}[Y]$
Here, it does not matter whether X and Y are independent or not.

What can be say about the expectation value of a product of two random variables?

In the general case, we can say very little.

Consider 2 variables X and Y that each take on the values +1 and -1 with probabilities 0.5.
If X and Y are independent, then $\mathbf{E}[X \cdot Y] = 0$.
If they always take on the same value (they are correlated), then $\mathbf{E}[X \cdot Y] = 1$.

Properties of the expectation of a random variable

If X and Y are independent then
E[X · Y] = **E**[X] · **E**[Y]

The **conditional expectation** of X given y is

$$E_P[X|y] = \sum_x x \cdot P(x|y)$$

Variance

The expectation of X tells us the mean value of X. However, it does not indicate how much X deviates from this value. A measure of this deviation is the **variance** of X:

$$Var_P[X] = \mathbf{E}_P[(X - \mathbf{E}_P[X])^2]$$

The variance is the **expectation** of the **squared difference** between X and its expected value. An alternative formulation of the variance is

$$Var[X] = \mathbf{E}[X^2] - (\mathbf{E}[X])^2$$

If X and Y are independent, then $Var[X + Y] = Var[X] + Var[Y]$

$$Var[a \cdot X + b] = a^2 Var[X]$$

For this reason, we are often interested in the square root of the variance, which is called the **standard deviation** of the random variable. We define

$$\sigma_X = \sqrt{Var[X]}$$

Variance

Let X be a random variable with Gaussian distribution $N(\mu;\sigma^2)$.

Then **E**[X] = μ and Var[X] = σ^2.
Thus, the parameters of the Gaussian distribution specify the expectation and the variance of the distribution.

The form of the Gaussian distribution implies that the density of values of X drops exponentially fast in the distance (x - μ) / σ.

Not all distributions show such a rapid decline in the probability of outcomes that are distant from the expectation.
However, even for arbitrary distributions, one can show that there is a decline.

The **Chebyshev inequality** states $P(|X - \mathbf{E}_P[X]| \geq t) \leq \frac{Var_P[X]}{t^2}$

or in terms of σ $P(|X - \mathbf{E}_P[X]| \geq k\sigma_X) \leq \frac{1}{k^2}$

Resources

Nice **online resources** on statistics:

https://www.khanacademy.org/math/statistics-probability

http://tutorials.istudy.psu.edu/basicstatistics/

https://stattrek.com/statistics/problems.aspx