Knowledge Based Systems for Bioinformatics
Lecture 1 2010

Professor Jan Komorowski

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

KSB in bioinformatics - Syllabus

- Lecture 1 Machine Learning
  - Unsupervised and Supervised
  - Clustering: hierarchical, k-means, k-nearest neighbors
- Lab: Clustering

- Lecture 2 and 3 rough sets
  - IS, information vectors, equivalence classes, reducts, discernibility matrices, decision systems, relative reducts, rule-generation
  - discretization, approximate reducts, training, cross-validation, accuracy, coverage, support, ROC/AUC, randomization
- Lab: Paper-based exercises: reduct computation etc

- Lecture 4 (application lecture)
  - Ontologies
  - classification with microarrays (cancer + time profiles)
  - HIV-1 applications (Kontijevskis' protease)
- Lab: Rosetta

KSB in bioinformatics - Syllabus

- Lecture 5 Decision trees,
  - Decision tress
  - MC feature selection
- Lab:
  - construction of decision trees (on paper)
  - MC (cancer data)

- Lecture 6
  - HIV-1 revisited (Kierczak's RT)
  - Histone modifications
- Lab:
  - MC applications: HIV-1 and histones
  - Histone modifications and HIV-1: MCFS ordering and rule generation, interpretation
- Lecture 7 Constraint programming
  - Biologically motivated exercises

KSB in bioinformatics - Syllabus

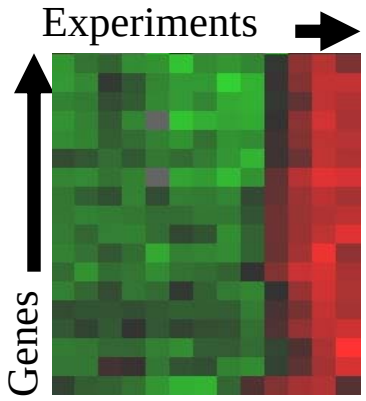
- Lecture 8 Unstructured databases (PubGene)
- Lecture 9 Gene networks

Course Contents – Methods and Theory

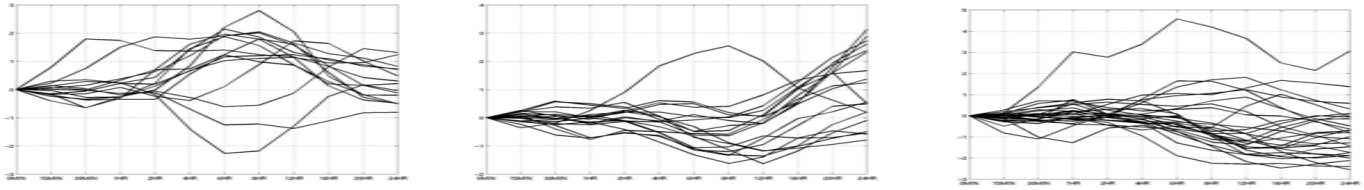
- Mathematical Prerequisites
  - Discrete structures
  - Statistics

Course Contents – Applications I

- Classification of samples (different cancer types)

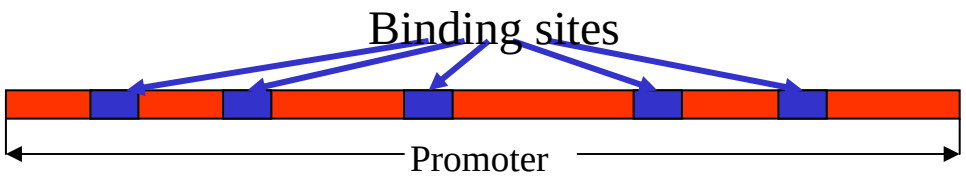


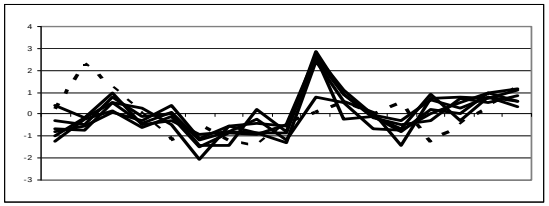
- Prediction of gene function from expression data



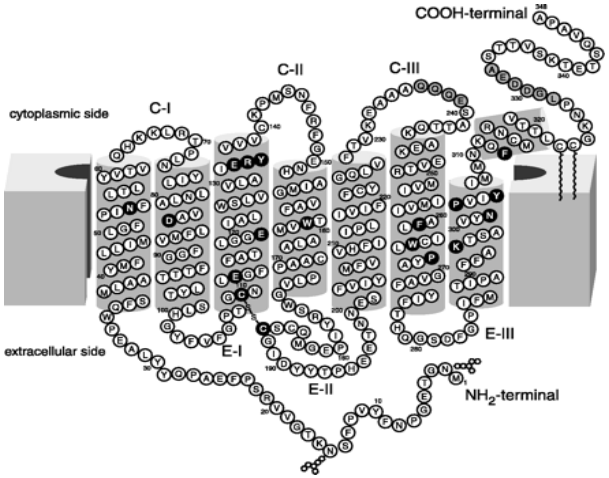
Course Contents – Applications II

- Identification of modules of transcription factor binding sites





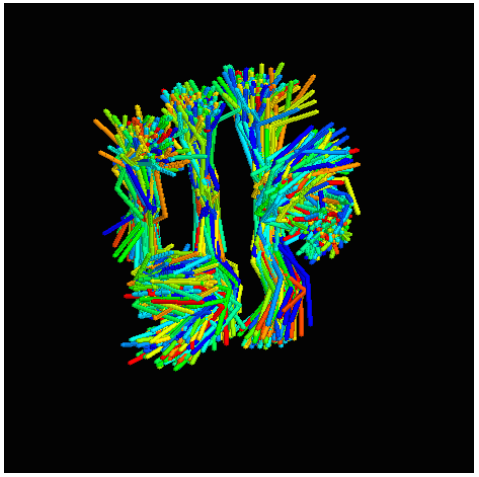
- Modeling ligand-receptor binding aiffinities.



- Medical applications: HIV, Predicting orgin of metastatic cancers

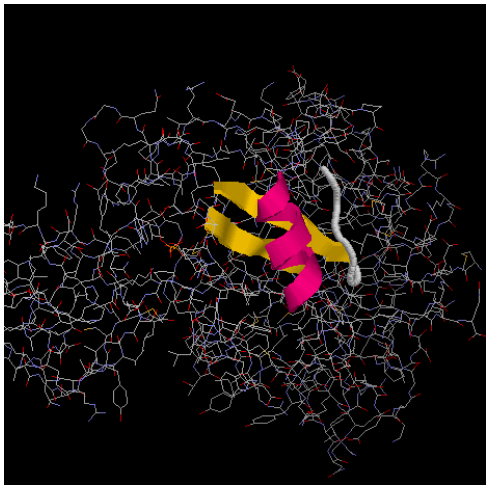
Course Contents – Applications III

- Structural alignment of local descriptors



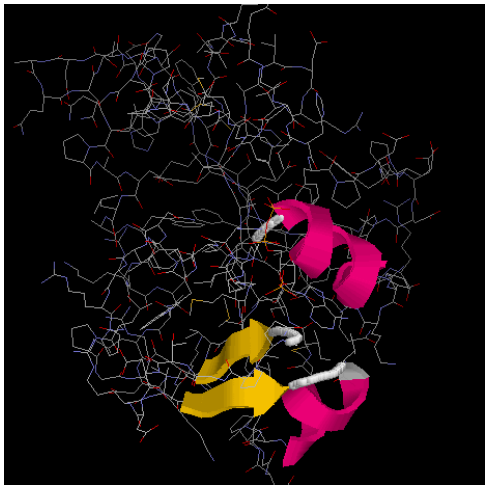
Predicting protein function from structure – A sample rule

IF



1gsa_2#218
(annotated 56 times to 24 different GO classes)

AND



1ra9__#62
(annotated 22 times to 12 different GO classes)

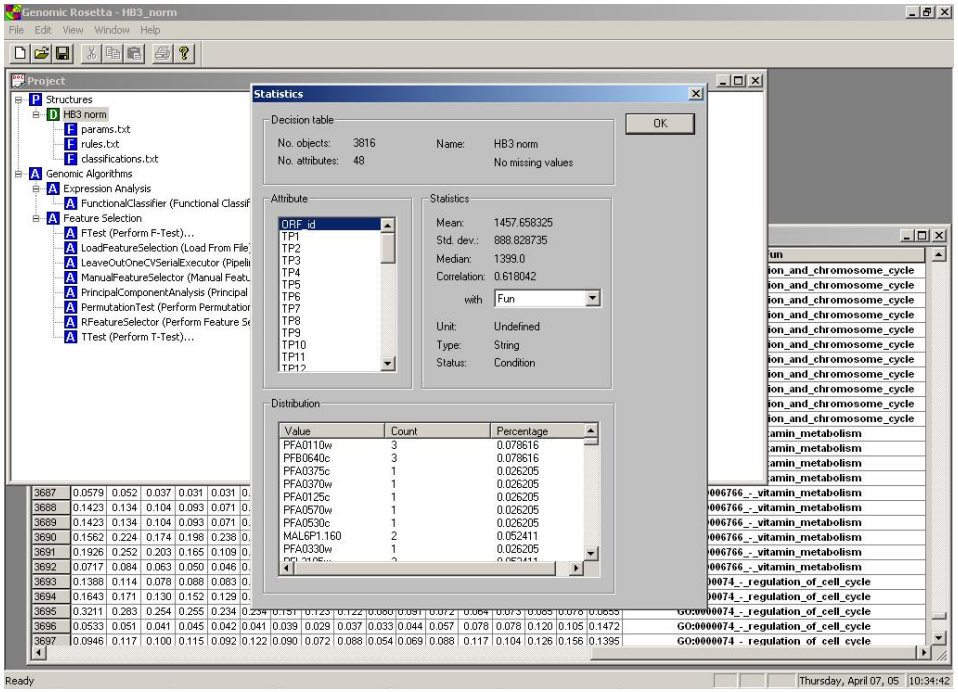
THEN GO:0016646 - *oxidoreductase activity, acting on the CH-NH group of donors, NAD or NADP as acceptor*

•(7 of 11 proteins annotated with GO:0016646)

Coverage = 7/11, Accuracy = 1

Course Contents - Practicals

- After most lectures, there will be practical computer exercises connected to the lecture.
- Hands-on experience on methods and problems.



Examination

- Practicals, 1p
- Project 1p
- Written Exam 3p

Supervised learning

- Supervised learning:
  - the task of inferring a function from supervised training data.
- The training data
  - a set of training examples
  - each example is a pair consisting of an input object (a vector) and a desired output value.
- A supervised learning algorithm analyzes the training data and produces an inferred function, which is called a classifier
  - Classification, if the output is discrete,
  - regression function, if the output is continuous
- The inferred function should predict the correct output value for any valid input object.
  - the learning algorithm generalizes from the training data to unseen situations in a "reasonable" way.

Unsupervised learning

- Unsupervised learning
  - a class of problems in which one seeks to determine how the data is organized.
  - the learning algorithm is given only unlabeled examples.

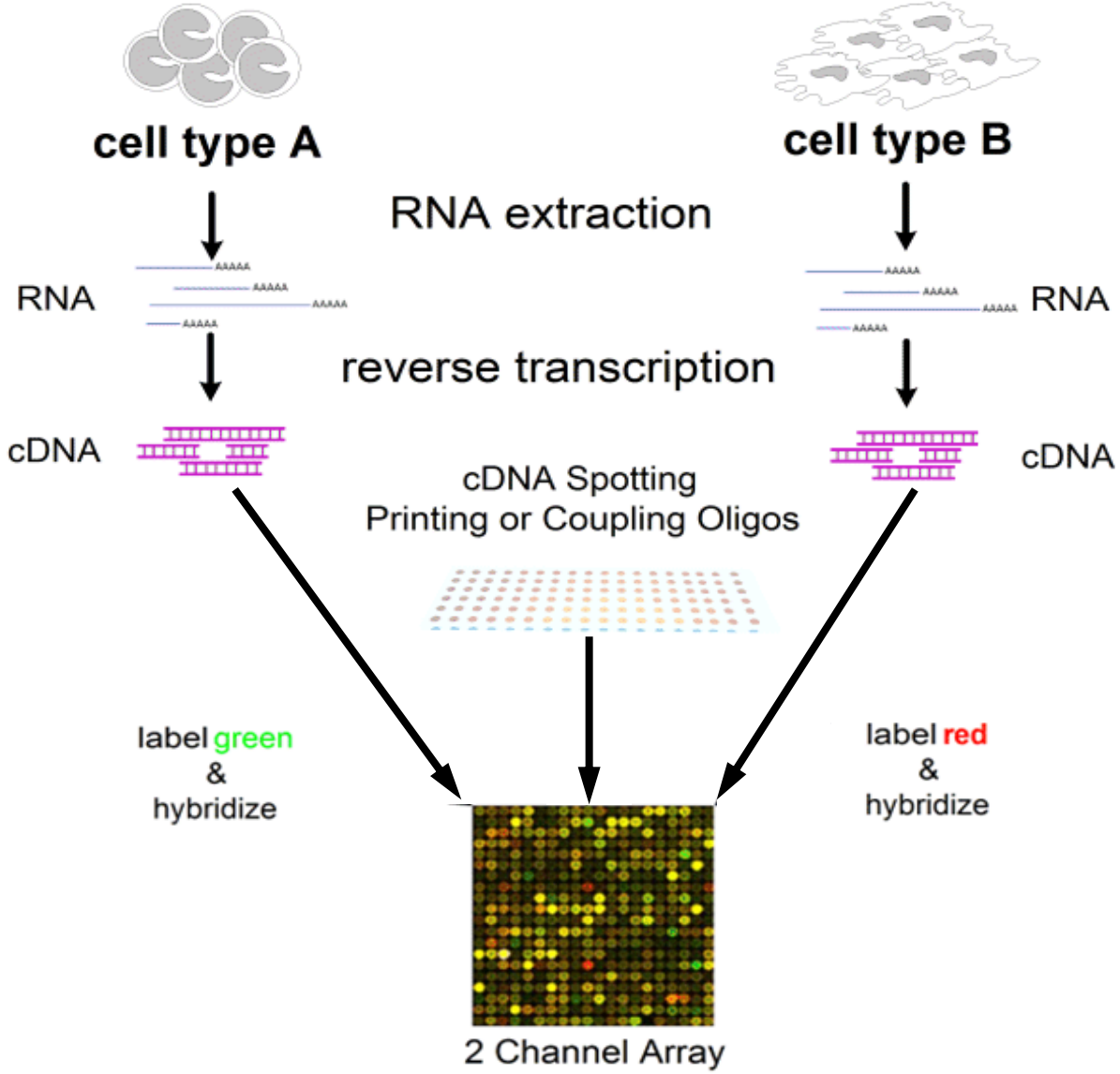
Introduction

- Topic: Application of machine learning to microarray data, for classification of samples.
- Background on microarray data
- Unsupervised learning (clustering, class discovery); used to “discover” natural groups of genes/experiments e.g.
  - discover subclasses of a form of cancer that is clinically homogenous
- Supervised learning; used to “learn” a model of a set of predefined classes of genes/experiments e.g.
  - diagnosis of cancer/subclasses of cancer
  - Example: Rough Sets

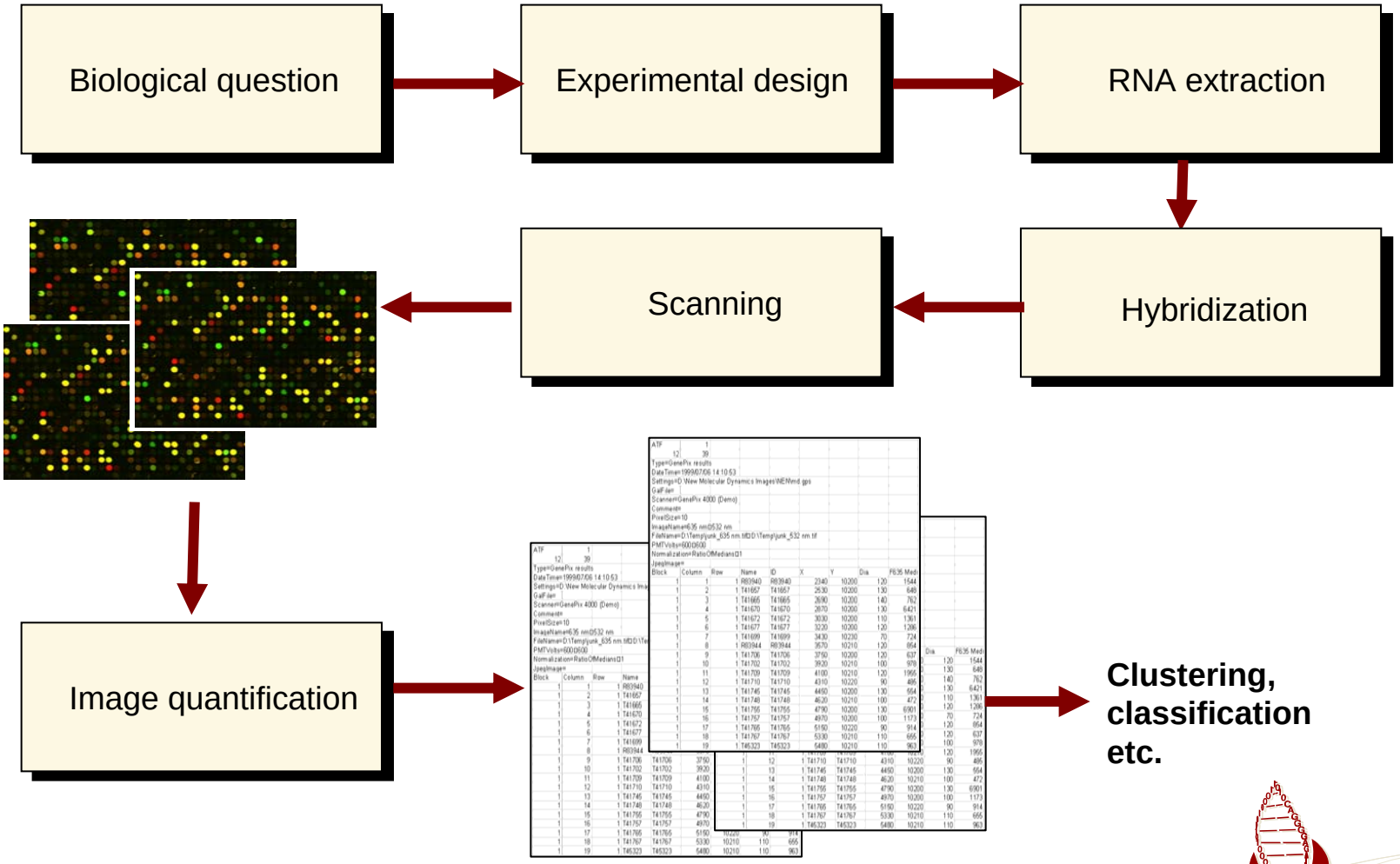
Microarray experiments

- Microarray technology is a method for measuring levels of expression of thousands of genes simultaneously.
  - Conceptually simple
  - Cost effective
- Large amounts of data → computer analysis
  - Removal of systematic errors, pre-processing
  - Hypothesis testing – significant genes
  - Unsupervised learning
  - Supervised learning

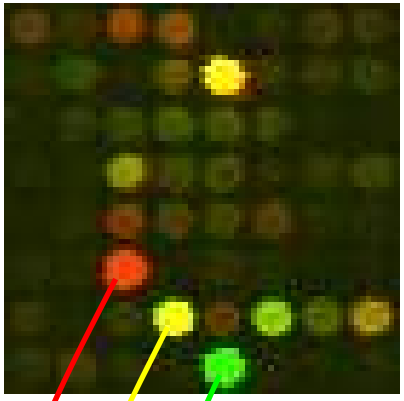
Microarray experiments II



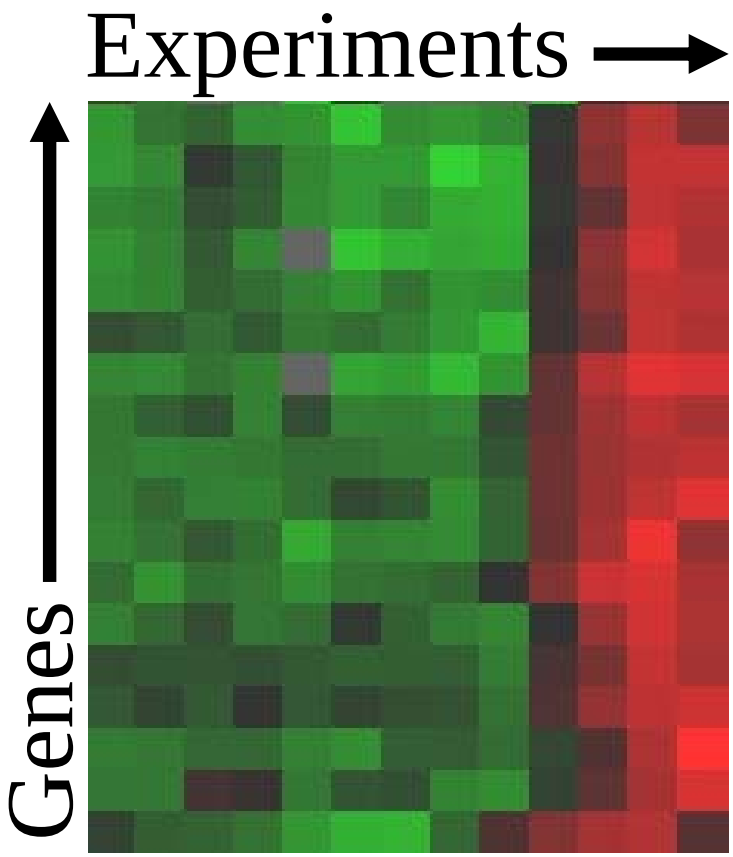
Microarray experiments III



Microarray data



Cy3	Cy5	$\frac{Cy5}{Cy3}$	$\log_2(\frac{Cy5}{Cy3})$
200	10000	50.00	5.64
4800	4800	1.00	0.00
9000	300	0.03	-4.91



Microarray data II

M < 100

Gene/Expr	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	…	EM
G1	0,72	0,10	0,57	1,08	0,66	0,39	0,49	0,28	0,50	0,66	…	0,52
G2	1,58	1,05	1,15	1,22	0,54	0,73	0,82	0,82	0,90	0,73	…	0,75
G3	1,10	0,97	1,00	0,90	0,67	0,81	0,88	0,77	0,71	0,57	…	0,46
G4	0,97	1,00	0,85	0,84	0,72	0,66	0,68	0,47	0,61	0,59	…	0,65
G5	1,21	1,29	1,08	0,89	0,88	0,66	0,85	0,67	0,58	0,82	…	0,60
G6	1,45	1,44	1,12	1,10	1,15	0,79	0,77	0,78	0,71	0,67	…	0,36
G7	1,15	1,10	1,00	1,08	0,79	0,98	1,03	0,59	0,57	0,46	…	0,39
G8	1,32	1,35	1,13	1,00	0,91	1,22	1,05	0,58	0,57	0,53	…	0,43
G9	1,01	1,38	1,21	0,79	0,85	0,78	0,73	0,64	0,58	0,43	…	0,47
…	…	…	…	…	…	…	…	…	…	…	…	…
GN	0,85	1,03	1,00	0,81	0,82	0,73	0,51	0,24	0,54	0,43	…	0,51

N ≈ 10000

2.3/2.6 = “Red/Green”

Microarray data III

- After log-transformation:

M < 100

Gene/Expr	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	...	EM
G1	-0,47	-3,32	-0,81	0,11	-0,60	-1,36	-1,03	-1,84	-1,00	-0,60	…	-0,94
G2	0,66	0,07	0,20	0,29	-0,89	-0,45	-0,29	-0,29	-0,15	-0,45	…	-0,42
G3	0,14	-0,04	0,00	-0,15	-0,58	-0,30	-0,18	-0,38	-0,49	-0,81	…	-1,12
G4	-0,04	0,00	-0,23	-0,25	-0,47	-0,60	-0,56	-1,09	-0,71	-0,76	…	-0,62
G5	0,28	0,37	0,11	-0,17	-0,18	-0,60	-0,23	-0,58	-0,79	-0,29	…	-0,74
G6	0,54	0,53	0,16	0,14	0,20	-0,34	-0,38	-0,36	-0,49	-0,58	…	-1,47
G7	0,20	0,14	0,00	0,11	-0,34	-0,03	0,04	-0,76	-0,81	-1,12	…	-1,36
G8	0,40	0,43	0,18	0,00	-0,14	0,29	0,07	-0,79	-0,81	-0,92	…	-1,22
G9	0,01	0,46	0,28	-0,34	-0,23	-0,36	-0,45	-0,64	-0,79	-1,22	…	-1,09
...	…	…	…	…	…	…	…	…	…	…	…	…
GN	-0,23	0,04	0,00	-0,30	-0,29	-0,45	-0,97	-2,06	-0,89	-1,22	…	-0,97

N ≈ 10000

log(2.3/2.6) = log(“Red/Green”)

Data analysis – what can we study?

What to study?

- Classes of experiments; changes in expression levels in tissue samples with different e.g. diseases, treatments, environmental effects etc.
- Classes of genes; expression profiles of genes with similar biological function
- Both of the above

Clustering - introduction

- Need to define;
  - measure of similarity (i.e. Euclidian, Pearson correlation)
  - algorithm for using the measure of similarity to discover natural groups in the data

The number of ways to divide n items into k clusters: $k^n/k!$

Example: $10^{500}/10! = 2.756 \times 10^{493}$

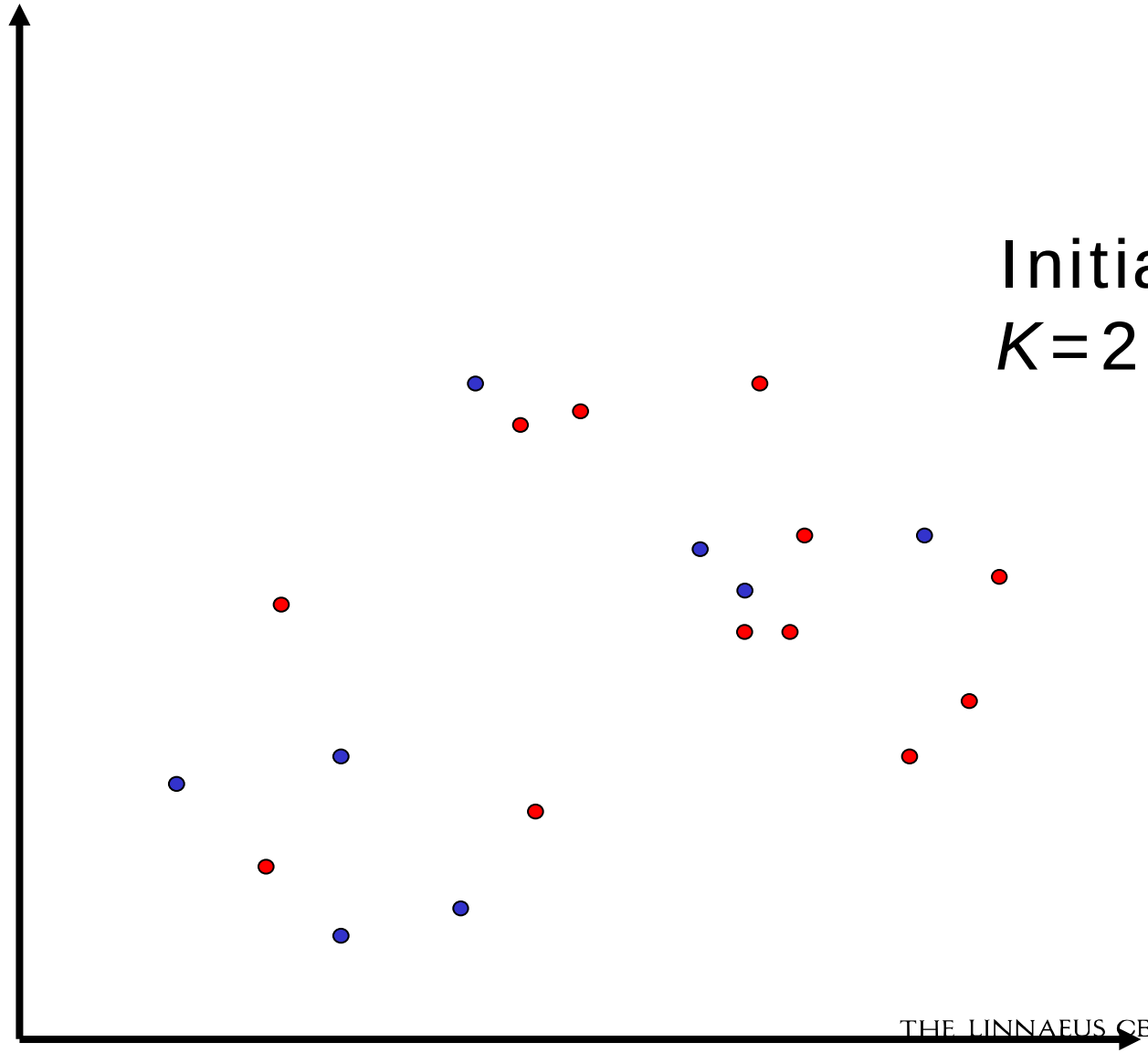
K-means clustering

- Split the data into *k* random clusters
- Repeat
  - calculate the centroid of each cluster
  - (re-)assign each gene/experiment to the closest centroid
  - stop if no new assignments are made

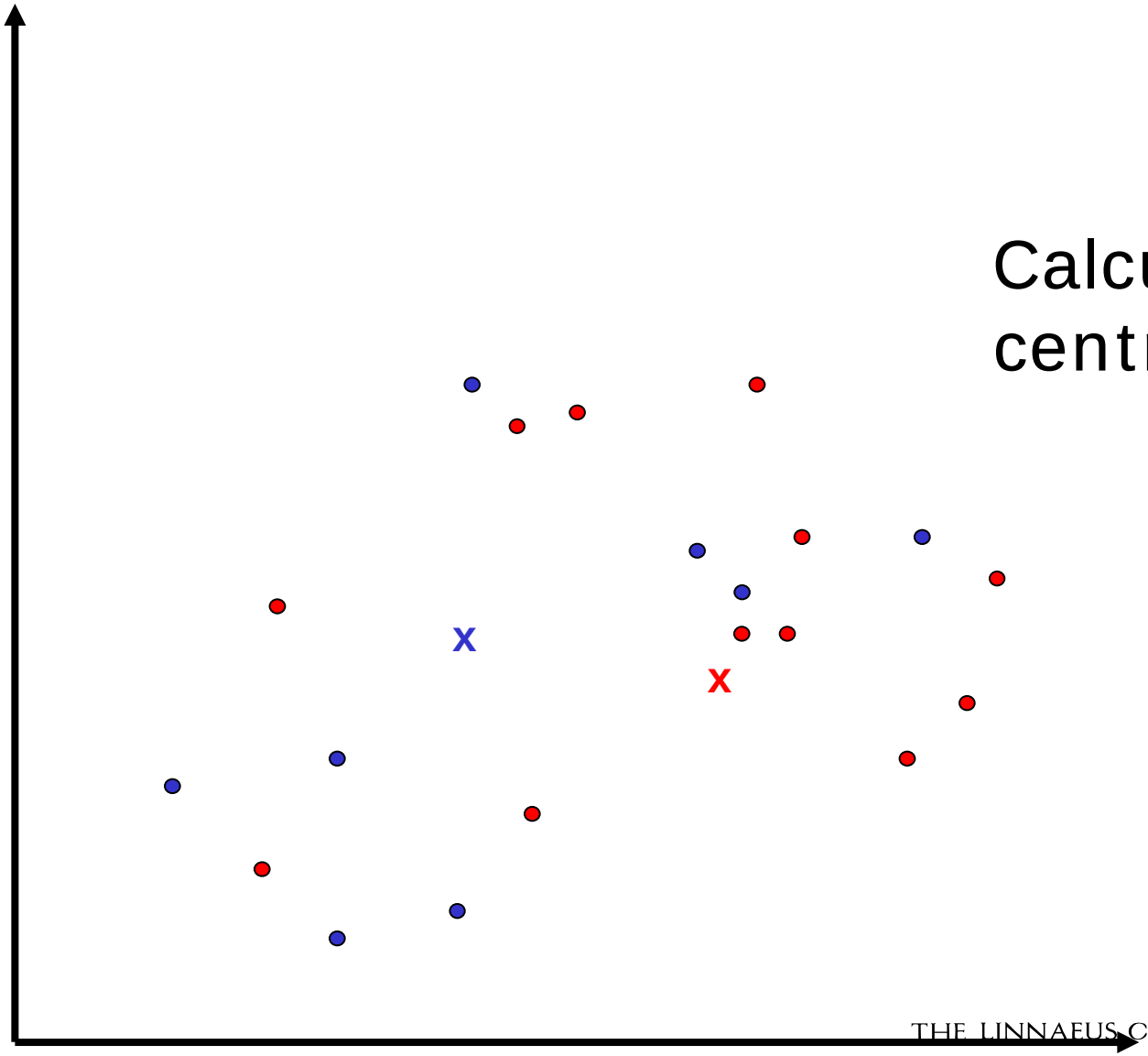
K-means clustering - features

- Low memory usage
- Running time: $O(n)$
- Improves iteratively: not trapped in previous mistakes
- Non-deterministic: will in general produce different clusters with different initializations
- Number of clusters must be decided in advance

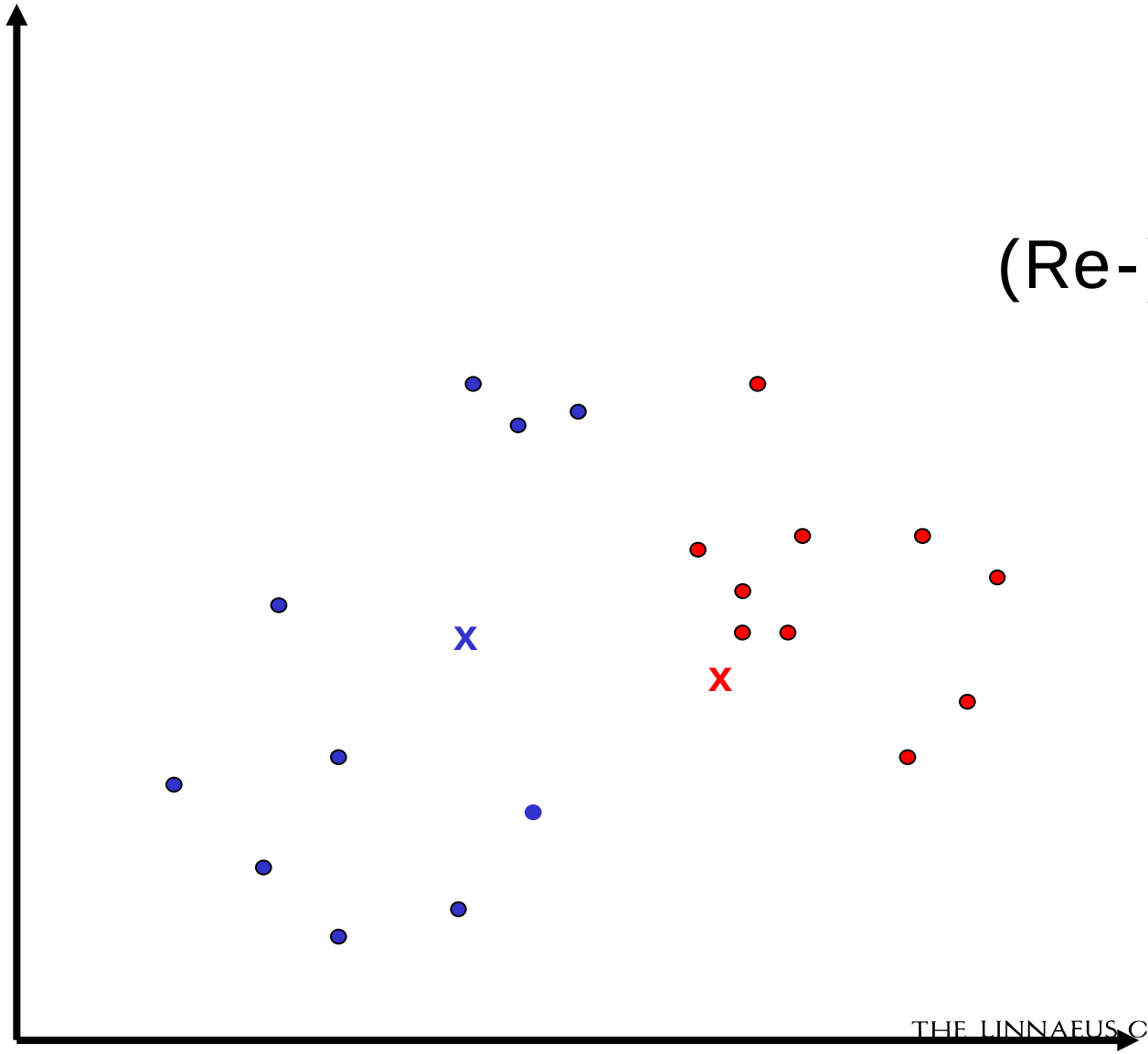
Example of K-means: two dimensions



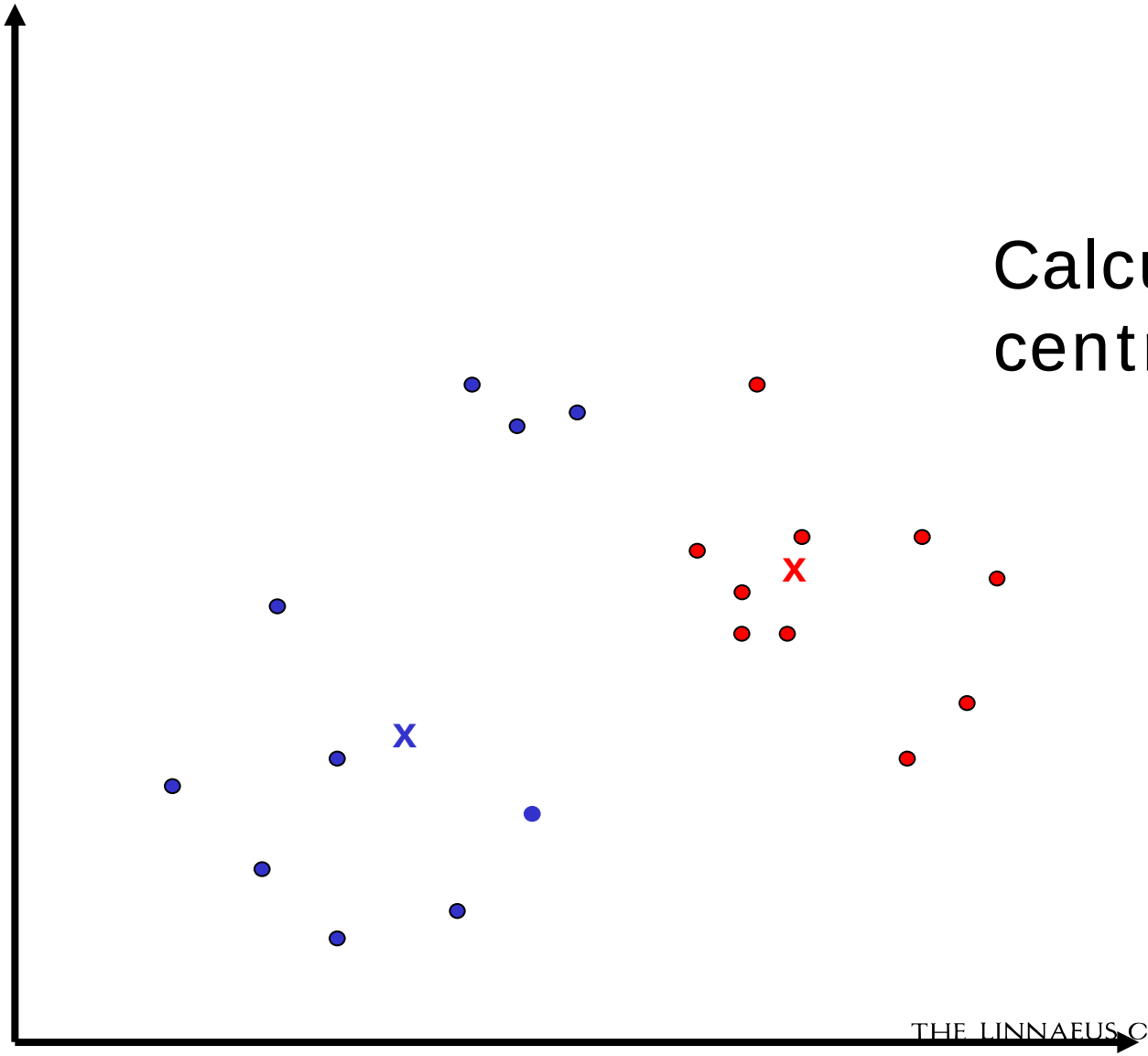
Iteration 1



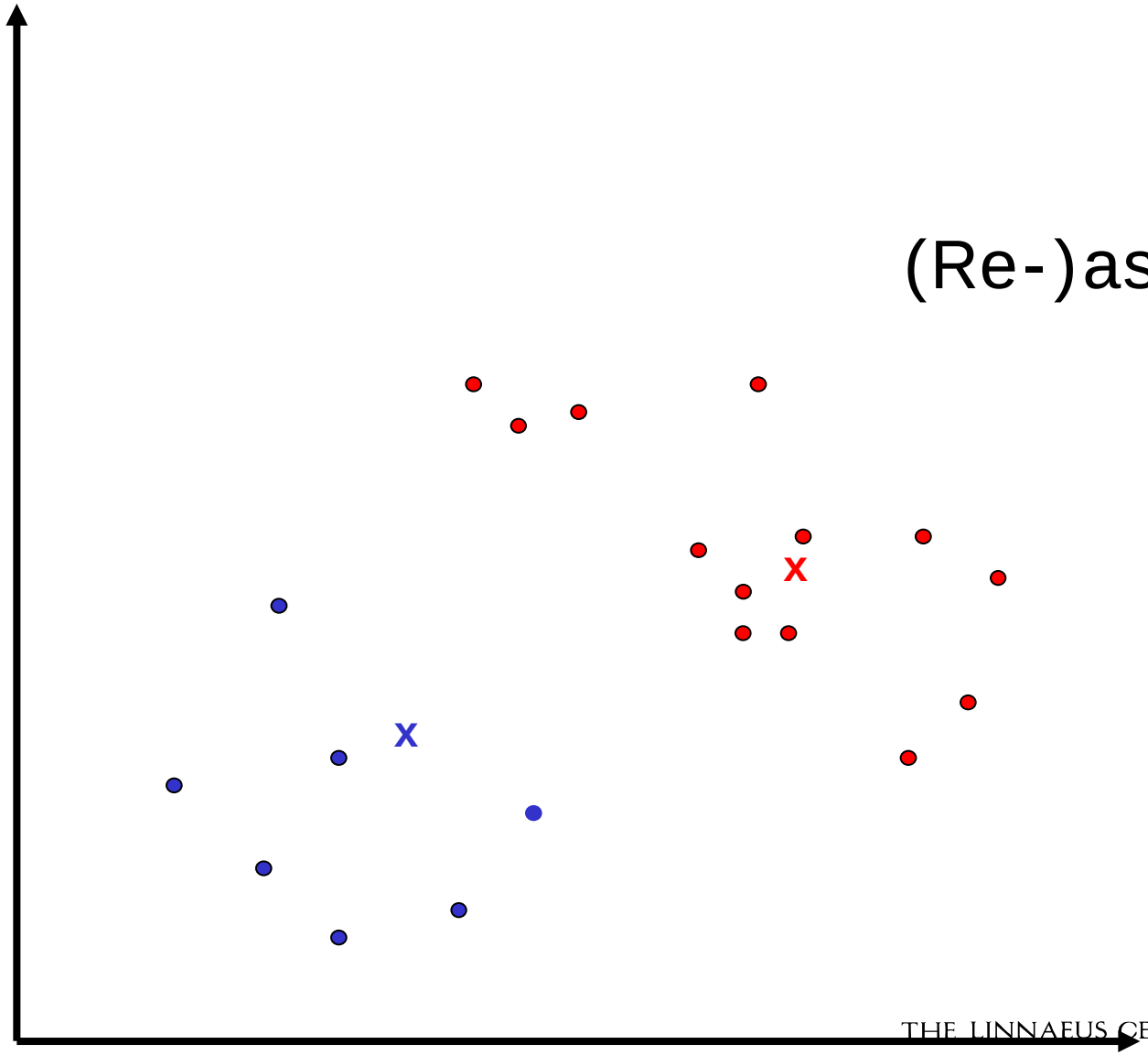
Iteration 1



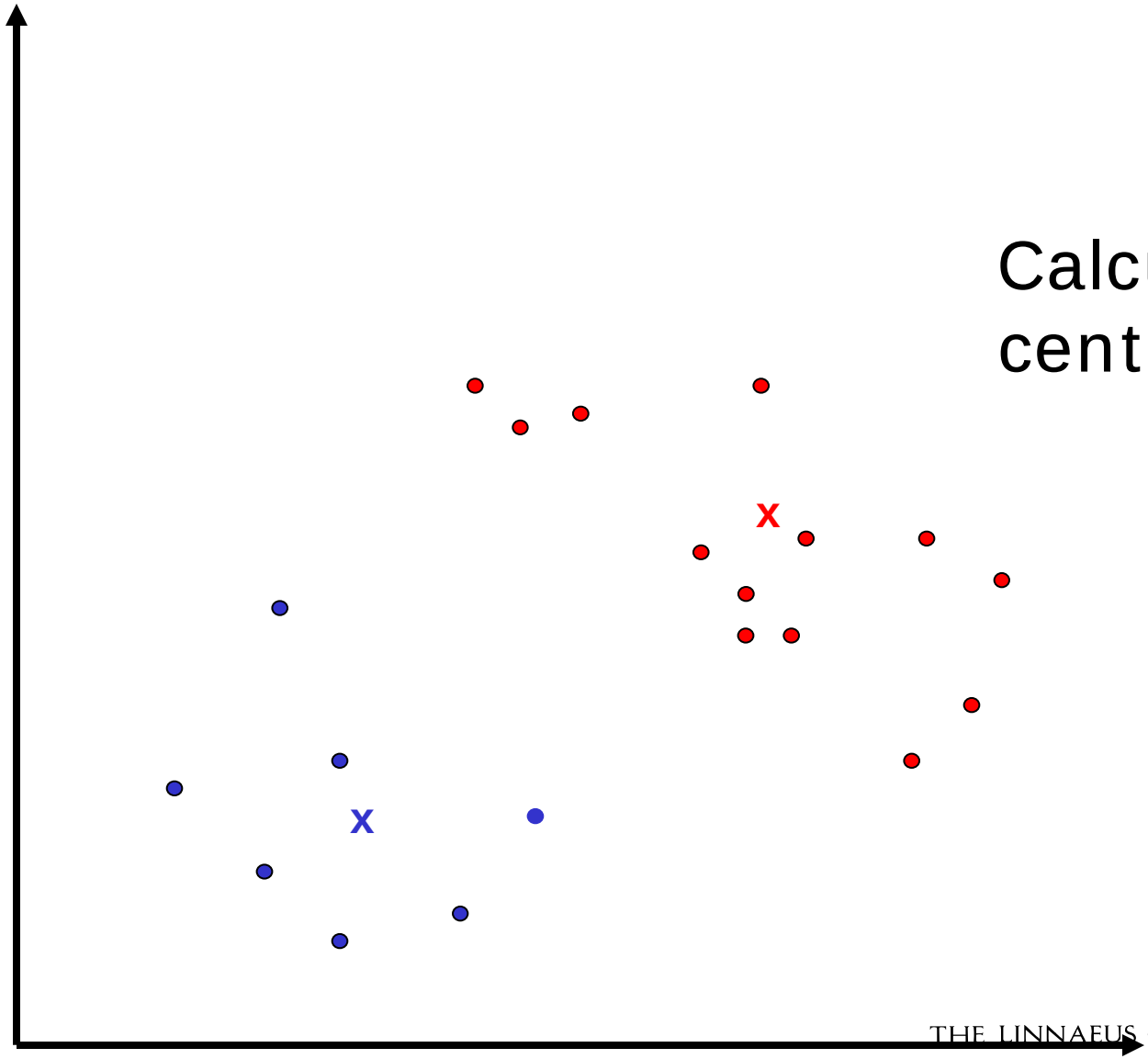
Iteration 2



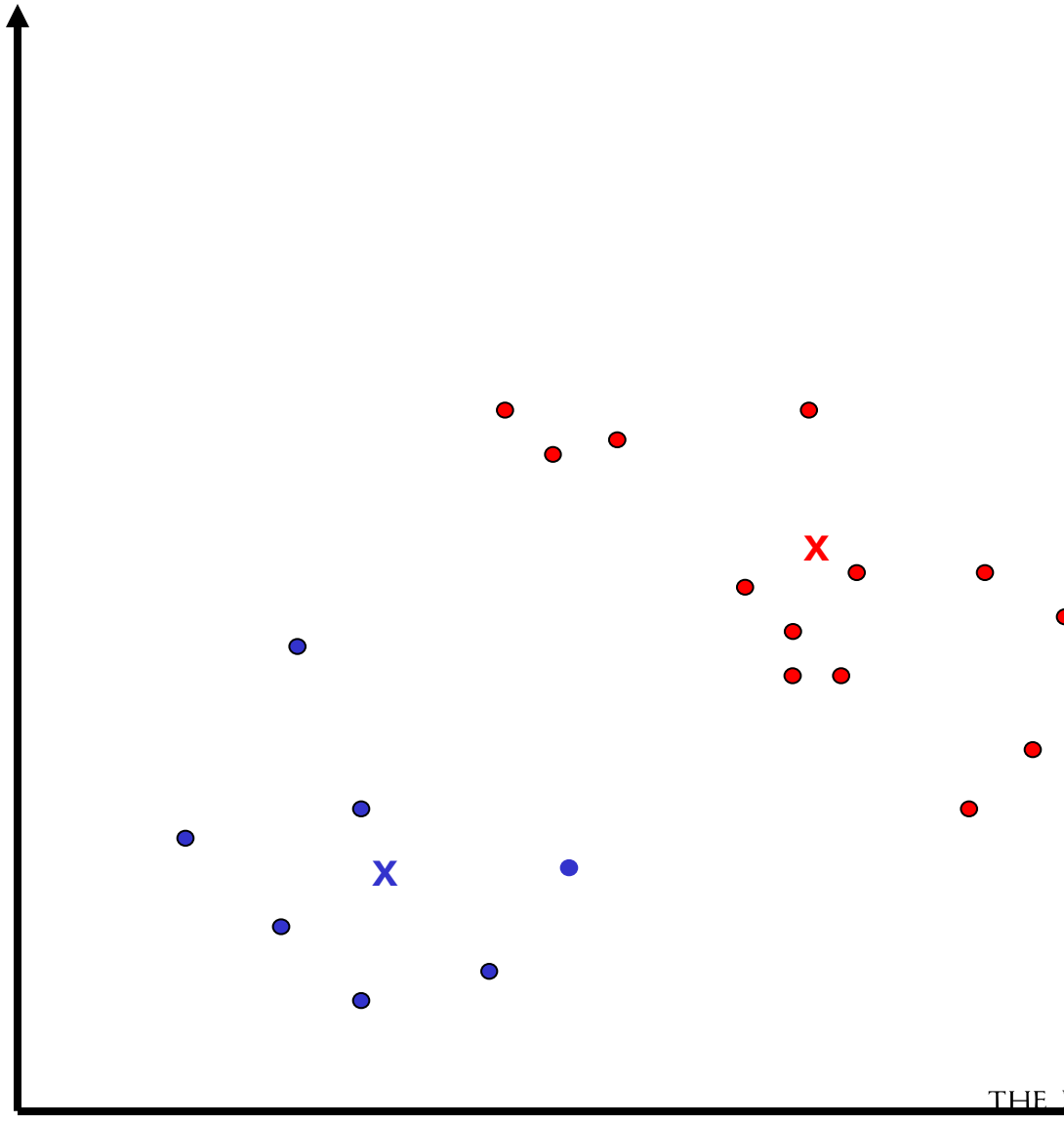
Iteration 2



Iteration 3



Iteration 3



(Re-)assign

No new assignments!
STOP

Hierarchical Clustering

INPUT: *n* genes/experiments

- Consider each gene/experiment as an individual cluster and initiate an $n \times n$ distance matrix ***d***
- Repeat
  - identify the two most similar clusters in ***d*** (i.e. smallest number in ***d***)
  - merge the two most similar clusters and update the matrix (i.e. substitute the two clusters with the new cluster)

OUTPUT: A tree of merged genes/experiments (called a dendrogram)

Hierarchical Clustering - Features

- Huge memory requirements: stores the $n \times n$ matrix
- Running time: $O(n^3)$
- Deterministic: produces the same clustering each time
- Nice visualization: dendrogram
- Number of clusters can be selected using the dendrogram

Example of hierarchical clustering: languages of Europe

TABLE 12.3 NUMERALS IN 11 LANGUAGES

English (E)	Norwegian (N)	Danish (Da)	Dutch (Du)	German (G)	French (Fr)	Spanish (Sp)	Italian (I)	Polish (P)	Hungarian (H)	Finnish (Fi)
one	en	en	een	eins	un	uno	uno	jeden	egy	yksi
two	to	to	twee	zwei	deux	dos	due	dwa	ketto	kaksi
three	tre	tre	drie	drei	trois	tres	tre	trzy	harom	kolme
four	fire	fire	vier	vier	quatre	cuatro	quattro	cztery	negy	neua
five	fem	fem	vijf	funf	cinq	cinco	cinque	piec	ot	viisi
six	seks	seks	zes	sechs	six	seis	sei	szesc	hat	kuusi
seven	sju	syv	zeven	sieben	sept	siete	sette	siedem	het	seitseman
eight	atte	otte	acht	acht	huit	ocho	otto	osiem	nyolc	kahdeksan
nine	ni	ni	negen	neun	neuf	nueve	nove	dziewiec	kilenc	yhdeksan
ten	ti	ti	tien	zehn	dix	diez	dieci	dziesiec	tiz	kymmenen

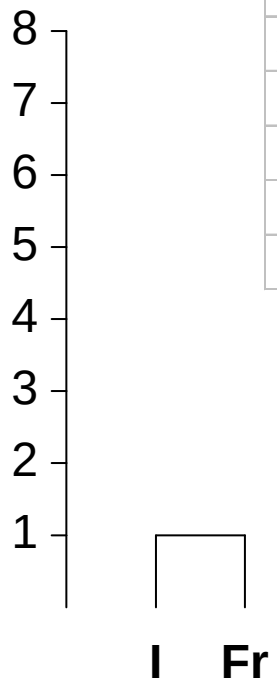
Distance: Frequency of numbers with different first letter e.g.

$d_{EN} = 2$ $d_{EDu} = 7$ $d_{SpI} = 1$

Intercluster strategy: SINGLE LINKAGE

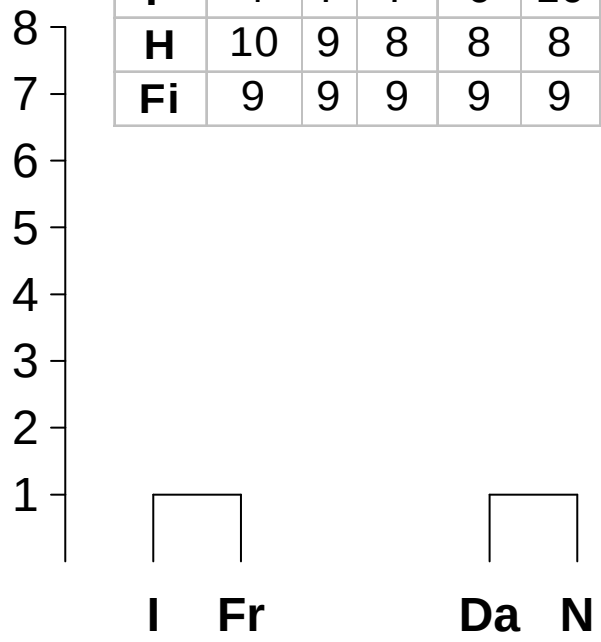
Iteration 1

	E	N	Da	Du	G	Fr	Sp	I	P	H	Fi
E	0										
N	2	0									
Da	2	1	0								
Du	7	5	6	0							
G	6	4	5	5	0						
Fr	6	6	6	9	7	0					
Sp	6	6	5	9	7	2	0				
I	6	6	5	9	7	1	1	0			
P	7	7	6	10	8	5	3	4	0		
H	9	8	8	8	9	10	10	10	10	0	
Fi	9	9	9	9	9	9	9	9	9	8	0



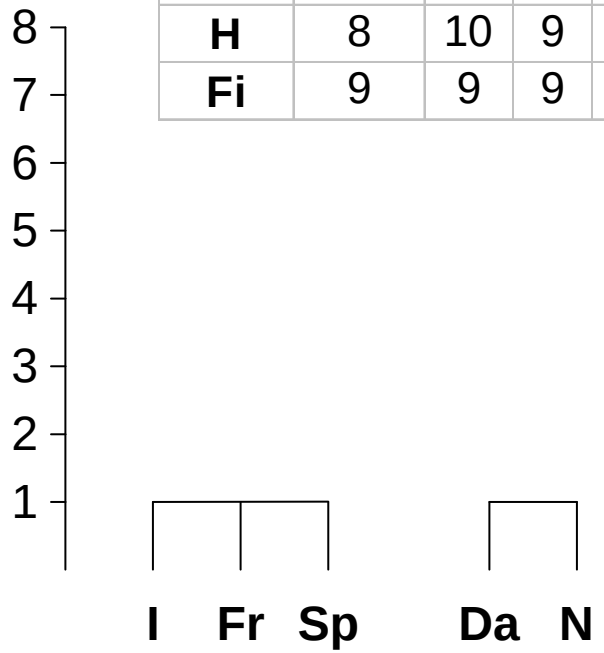
Iteration 2

	I Fr	E	N	Da	Du	G	Sp	P	H	Fi
I Fr	0									
E	6	0								
N	6	2	0							
Da	5	2	1	0						
Du	9	7	5	6	0					
G	7	6	4	5	5	0				
Sp	1	6	6	5	9	7	0			
P	4	7	7	6	10	8	3	0		
H	10	9	8	8	8	9	10	10	0	
Fi	9	9	9	9	9	9	9	9	8	0



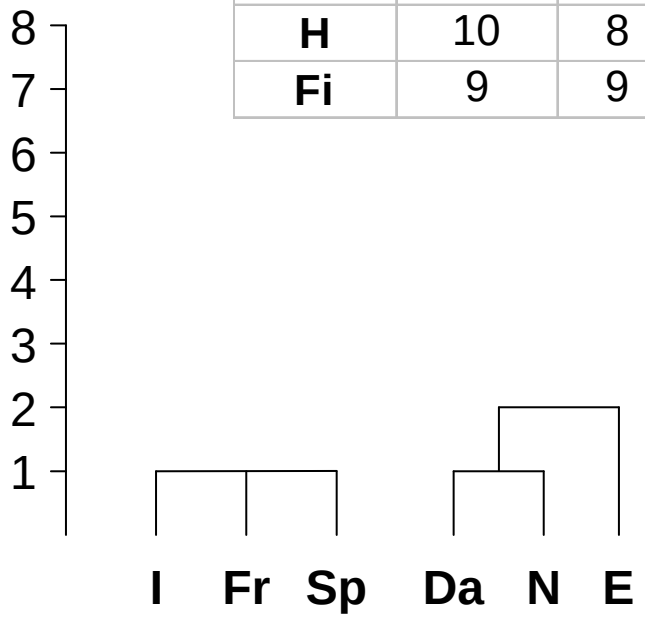
Iteration 3

	Da N	I Fr	E	Du	G	Sp	P	H	Fi
Da N	0								
I Fr	5	0							
E	2	6	0						
Du	5	9	7	0					
G	4	7	6	5	0				
Sp	5	1	6	9	7	0			
P	6	4	7	10	8	3	0		
H	8	10	9	8	9	10	10	0	
Fi	9	9	9	9	9	9	9	8	0



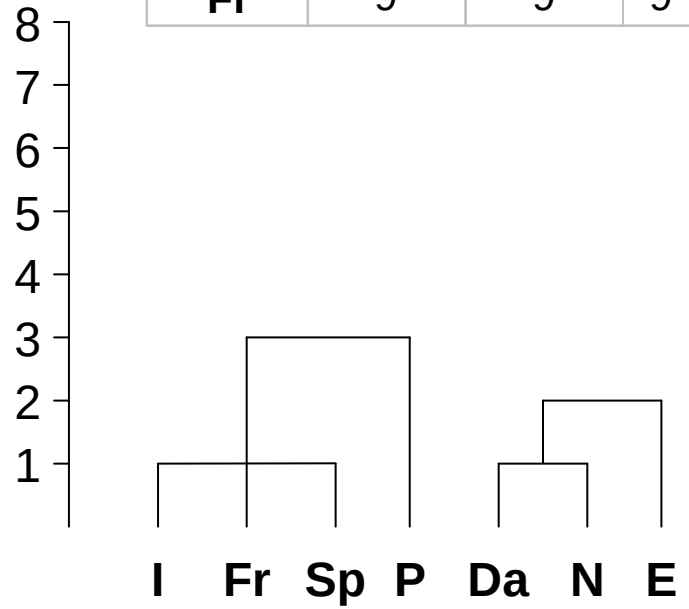
Iteration 4

	Sp I Fr	Da N	E	Du	G	P	H	Fi
Sp I Fr	0							
Da N	5	0						
E	6	2	0					
Du	9	5	7	0				
G	7	4	6	5	0			
P	3	6	7	10	8	0		
H	10	8	9	8	9	10	0	
Fi	9	9	9	9	9	9	8	0



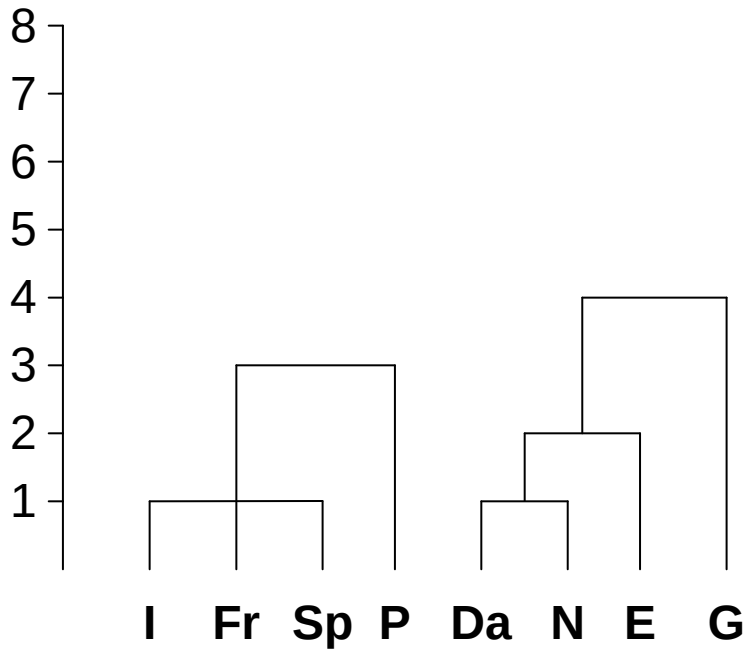
Iteration 5

	E Da N	Sp I Fr	Du	G	P	H	Fi
E Da N	0						
Sp I Fr	5	0					
Du	5	9	0				
G	4	7	5	0			
P	6	3	10	8	0		
H	8	10	8	9	10	0	
Fi	9	9	9	9	9	8	0



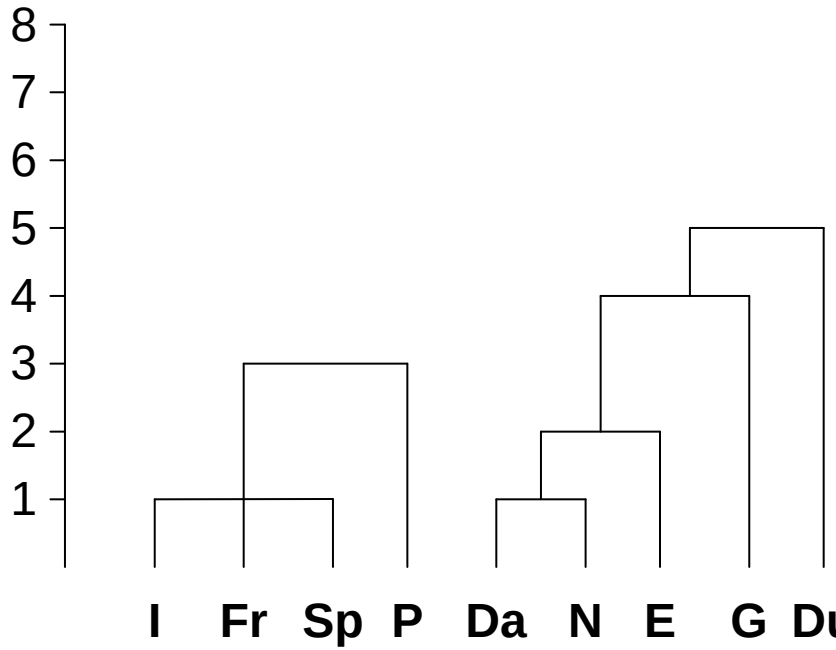
Iteration 6

	P Sp I Fr	E Da N	Du	G	H	Fi
P Sp I Fr	0					
E Da N	5	0				
Du	9	5	0			
G	7	4	5	0		
H	10	8	8	9	0	
Fi	9	9	9	9	8	0



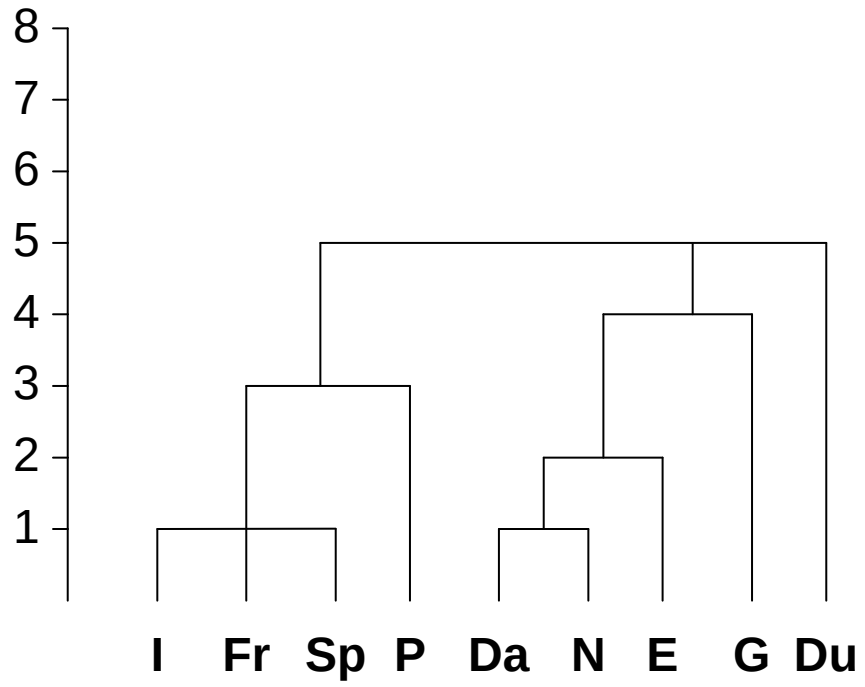
Iteration 7

	G E Da N	P Sp I Fr	Du	H	Fi
G E Da N	0				
P Sp I Fr	5	0			
Du	5	9	0		
H	8	10	8	0	
Fi	9	9	9	8	0



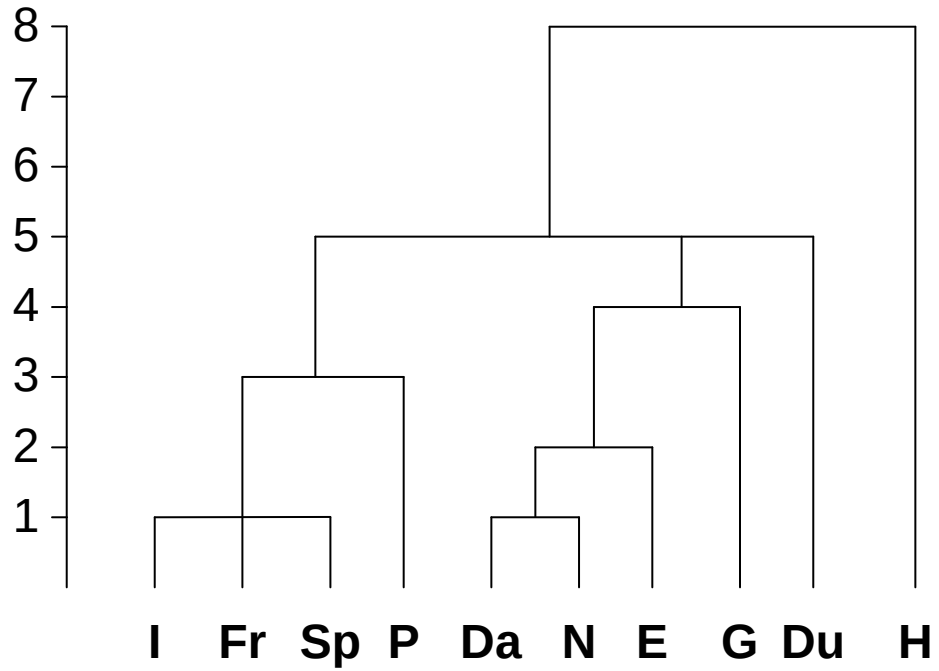
Iteration 8

	N	Fr	H	Fi
N	0			
P Sp I Fr	5	0		
H	8	10	0	
Fi	9	9	8	0



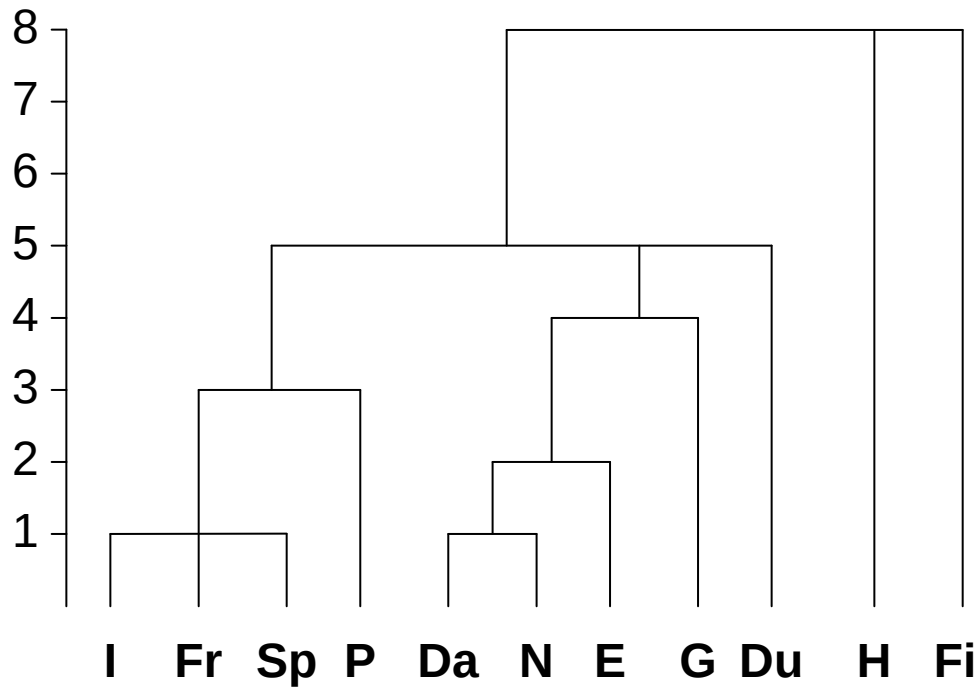
Iteration 9

	P Sp I Fr Du G E Da N	H	Fi
P Sp I Fr Du G E Da N	0		
H	8	0	
Fi	9	8	0



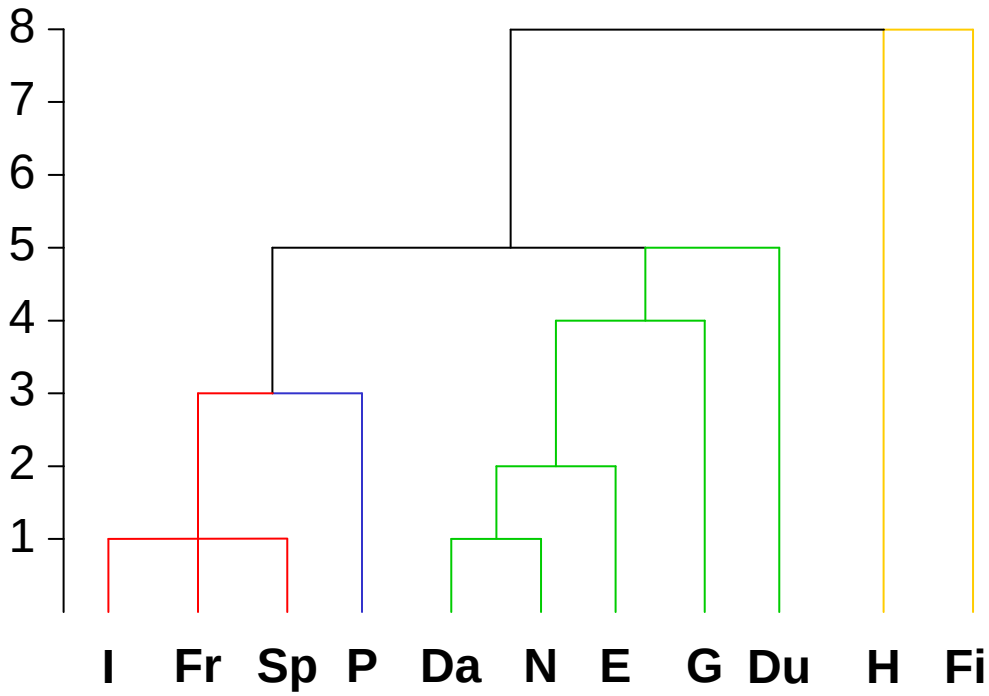
Iteration 10

	Fi H	P Sp I Fr Du G E Da N
Fi H	0	
P Sp I Fr Du G E Da N	8	0



Evaluation of clusters

• Clusters may be evaluated according to how well they describe current knowledge



Roman
Slavic
Germanic
Ugro-Finnish

Hierarchical vs. k-means

- Hierarchical clustering:
  - computationally expensive -> relatively small data sets
  - nice visualization, no. of clusters can be selected
  - deterministic
  - cannot correct early ”mistakes”
- K-means:
  - computationally efficient -> large data sets
  - predefined no. of clusters
  - non-deterministic -> should be run several times
  - iterative improvement
- Hierarchical k-means: top-down hierarchical clustering using k-means iteratively with k=2 -> best of both worlds!