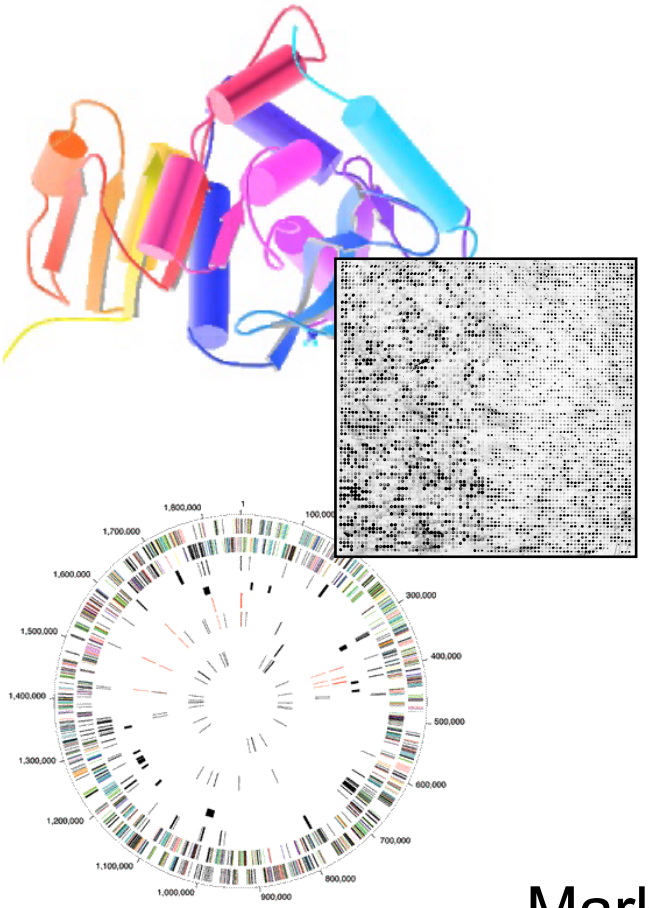
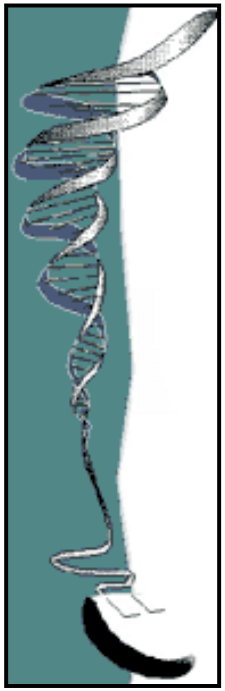
BIOINFORMATICS
Datamining #1





Mark Gerstein, Yale University

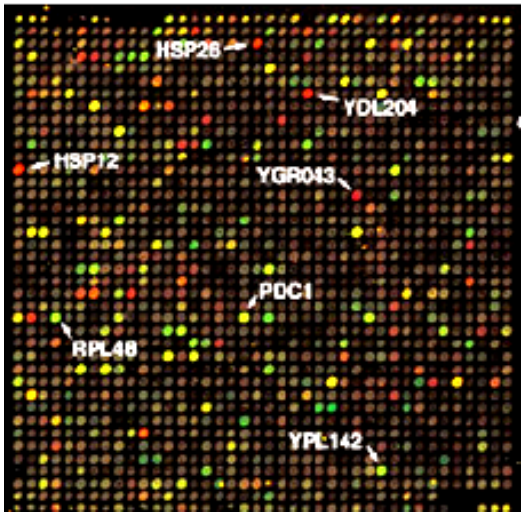
gersteinlab.org/courses/452

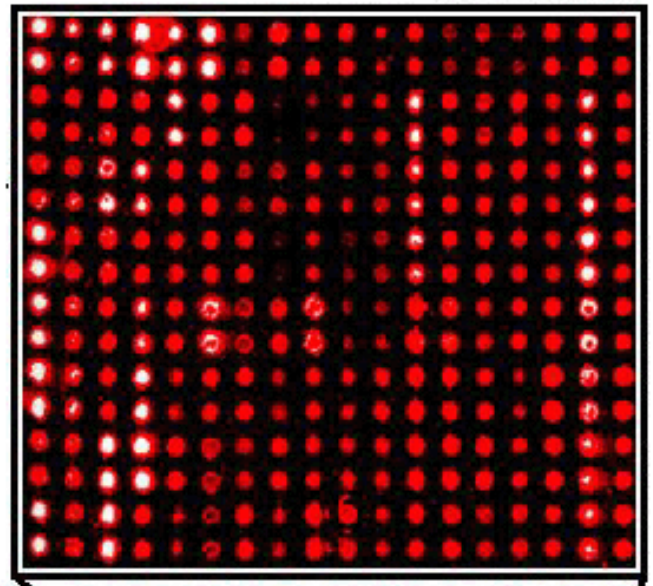
Large-scale Datamining

- Gene Expression
  - ◊ Representing Data in a Grid
  - ◊ Description of function prediction in abstract context
- Unsupervised Learning
  - ◊ clustering & k-means
  - ◊ Local clustering
- Supervised Learning
  - ◊ Discriminants & Decision Tree
  - ◊ Bayesian Nets
- Function Prediction EX
  - ◊ Simple Bayesian Approach for Localization Prediction

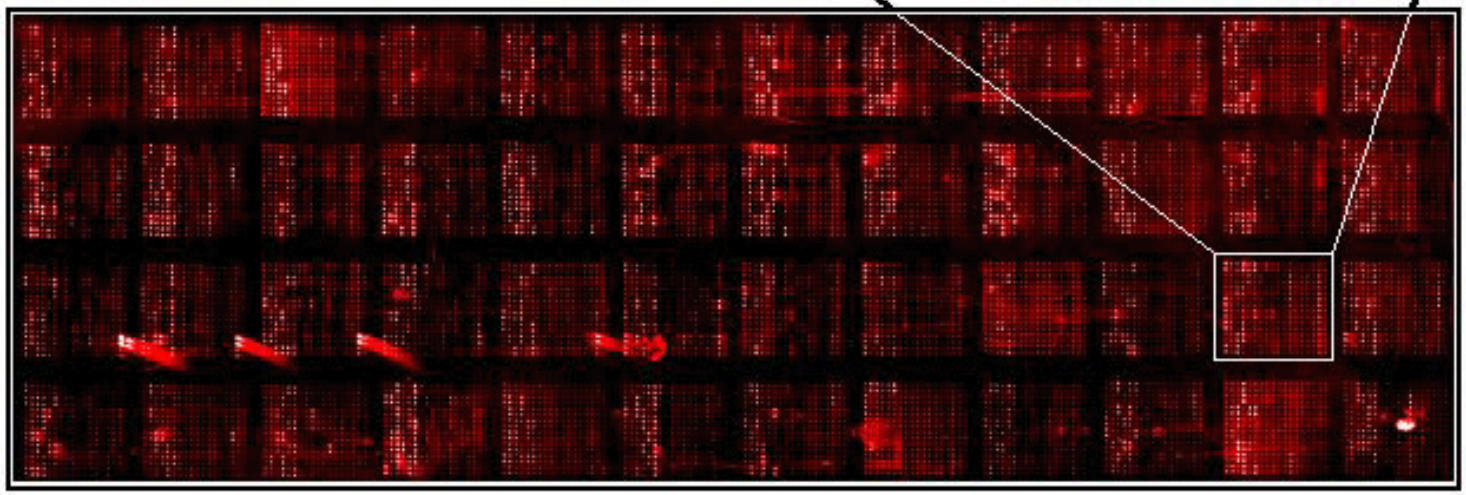
The recent advent and subsequent onslaught of microarray data

1st generation, Expression Arrays (Brown)





2nd gen., Proteome Chips (Snyder)

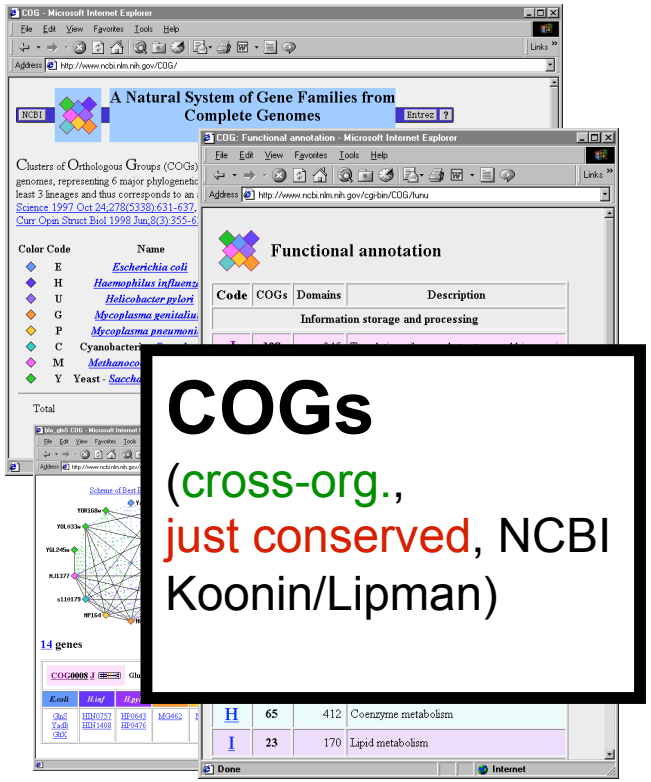


Gene Expression Information and Protein Features

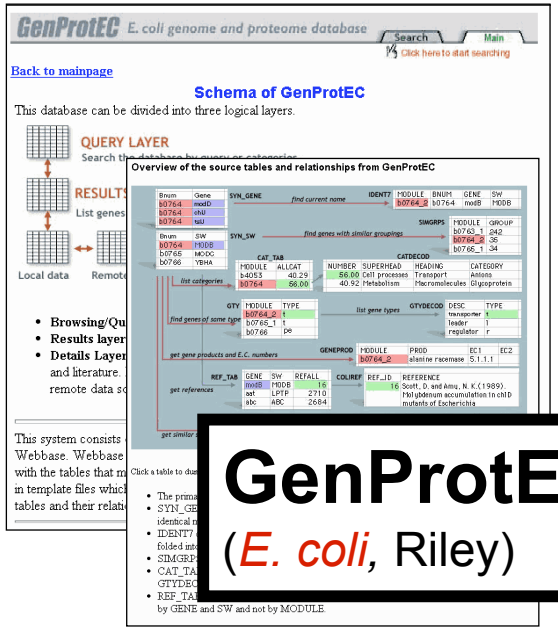
Basics		Predictors																																	
		Sequence Features																Genomic Features																	
		seq. length	Amino Acid Composition							How many times does the sequence have these motif features?							Abs. expr. Level (mRNA copies / cell)		Prot. Abun-dance	Cell cycle timecourse																	
Yeast Gene ID	Sequence		A	C	D	E	F	W	Y	farn site	NLS	hdel motif		nuc2	signalp	tms1	Gene-Chip expt. from RY Lab	sage tag freq.	(1000 copies /cell)	t=0	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12	t=13	t=14	t=15	t=16
YAL001C	M	1160	.08	.02	.06			.01	.04	0	1	0		1	0	0	0.3	0	?	5	3	4	4	5	4	3	5	5	3	5	7	9	4	4	4	5
YAL002W	K	1176	.09	.02	.06			.01	.04	0	0	0		0	0	1	0.2	?	?	8	4	2	3	4	3	4	5	5	3	4	4	6	4	5	4	3
YAL003W	K	206	.08	.02	.06			.01	.04	0	0	0		0	0	0	19.1	19	23	70	73	91	69	105	52	112	88	64	159	106	104	75	103	140	98	126
YAL004W	F	215	.08	.02	.06			.01	.04	0	0	0		0	0	0	?	0	?	18	12	9	5	5	3	6	4	4	3	3	5	5	4	5	4	6
YAL005C	V	641	.08	.02	.06			.01	.04	0	0	0		0	0	1	13.4	16	17	39	38	30	13	17	8	11	8	7	8	6	8	8	7	9	8	14
YAL007C	K	190	.08	.02	.06			.01	.04	0	0	0		0	1	4	2.2	8	?	15	20	32	20	21	19	29	19	16	22	20	26	23	22	25	16	17
YAL008W	H	198	.08	.02	.06			.01	.04	0	0	0		0	0	3	1.2	?	?	9	6	7	1	3	2	4	2	2	3	3	4	4	3	3	2	3
YAL009W	E	259	.08	.02	.06			.01	.04	0	2	0		0	0	3	0.6	?	?	6	2	4	3	5	3	5	5	5	3	4	6	6	4	4	3	5
YAL010C	M	493	.08	.02	.06			.02	.04	0	0	0		0	0	1	0.3	?	?	11	6	4	5	6	4	7	8	7	4	5	6	7	5	6	6	6
YAL011W	K	616	.08	.02	.06			.01	.04	0	8	0		1	0	0	0.4	?	?	6	5	4	4	8	5	8	8	6	6	5	6	6	7	6	5	6
YAL012W	G	393	.08	.02	.06			.01	.04	0	0	0		0	0	1	8.9	4	6.7	29	26	25	27	53	26	43	36	25	28	23	28	31	29	34	23	29
YAL013W	F	362	.08	.02	.06			.01	.04	0	0	0		0	0	0	0.6	?	?	7	9	6	5	14	6	12	14	10	9	9	9	10	9	8	6	10
YAL014C	G	202	.08	.02	.06			.01	.04	0	0	0		0	0	0	1.1	?	?	12	13	10	8	10	10	12	13	12	14	11	11	11	10	11	9	12
YAL015C	M	399	.08	.02	.06			.01	.04	0	1	0		0	0	0	0.7	0	1	19	18	14	10	14	12	17	17	14	13	11	13	16	11	14	12	13
YAL016W	K	635	.08	.02	.06			.01	.04	0	0	0		0	0	1	3.3	5	?	15	20	20	102	20	20	30	22	18	19	18	20	21	21	23	16	16
YAL017W	V	1356	.08	.02	.06			.01	.04	0	0	0		0	0	0	0.4	?	?	14	3	3	4	8	5	6	6	5	5	8	9	10	6	5	4	7
YAL018C	K	325	.08	.02	.06			.01	.04	0	0	0		0	0	4	?	?	?	4	2	2	2	1	1	2	2	2	1	2	1	2	2	1	2	1

...

Functional Classification

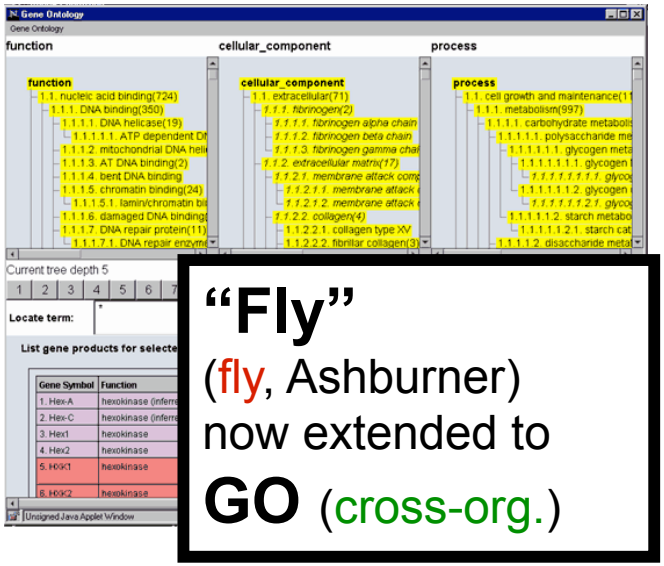


COGs
(cross-org., just conserved, NCBI Koonin/Lipman)

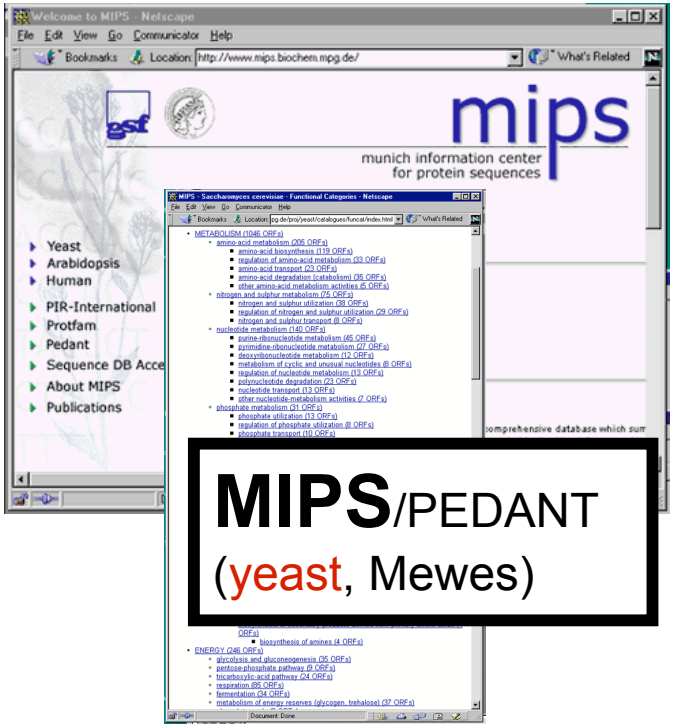


GenProtEC
(*E. coli*, Riley)

ENZYME
(SwissProt Bairoch/ Apweiler, just enzymes, cross-org.)



"Fly"
(fly, Ashburner) now extended to **GO** (cross-org.)



MIPS/PEDANT
(yeast, Mewes)

Also:

Other SwissProt Annotation

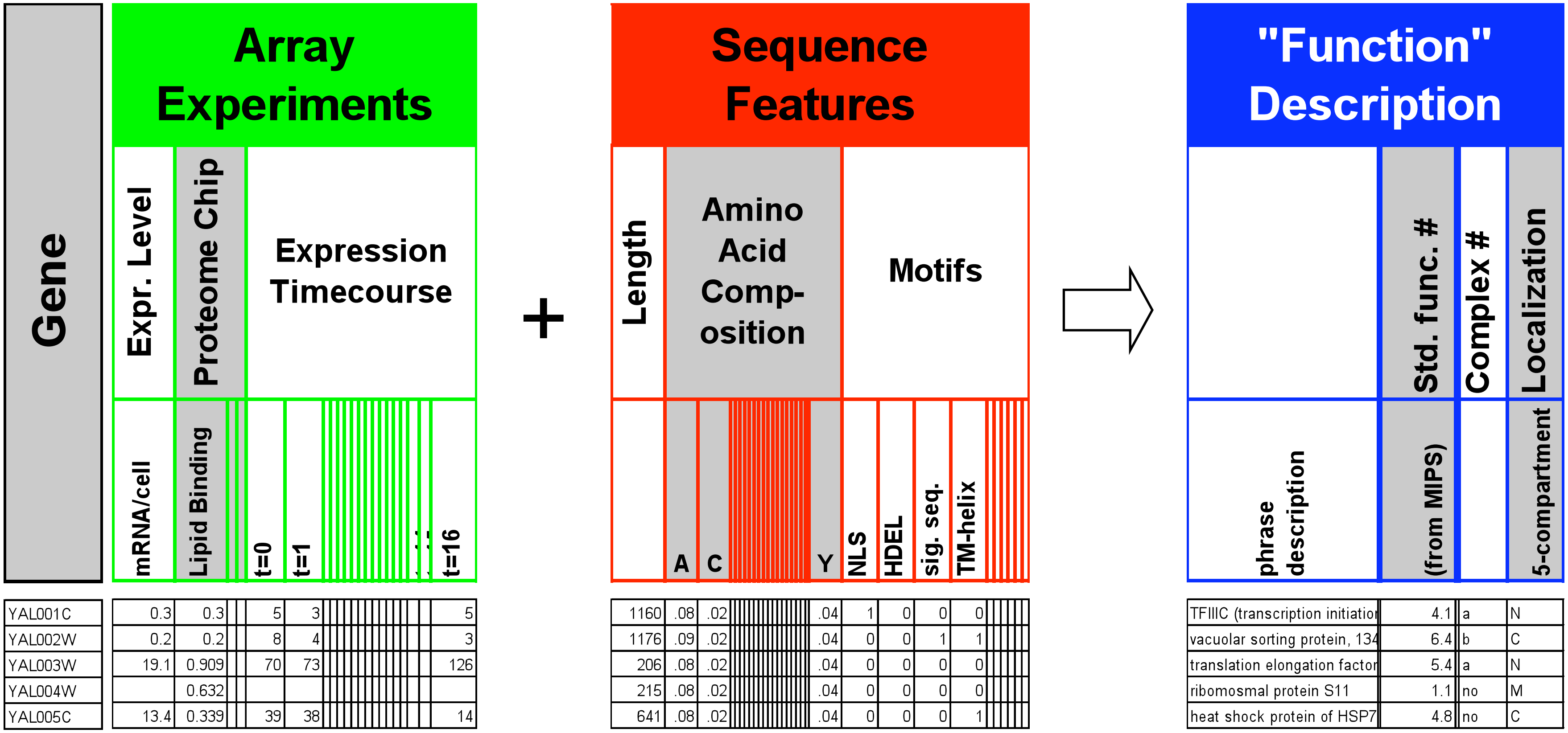
WIT, KEGG (just pathways)

TIGR EGAD (human ESTs)

SGD (yeast)

Cor

Prediction of Function on a Genomic Scale from Array Data & Sequence Features



Gene

Array Experiments: Expr. Level (mRNA/cell), Proteome Chip (Lipid Binding), Expression Timecourse (t=0, t=1, … t=16)

Gene	mRNA/cell	Lipid Binding	t=0	t=1	t=16
YAL001C	0.3	0.3	5	3	5
YAL002W	0.2	0.2	8	4	3
YAL003W	19.1	0.909	70	73	126
YAL004W		0.632			
YAL005C	13.4	0.339	39	38	14

\+

Sequence Features: Length, Amino Acid Comp-osition (A, C, … Y), Motifs (NLS, HDEL, sig. seq., TM-helix)

Length	A	C	Y	NLS	HDEL	sig. seq.	TM-helix
1160	.08	.02	.04	1	0	0	0
1176	.09	.02	.04	0	0	1	1
206	.08	.02	.04	0	0	0	0
215	.08	.02	.04	0	0	0	0
641	.08	.02	.04	0	0	0	1

"Function" Description

phrase description	Std. func. # (from MIPS)	Complex #	Localization 5-compartment
TFIIIC (transcription initiatior	4.1	a	N
vacuolar sorting protein, 134	6.4	b	C
translation elongation factor	5.4	a	N
ribomosmal protein S11	1.1	no	M
heat shock protein of HSP7	4.8	no	C

6000+

Different Aspects of function: molecular action, cellular role, phenotypic manifestation

Also: localization, interactions, complexes

Arrange data in a tabulated form, each row representing an example and each column representing a feature, including the dependent experimental quantity to be predicted.

	predictor1	Predictor2	predictor3	predictor4	response
G1	A(1,1)	A(1,2)	A(1,3)	A(1,4)	Class A
G2	A(2,1)	A(2,2)	A(2,3)	A(2,4)	Class A
G3	A(3,1)	A(3,2)	A(3,3)	A(3,4)	Class B

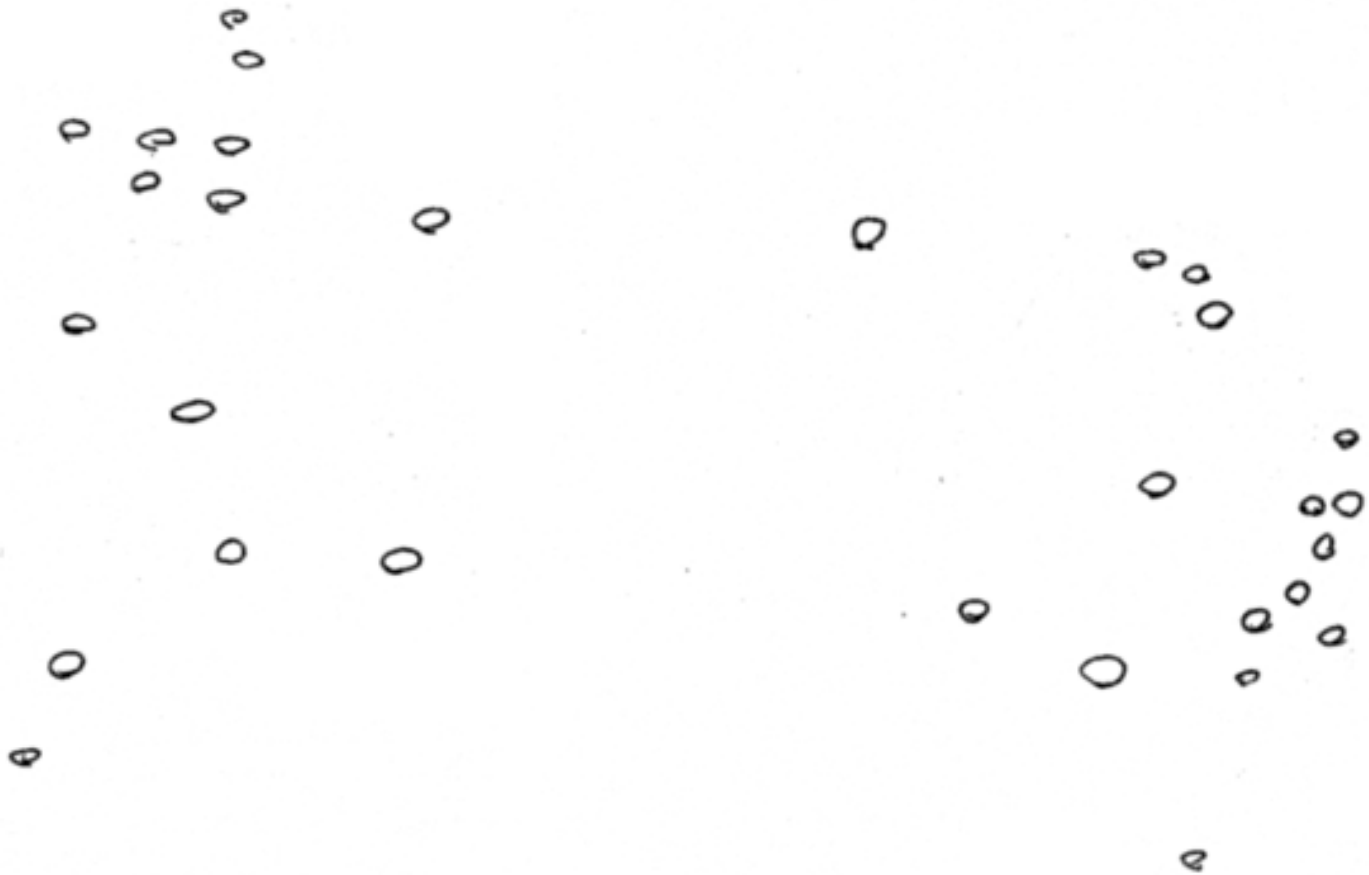
(adapted from Y Kluger)

Typical Predictors and Response for Yeast

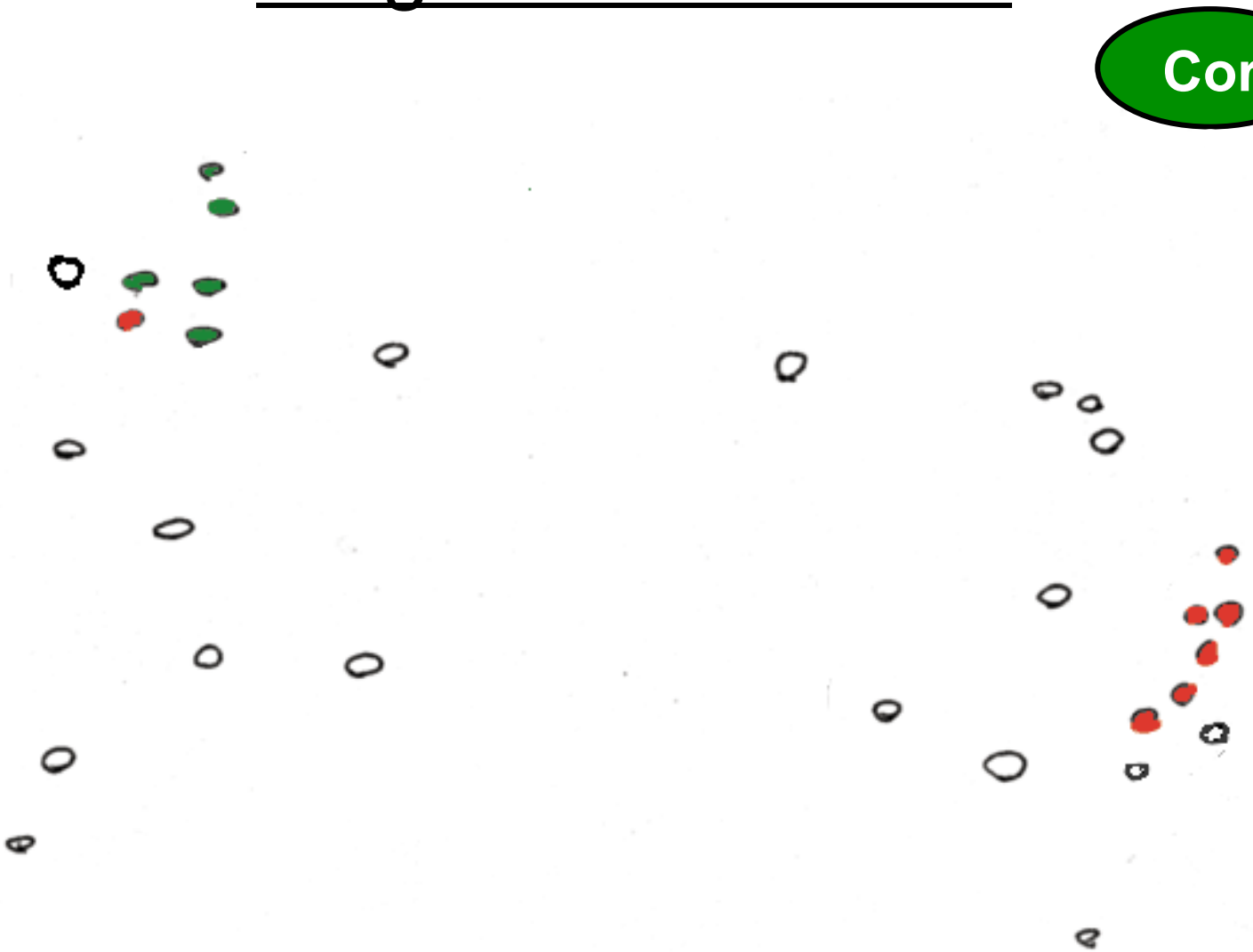
Basics		Predictors																						Response		
		Sequence Features															Genomic Features									
		seq. length	Amino Acid Composition							How many times does the sequence have these motif features?							Abs. expr. Level (mRNA copies / cell)		Prot. Abun-dance	Cell cycle timecourse				Function		Localization
Yeast Gene ID	Sequence		A	C	D	E	F	W	Y	farn site	NLS	hdel motif		nuc2	signalp	tms1	Gene-Chip expt. from RY Lab	sage tag freq.	(1000 copies /cell)	t=0	t=1		t=15	t=16	function ID(s) (from MIPS)	function description	5-compartment
YAL001C	MNIFEMLRI	1160	.08	.02	.06			.01	.04	0	1	0		1	0	0	0.3	0	?	5	3		4	5	04.01.01;04.03	TFIIIC (transcription initi	N
YAL002W	KVFGRCELA	1176	.09	.02	.06			.01	.04	0	0	0		0	0	1	0.2	?	?	8	4		4	3	06.04;08.13	vacuolar sorting protein,	C
YAL003W	KMLQFNLRW	206	.08	.02	.06			.01	.04	0	0	0		0	0	0	19.1	19	23	70	73		98	126	05.04;30.03	translation elongation fac	N
YAL004W	RPDFCLEPP	215	.08	.02	.06			.01	.04	0	0	0		0	0	0	?	0	?	18	12		4	6	01.01.01	0	N
YAL005C	VINTFDGVA	641	.08	.02	.06			.01	.04	0	0	0		0	0	1	13.4	16	17	39	38		8	14	06.01;06.04;08	heat shock protein of HS	????
YAL007C	KKAVINGEQ	190	.08	.02	.06			.01	.04	0	0	0		0	1	4	2.2	8	?	15	20		16	17	99	????	????
YAL008W	HPETLVKVK	198	.08	.02	.06			.01	.04	0	0	0		0	0	3	1.2	?	?	9	6		2	3	99	????	????
YAL009W	PTLEWFLSH	259	.08	.02	.06			.01	.04	0	2	0		0	0	3	0.6	?	?	6	2		3	5	03.10;03.13	meiotic protein	????
YAL010C	MEQRITLKD	493	.08	.02	.06			.02	.04	0	0	0		0	0	1	0.3	?	?	11	6		6	6	30.16	involved in mitochondrial	????
YAL011W	KSFPEVVGK	616	.08	.02	.06			.01	.04	0	8	0		1	0	0	0.4	?	?	6	5		5	6	30.16;99	protein of unknown funct	????
YAL012W	GVQVETISP	393	.08	.02	.06			.01	.04	0	0	0		0	0	1	8.9	4	6.7	29	26		23	29	01.01.01;30.03	cystathionine gamma-lya	C
YAL013W	RTDCYGNVN	362	.08	.02	.06			.01	.04	0	0	0		0	0	0	0.6	?	?	7	9		6	10	01.06.10;30.03	regulator of phospholipid	N
YAL014C	GDVEKGKKI	202	.08	.02	.06			.01	.04	0	0	0		0	0	0	1.1	?	?	12	13		9	12	99	????	N
YAL015C	MTPAVTTYK	399	.08	.02	.06			.01	.04	0	1	0		0	0	0	0.7	0	1	19	18		12	13	11.01;11.04	DNA repair protein	N
YAL016W	KKPLTQEQL	635	.08	.02	.06			.01	.04	0	0	0		0	0	1	3.3	5	?	15	20		16	16	03.01;03.04;03	ser/thr protein phosphata	????

Represent predictors in abstract high dimensional space

Cor

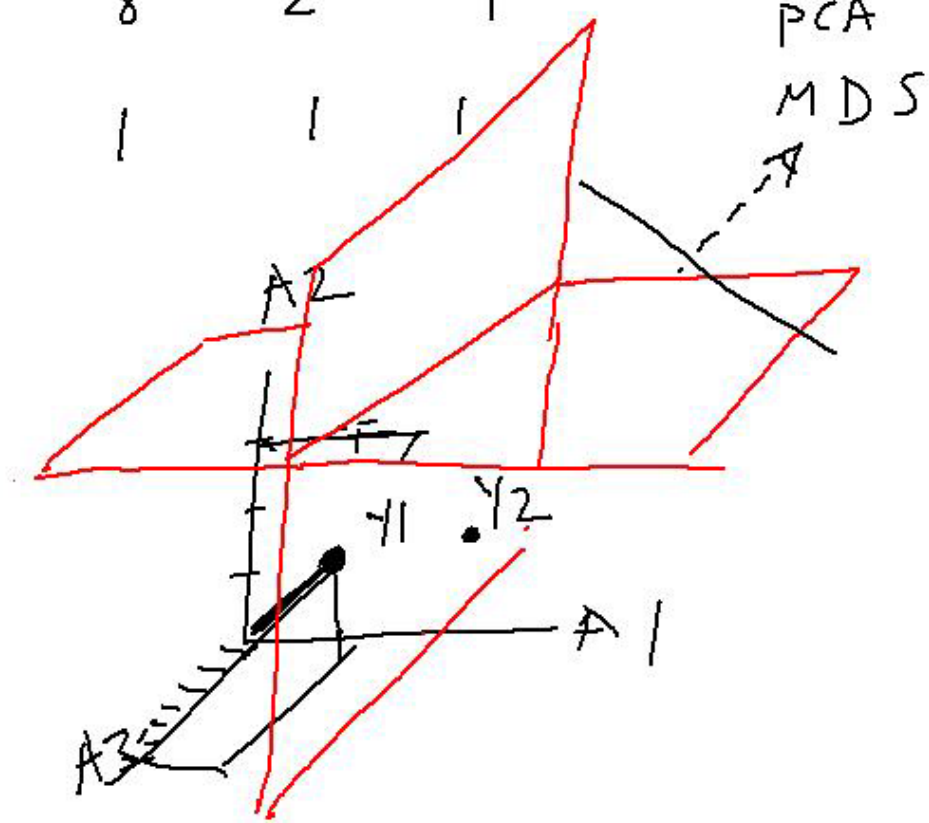


“Tag” Certain Points



Abstract high-dimensional space representation

	A1	A2	A3	A4 ...
Y1	1	3	7	
Y2	8	2	1	
Y3	1	1	1	
...				

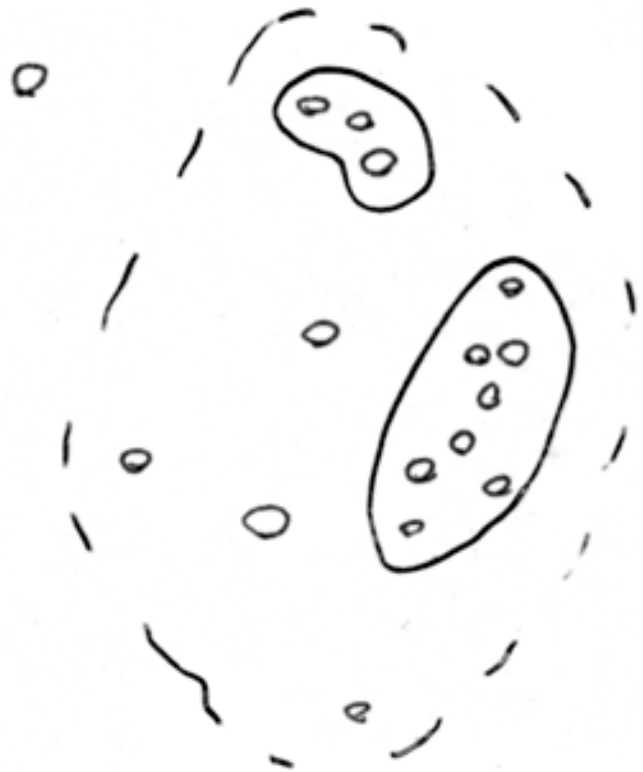


Large-scale Datamining

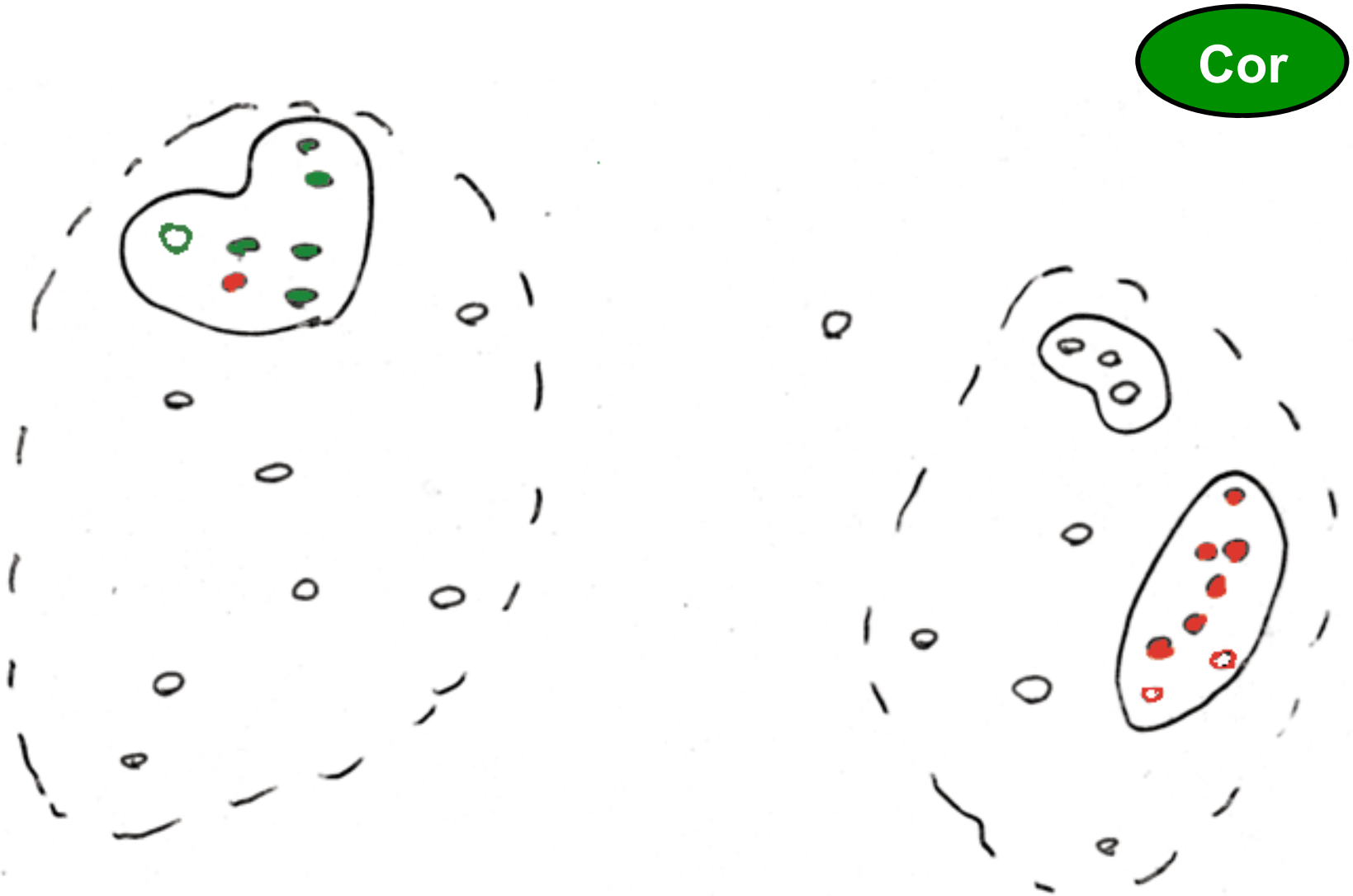
- Gene Expression
  - ◊ Representing Data in a Grid
  - ◊ Description of function prediction in abstract context
- Unsupervised Learning
  - ◊ clustering & k-means
  - ◊ Local clustering
- Supervised Learning
  - ◊ Discriminants & Decision Tree
  - ◊ Bayesian Nets
- Function Prediction EX
  - ◊ Simple Bayesian Approach for Localization Prediction

“cluster” predictors

Cor



Use clusters to predict Response

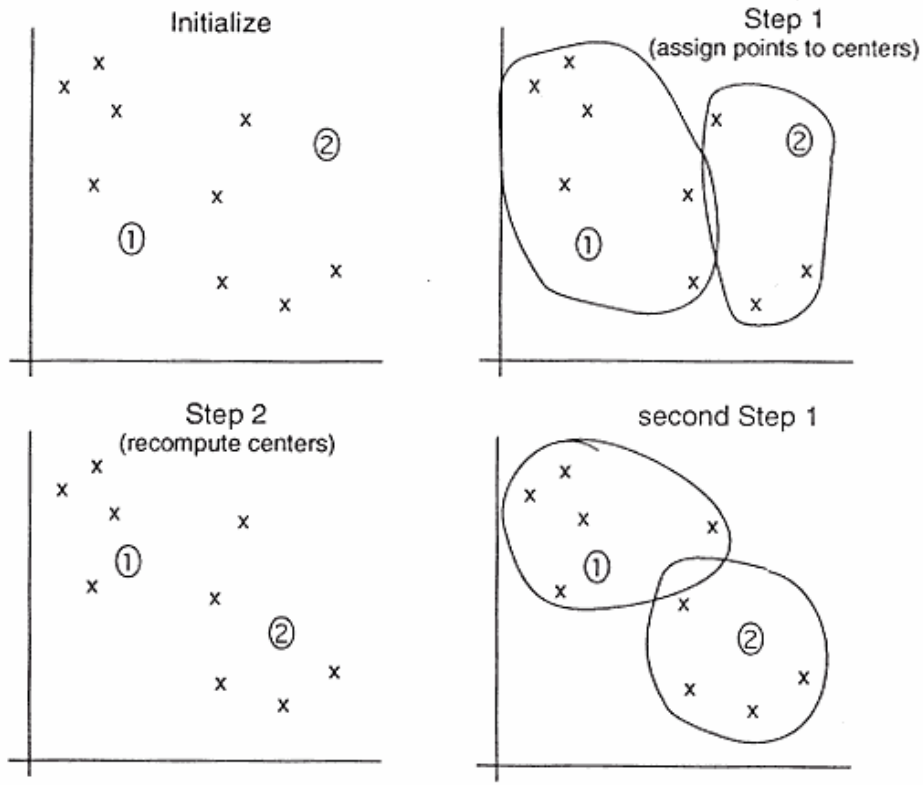


K-means

Cor

HRI Heuristicrats Research, Inc.

K-means algorithm in 2-D clustering



6

K-means

Top-down vs. Bottom up

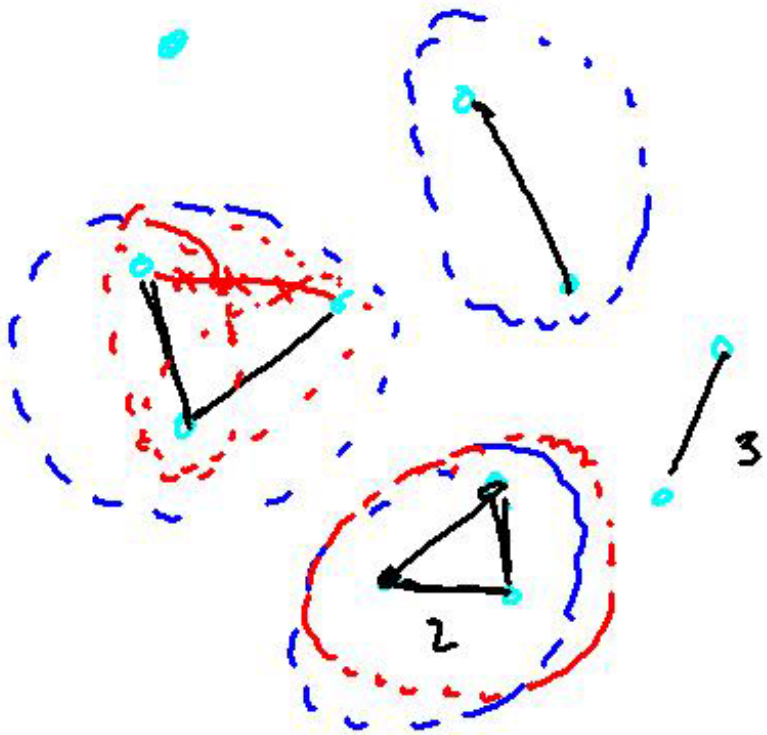
Top-down when you know how many subdivisions

k-means as an example of top-down

1) Pick ten (i.e. k?) random points as putative cluster centers.
2) Group the points to be clustered by the center to which they are closest.
3) Then take the mean of each group and repeat, with the means now at the cluster center.
4) I suppose you stop when the centers stop moving.

Bottom up clustering

Cor



SINGLE LINK

MULTI LINK

(THRESHOLD)

BOTTOM UP

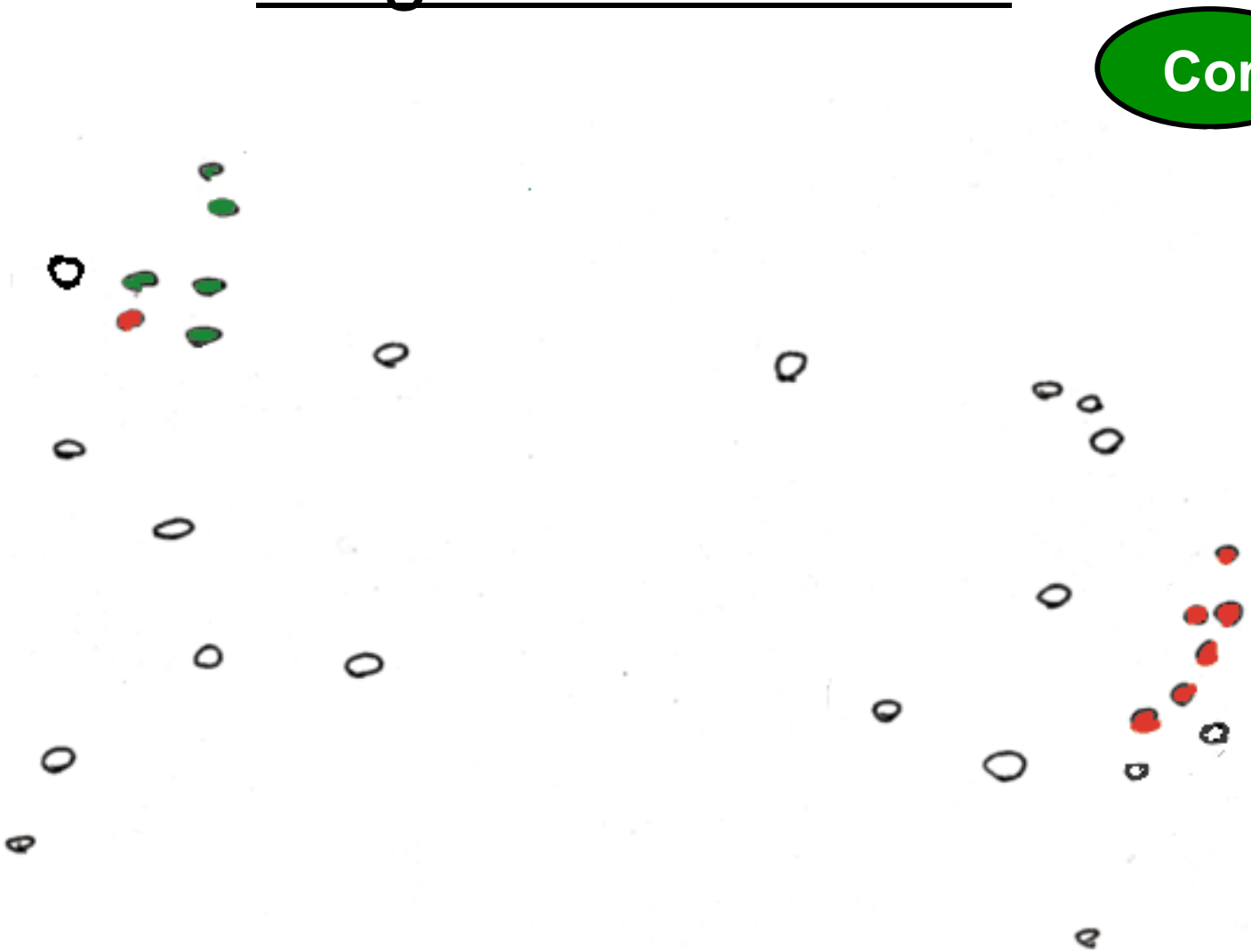
TOP-DOWN

K MEANS

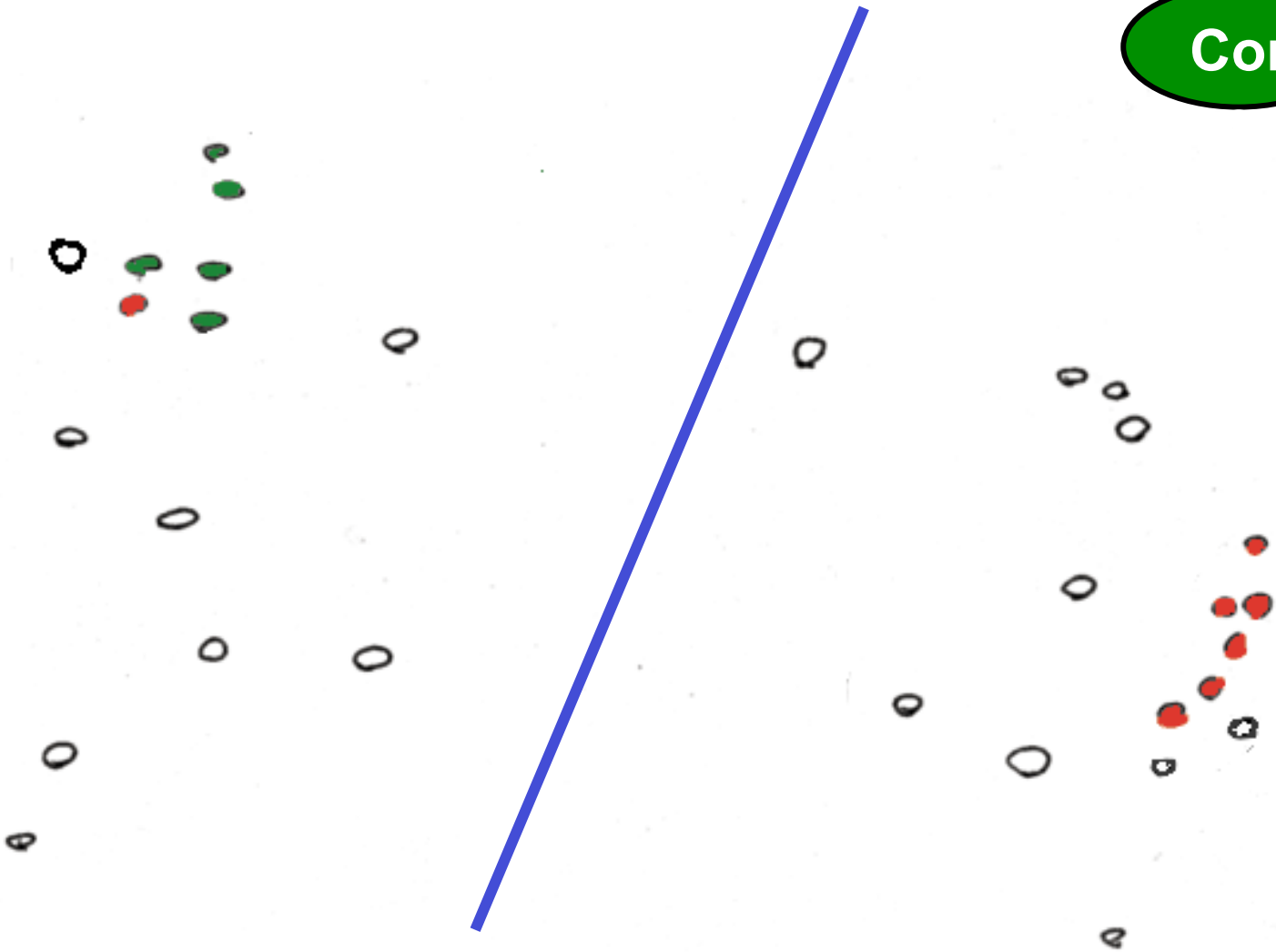
Large-scale Datamining

- Gene Expression
  - ◊ Representing Data in a Grid
  - ◊ Description of function prediction in abstract context
- Unsupervised Learning
  - ◊ clustering & k-means
  - ◊ Local clustering
- Supervised Learning
  - ◊ Discriminants & Decision Tree
  - ◊ Bayesian Nets
- Function Prediction EX
  - ◊ Simple Bayesian Approach for Localization Prediction

“Tag” Certain Points

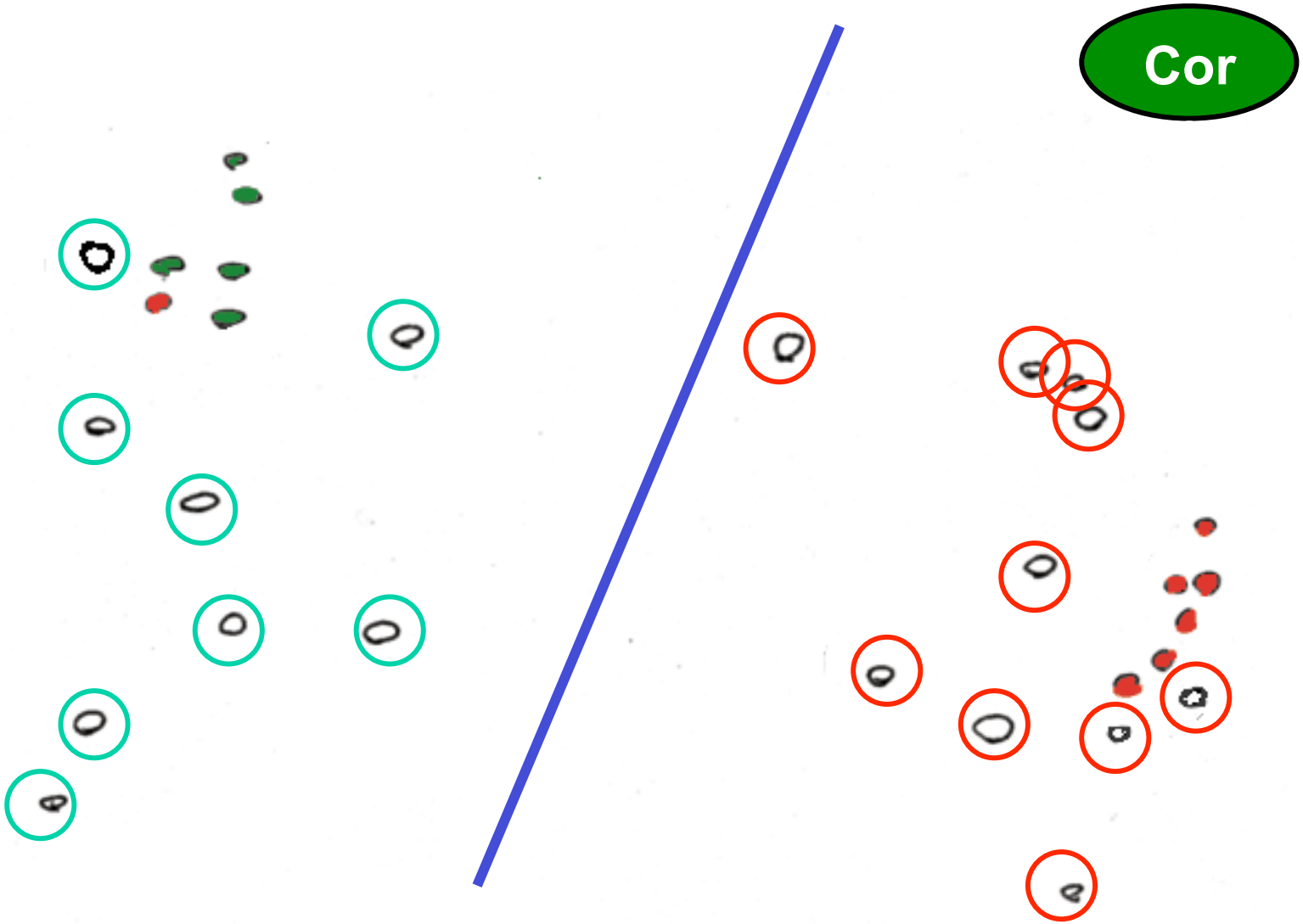


Find a Division to Separate Tagged Points

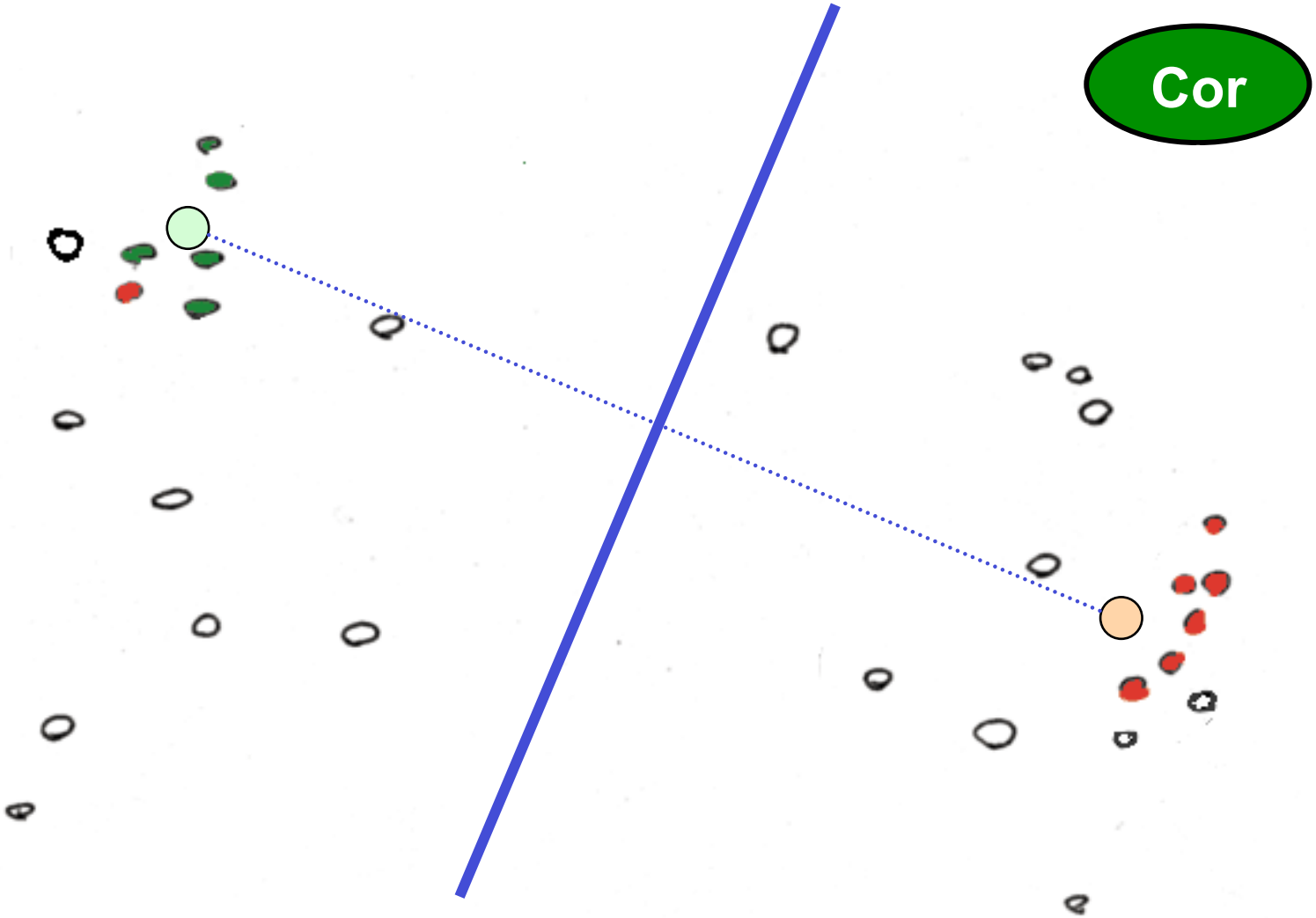


Cor

Extrapolate to Untagged Points



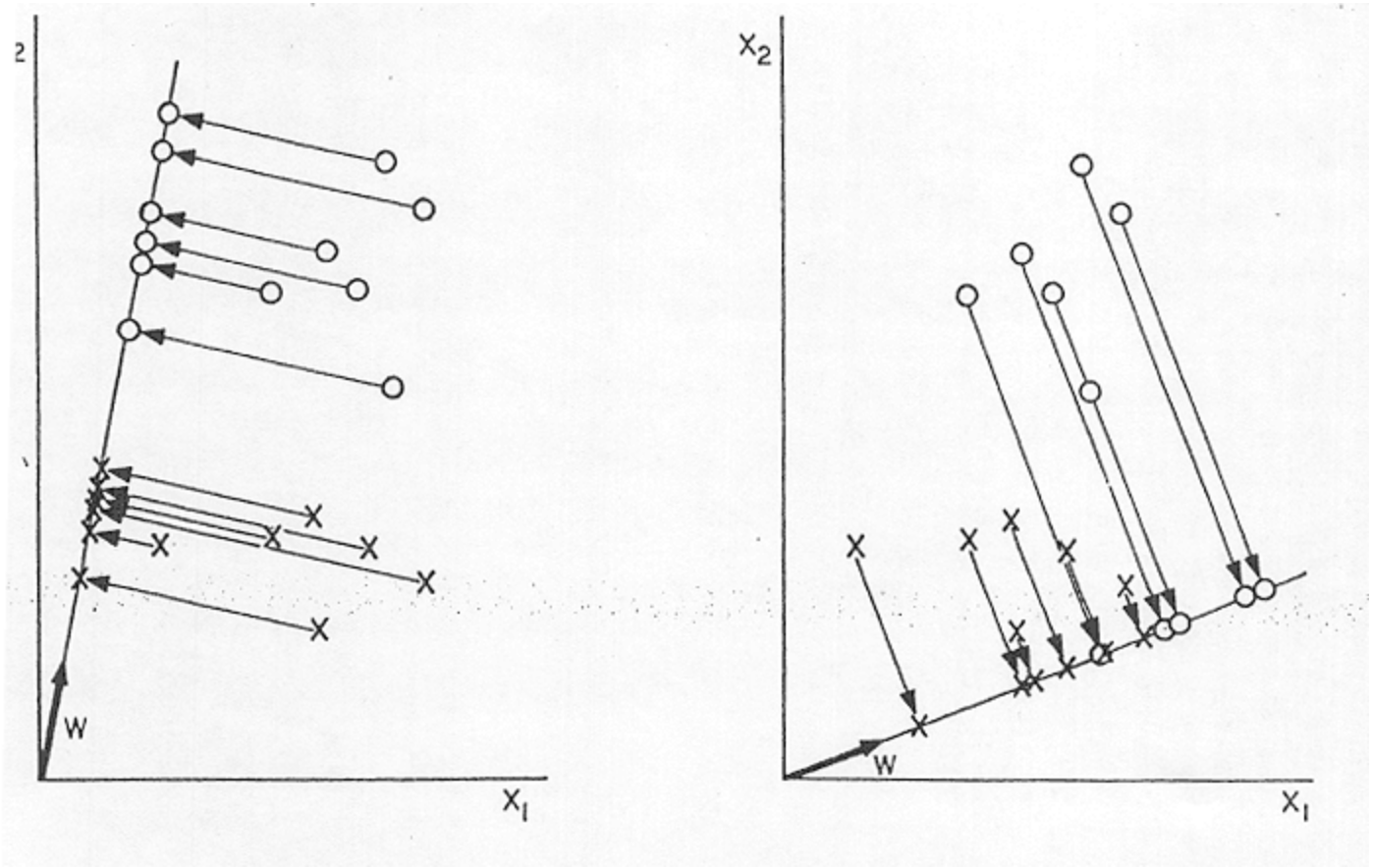
Discriminant to Position Plane



Fisher discriminant analysis

- Use the training set to reveal the structure of class distribution by seeking a linear combination
- $y = w_1x_1 + w_2x_2 + ... + w_nx_n$ which maximizes the ratio of the separation of the class means to the sum of each class variance (within class variance). This linear combination is called the first linear discriminant or first canonical variate. Classification of a future case is then determined by choosing the nearest class in the space of the first linear discriminant and significant subsequent discriminants, which maximally separate the class means and are constrained to be uncorrelated with previous ones.

Fischer's Discriminant



(Adapted from ???)

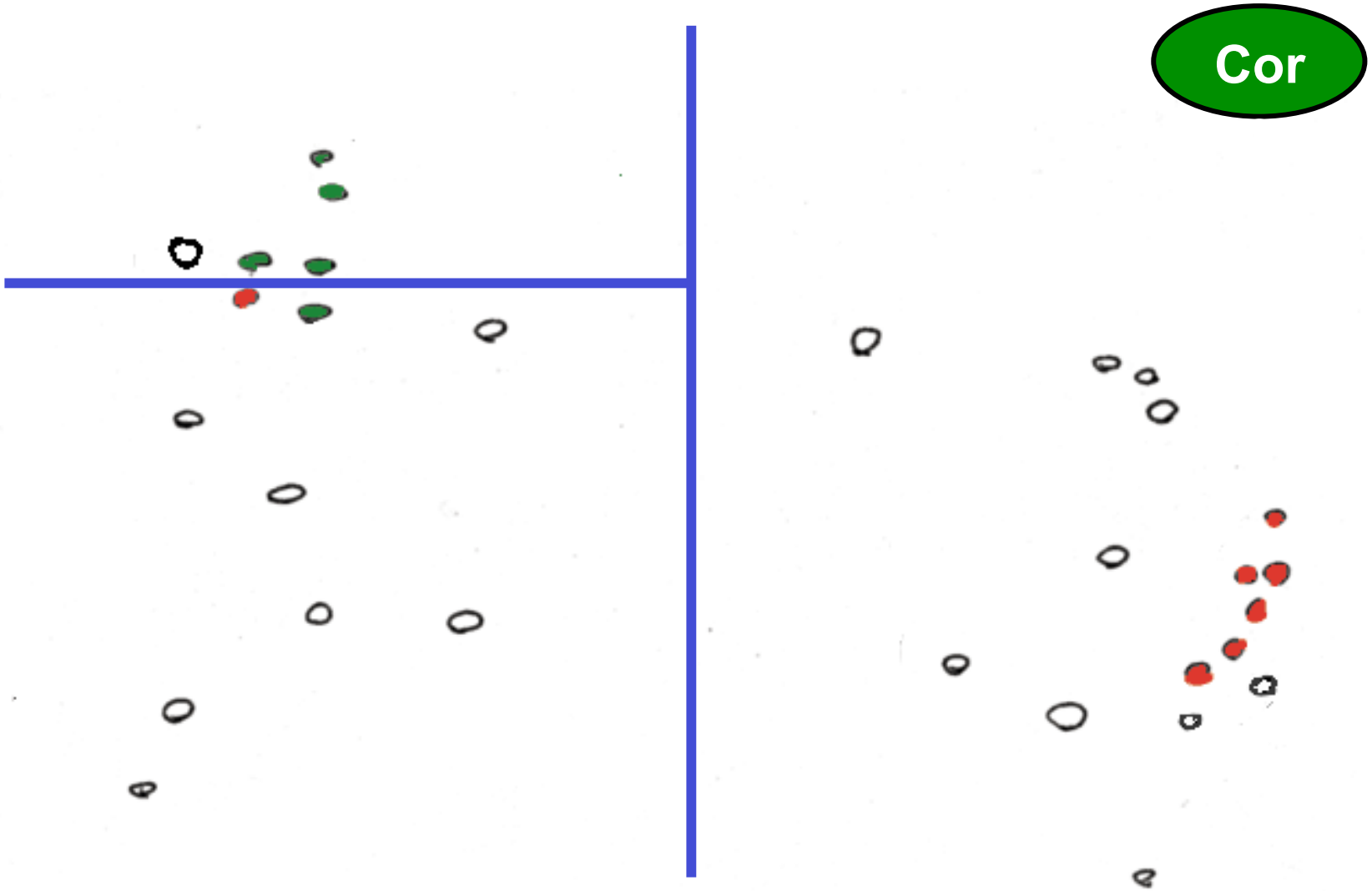
Fisher cont.

$$m_i = \vec{w} \cdot \vec{m}_i \qquad s_i^2 = \sum_{y \in Y_i} (y - m_i)^2$$

Solution of 1st variate

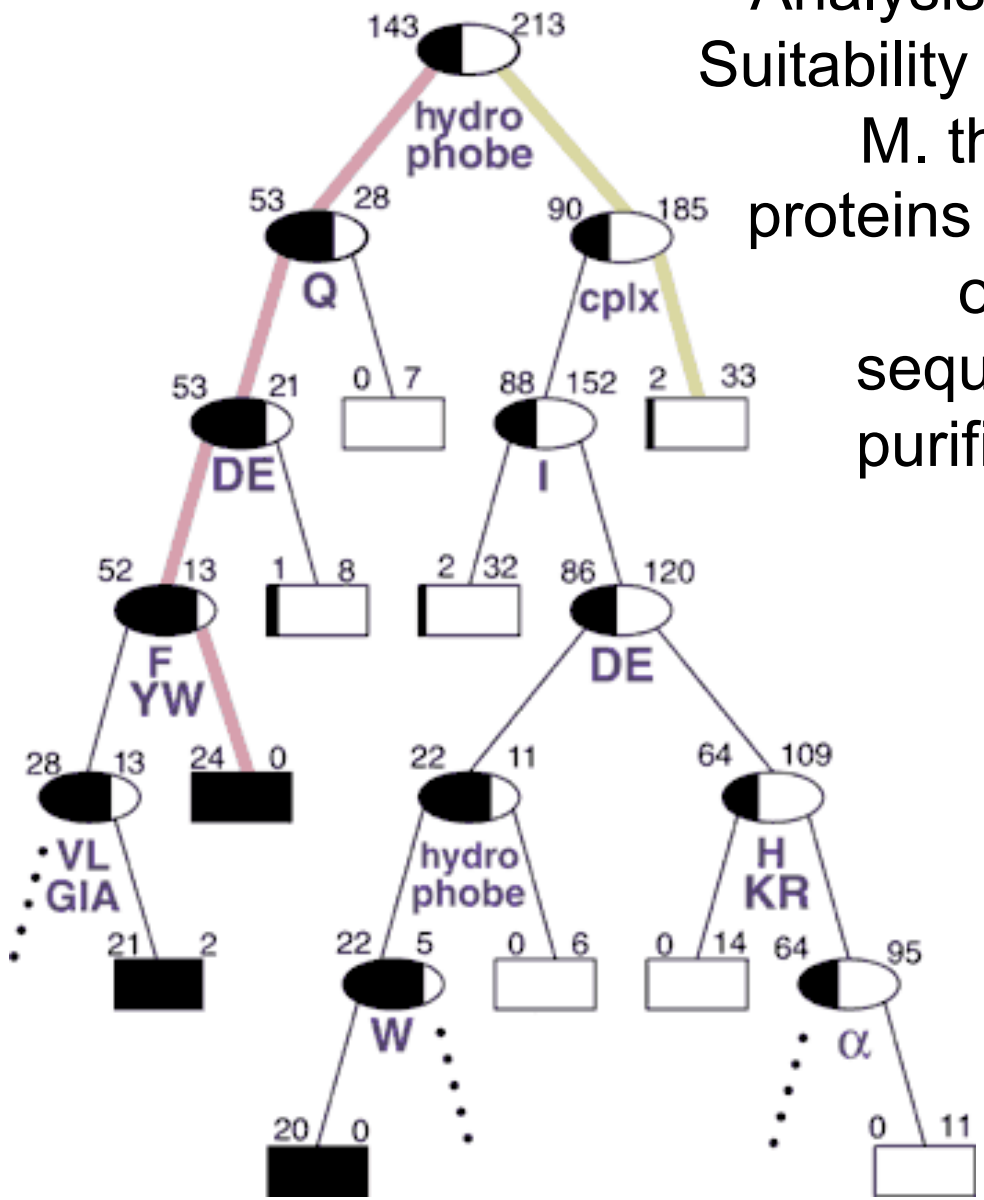
$$\vec{w} = S_W^{-1} (\vec{m}_1 - \vec{m}_2)$$

Find a Division to Separate Tagged Points



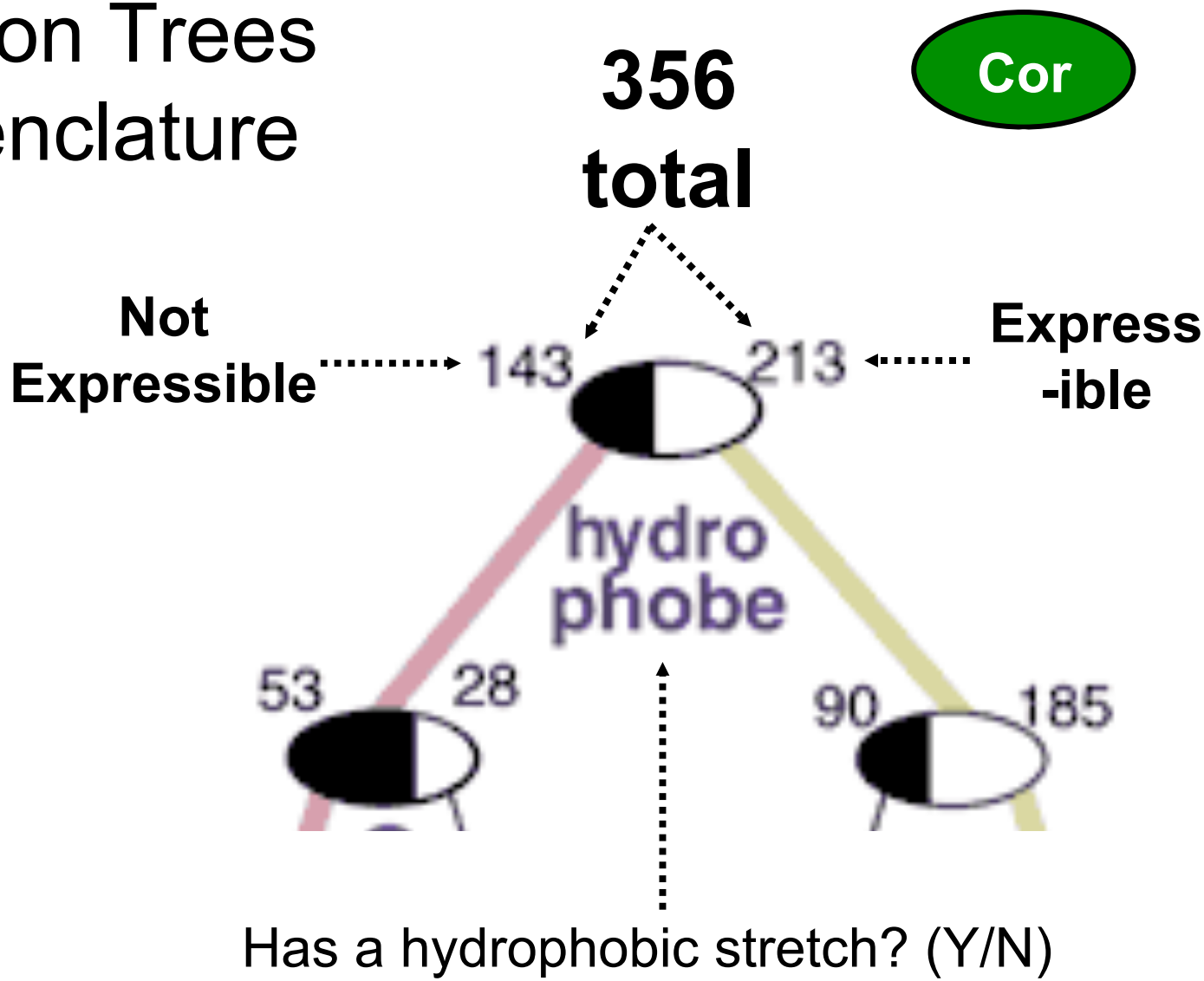
Retrospective Decision Trees

Analysis of the Suitability of 500 M. thermo. proteins to find optimal sequences purification

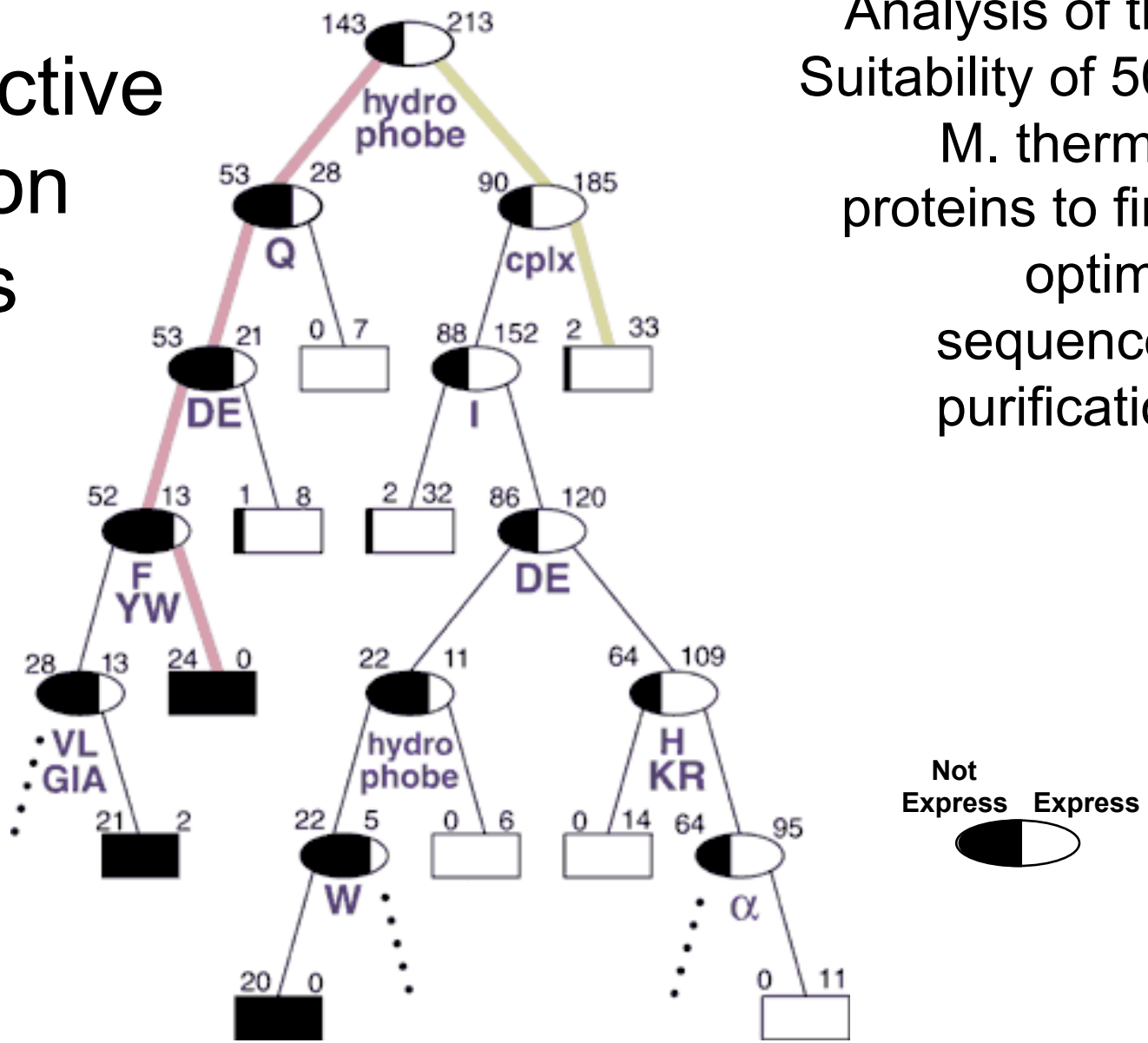


Cor

Retrospective Decision Trees Nomenclature

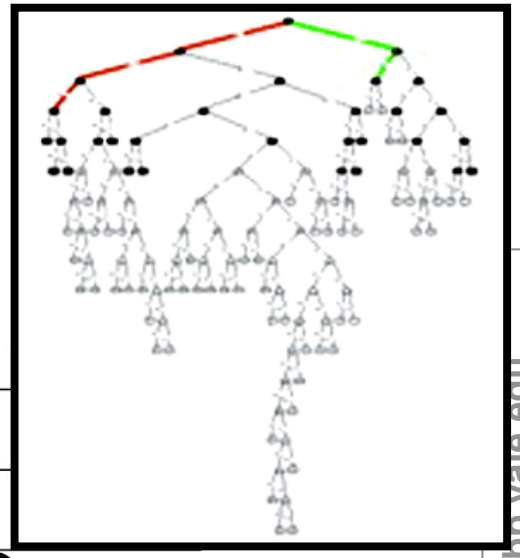


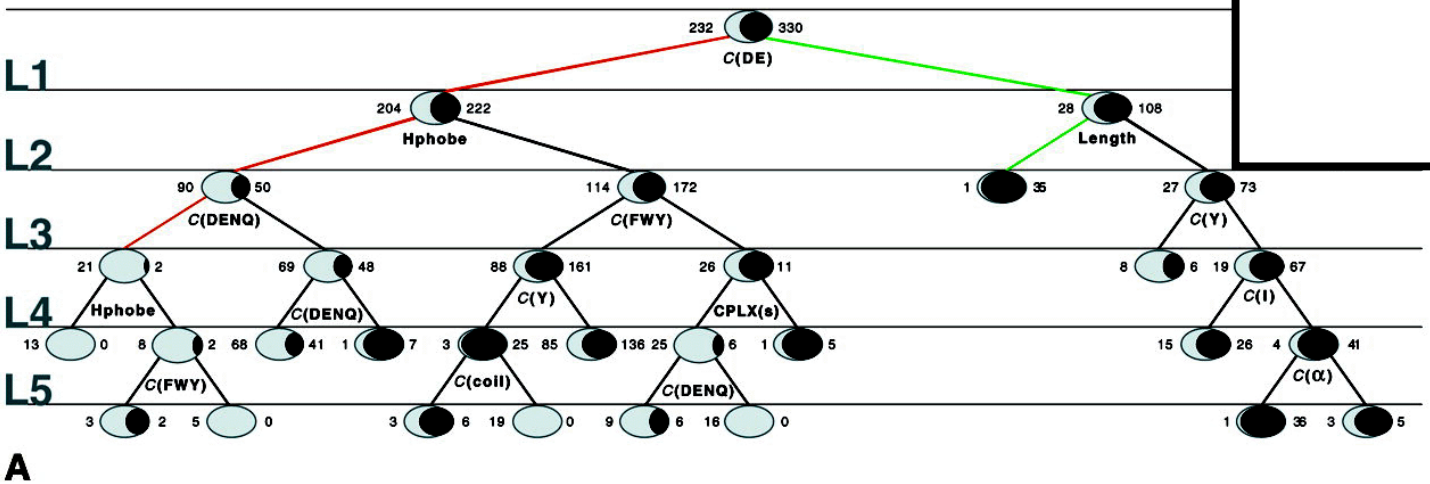
Retrospective Decision Trees



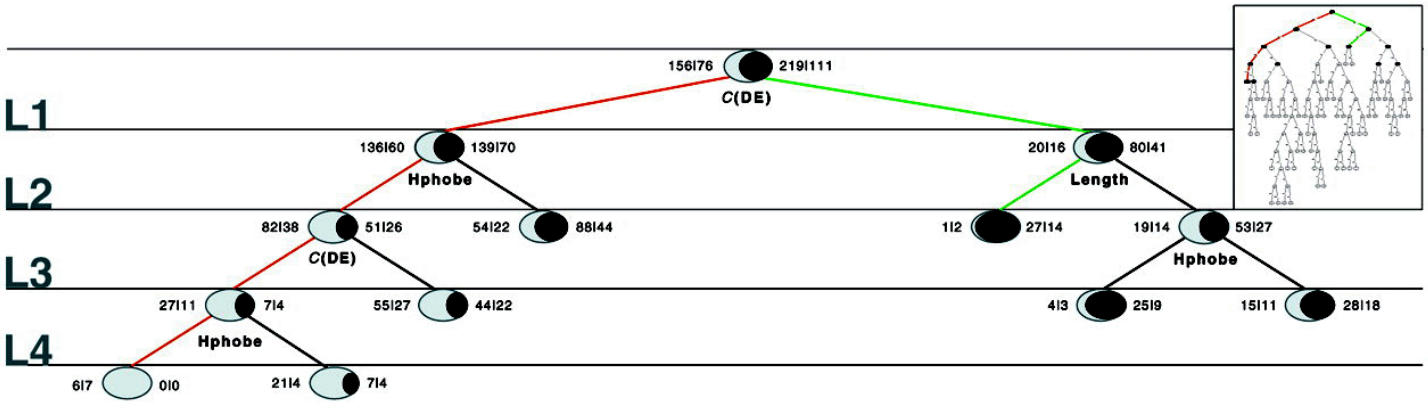
Analysis of the Suitability of 500 M. thermo. proteins to find optimal sequences purification

Overfitting, Cross Validation, and Pruning





A



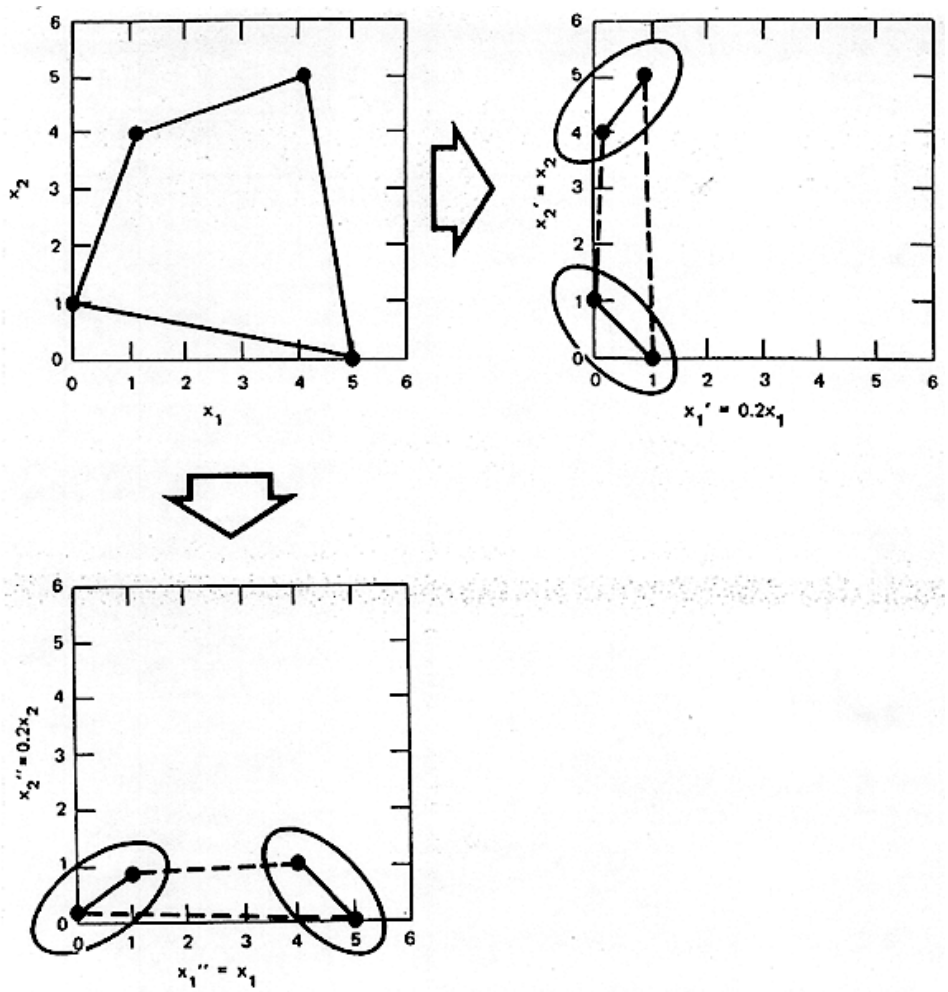
Decision Trees

- can handle data that is not linearly separable.
- A decision tree is an upside down tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a classification or *decision.* One classifies instances by sorting them down the tree from the root to some leaf nodes. To classify an instance the tree calls first for a test at the root node, testing the feature indicated on this node and choosing the next node connected to the root branch where the outcome agrees with the value of the feature of that instance. Thereafter a second test on another feature is made on the next node. This process is then repeated until a leaf of the tree is reached.
- Growing the tree, based on a training set, requires strategies for (a) splitting the nodes and (b) pruning the tree. Maximizing the decrease in average impurity is a common criterion for splitting. In a problem with noisy data (where distribution of observations from the classes overlap) growing the tree will usually over-fit the training set. The strategy in most of the cost-complexity pruning algorithms is to choose the smallest tree whose error rate performance is close to the minimal error rate of the over-fit larger tree. More specifically, growing the trees is based on splitting the node that maximizes the reduction in deviance (or any other impurity-measure of the distribution at a node) over all allowed binary splits of all terminal nodes. Splits are *not* chosen based on misclassification rate .A binary split for a continuous feature variable *v* is of the form *v<threshold* versus *v>threshold* and for a "descriptive" factor it divides the factor's levels into two classes. Decision tree-models have been successfully applied in a broad range of domains. Their popularity arises from the following: Decision trees are easy to interpret and use when the predictors are a mix of numeric and nonnumeric (factor) variables. They are invariant to scaling or re-expression of numeric variables. Compared with linear and additive models they are effective in treating missing values and capturing non-additive behavior. They can also be used to predict nonnumeric dependent variables with more than two levels. In addition, decision-tree models are useful to devise prediction rules, screen the variables and summarize the multivariate data set in a comprehensive fashion. We also note that ANN and decision tree learning often have comparable prediction accuracy [Mitchell p. 85] and SVM algorithms are slower compared with decision tree. These facts suggest that the decision tree method should be one of our top candidates to "data-mine" proteomics datasets. C4.5 and CART are among the most popular decision tree algorithms.

Optional: not needed for Quiz

(adapted from Y Kluger)

Effect of Scaling

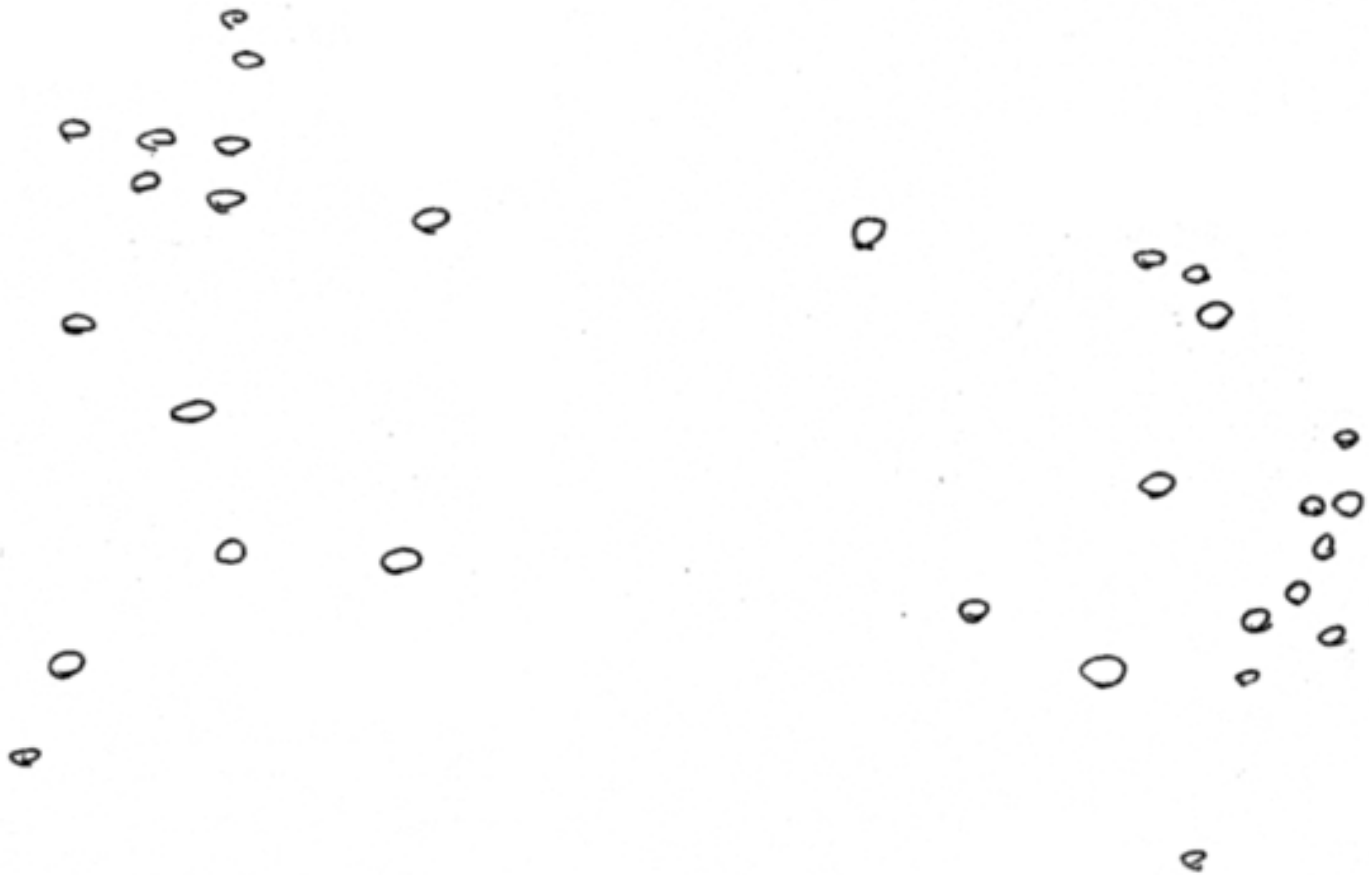


(adapted from ref?)

Large-scale Datamining

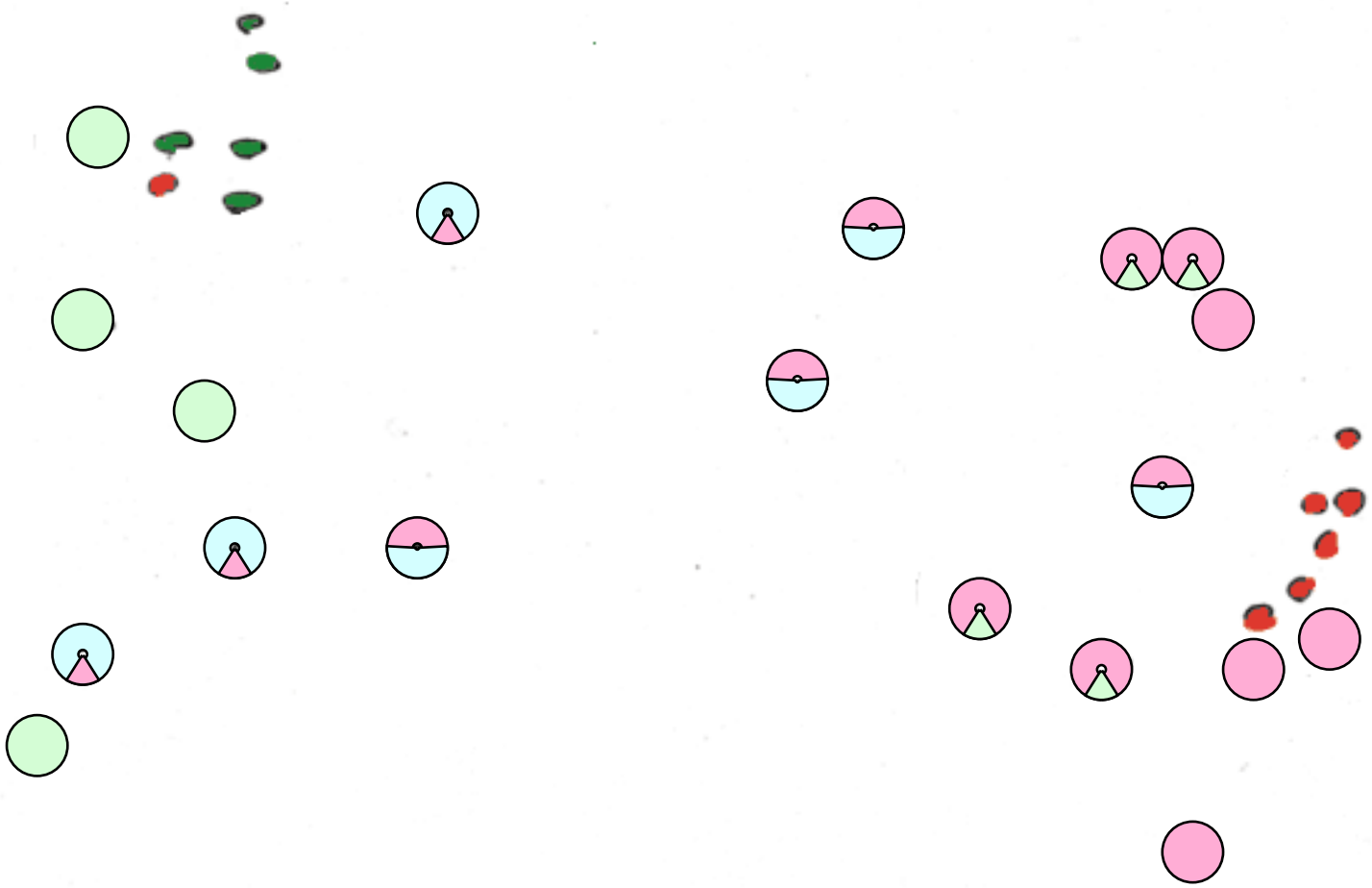
- Gene Expression
  - ◊ Representing Data in a Grid
  - ◊ Description of function prediction in abstract context
- Unsupervised Learning
  - ◊ clustering & k-means
  - ◊ Local clustering
- Supervised Learning
  - ◊ Discriminants & Decision Tree
  - ◊ Bayesian Nets
- Function Prediction EX
  - ◊ Simple Bayesian Approach for Localization Prediction

Represent predictors in abstract high dimensional space



Tagged Data

Probabilistic Predictions of Class



Yeast Tables for Localization Prediction

Basics		Predictors																									Response			Bayesian Localization						
		Sequence Features															Genomic Features																			
		seq. length	Amino Acid Composition							How many times does the sequence have these motif features?							Abs. expr. Level (mRNA copies / cell)			Cell cycle timecourse						unction	Localization		State Vector giving localization prediction					Collapsed Prediction	
Yeast Gene ID	Sequence		A	C	D			W	Y	farn site	NLS	hdel motif			nuc2	signalp	tms1	Gene-Chip expt. from RY Lab	sage tag freq.		t=0	t=1			t=15	t=16	function	5-compartment		C	N	M	T	E	Training	Extrapolation
YAL001C	M	1160	.08	.02	.06			.01	.04	0	1	0			1	0	0	0.3	0		5	3			4	5	04	TR	N	0%	100%	0%	0%	0%	N	
YAL002W	K	1176	.09	.02	.06			.01	.04	0	0	0			0	0	1	0.2	?		8	4			4	3	06	va	C	95%	3%	2%	0%	0%	C	
YAL003W	K	206	.08	.02	.06			.01	.04	0	0	0			0	0	0	19.1	19		70	73			98	126	05	tra	N	67%	33%	0%	0%	0%	C	
YAL004W	F	215	.08	.02	.06			.01	.04	0	0	0			0	0	0	?	0		18	12			4	6	01	0	N	41%	59%	0%	0%	0%	N	
YAL005C	V	641	.08	.02	.06			.01	.04	0	0	0			0	0	1	13.4	16		39	38			8	14	06	he	????	68%	32%	0%	0%	0%		C
YAL007C	K	190	.08	.02	.06			.01	.04	0	0	0			0	1	4	2.2	8		15	20			16	17	#	??	????	26%	43%	31%	0%	0%		-
YAL008W	H	198	.08	.02	.06			.01	.04	0	0	0			0	0	3	1.2	?		9	6			2	3	#	??	????	37%	60%	3%	0%	0%		-
YAL009W	E	259	.08	.02	.06			.01	.04	0	2	0			0	0	3	0.6	?		6	2			3	5	03	m	????	2%	98%	0%	0%	0%		N
YAL010C	M	493	.08	.02	.06			.02	.04	0	0	0			0	0	1	0.3	?		11	6			6	6	#	in	????	6%	90%	4%	0%	0%		N
YAL011W	K	616	.08	.02	.06			.01	.04	0	8	0			1	0	0	0.4	?		6	5			5	6	30	pr	????	28%	62%	10%	0%	0%		N
YAL012W	G	393	.08	.02	.06			.01	.04	0	0	0			0	0	1	8.9	4		29	26			23	29	01	cy	C	92%	5%	4%	0%	0%	C	
YAL013W	F	362	.08	.02	.06			.01	.04	0	0	0			0	0	0	0.6	?		7	9			6	10	01	re	N	0%	98%	0%	0%	1%	N	
YAL014C	G	202	.08	.02	.06			.01	.04	0	0	0			0	0	0	1.1	?		12	13			9	12	#	??	N	1%	96%	4%	0%	0%	N	
YAL015C	M	399	.08	.02	.06			.01	.04	0	1	0			0	0	0	0.7	0		19	18			12	13	11	D	N	4%	96%	0%	0%	0%	N	
YAL016W	K	635	.08	.02	.06			.01	.04	0	0	0			0	0	1	3.3	5		15	20			16	16	03	se	????	74%	26%	0%	0%	0%		C
YAL017W	V	1356	.08	.02	.06			.01	.04	0	0	0			0	0	0	0.4	?		14	3			4	7	#	??	????	0%	1%	99%	0%	0%		M
YAL018C	K	325	.08	.02	.06			.01	.04	0	0	0			0	0	4	?	?		4	2			2	1	#	??	????	0%	100%	0%	0%	0%		N

Large-scale Datamining

- Gene Expression
  - ◊ Representing Data in a Grid
  - ◊ Description of function prediction in abstract context
- Unsupervised Learning
  - ◊ clustering & k-means
  - ◊ Local clustering
- Supervised Learning
  - ◊ Discriminants & Decision Tree
  - ◊ Bayesian Nets
- Function Prediction EX
  - ◊ Simple Bayesian Approach for Localization Prediction

Spectral Methods Outline & Papers

- Simple background on PCA (emphasizing lingo)
- More abstract run through on SVD
- Application to
  - ◊ O Alter et al. (2000). "Singular value decomposition for genome-wide expression data processing and modeling." PNAS vol. 97: 10101-10106
  - ◊ Y Kluger et al. (2003). "Spectral biclustering of microarray data: coclustering genes and conditions." Genome Res 13: 703-16.

PCA

PCA section will be a "mash up" up a number of PPTs on the web

- pca-1 - black ---> www.astro.princeton.edu/~gk/A542/PCA.ppt
- by Professor Gillian R. Knapp gk@astro.princeton.edu

- pca-2 - yellow ---> myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt
- by Hal Whitehead.
- This is the class main url http://myweb.dal.ca/~hwhitehe/BIOL4062/handout4062.htm

- pca.ppt - what is cov. matrix ----> hebb.mit.edu/courses/9.641/lectures/pca.ppt
- by Sebastian Seung. Here is the main page of the course
- http://hebb.mit.edu/courses/9.641/index.html

- from BIIS_05lecture7.ppt ----> www.cs.rit.edu/~rsg/BIIS_05lecture7.ppt
- by R.S.Gaborski Professor

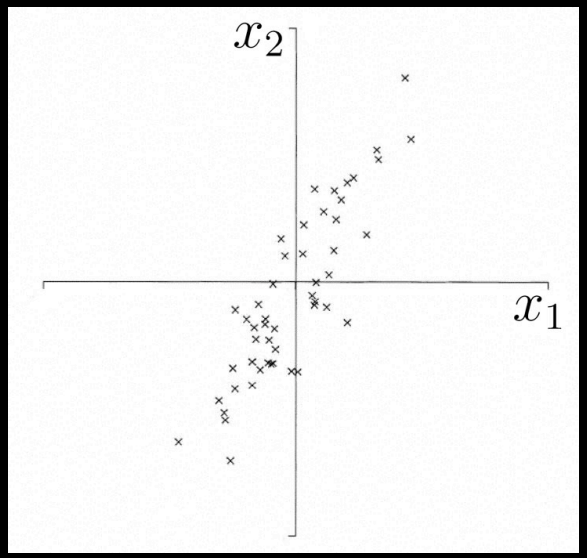
abstract

Principal component analysis (PCA) is a technique that is useful for
compression and classification of data. The purpose is to reduce the
dimensionality of a data set (sample) by finding a new set of variable
smaller than the original set of variables, that nonetheless retains mo
of the sample's information.

By information we mean the variation present in the sample,
given by the correlations between the original variables. The new
variables, called principal components (PCs), are uncorrelated, and a
ordered by the fraction of the total information each retains.

Adapted from http://www.astro.princeton.edu/~gk/A542/PCA.p

Geometric picture of principal components (PCs)

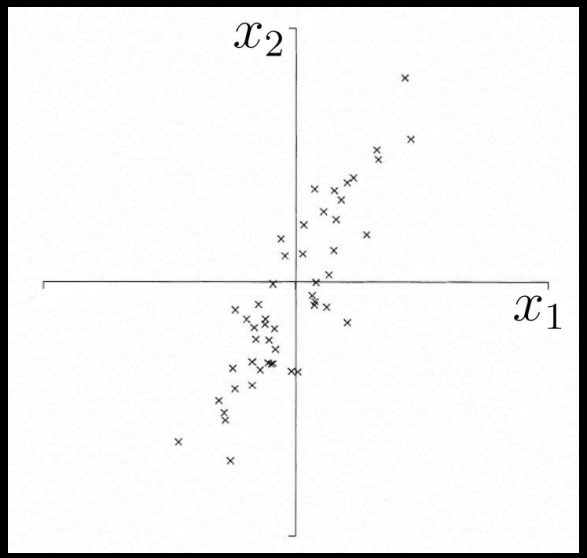


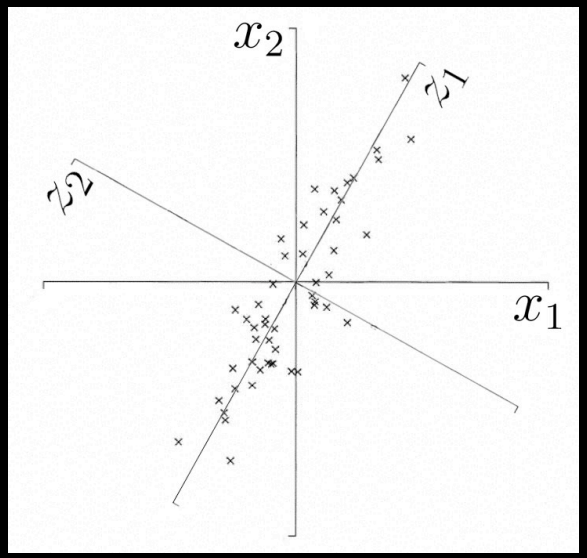
A sample of *n* observations in the 2-D space $\mathbf{x} = (x_1, x_2)$

Goal: to account for the variation in a sample
in as few variables as possible, to some accuracy

Adapted from http://www.astro.princeton.edu/~gk/A542/PCA.ppt

Geometric picture of principal components (PCs)





- the 1st PC z_1 is a minimum distance fit to a line in $\mathbf{x}$ space
- the 2nd PC z_2 is a minimum distance fit to a line in the plane perpendicular to the 1st PC

PCs are a series of linear least squares fits to a sample, each orthogonal to all the previous.

Adapted from http://www.astro.princeton.edu/~gk/A542/PCA.ppt

PCA: *General methodology*

From k original variables: $x_1, x_2, \ldots, x_k$:

Produce k new variables: $y_1, y_2, \ldots, y_k$:

$$y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k$$
$$y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k$$
$$\ldots$$
$$y_k = a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kk}x_k$$

such that:

y_k's are uncorrelated (orthogonal)

$y = Ax$

y_1 explains as much as possible of original variance in data set

y_2 explains as much as possible of remaining variance

etc.

Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

PCA: *General methodology*

From k original variables: $x_1, x_2, \ldots, x_k$:

Produce k new variables: $y_1, y_2, \ldots, y_k$:

$$y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k$$

$$y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k$$

$$\ldots$$

$$y_k = a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kk}x_k$$

y_k's are **Principal Components**

such that:

y_k's are uncorrelated (orthogonal)

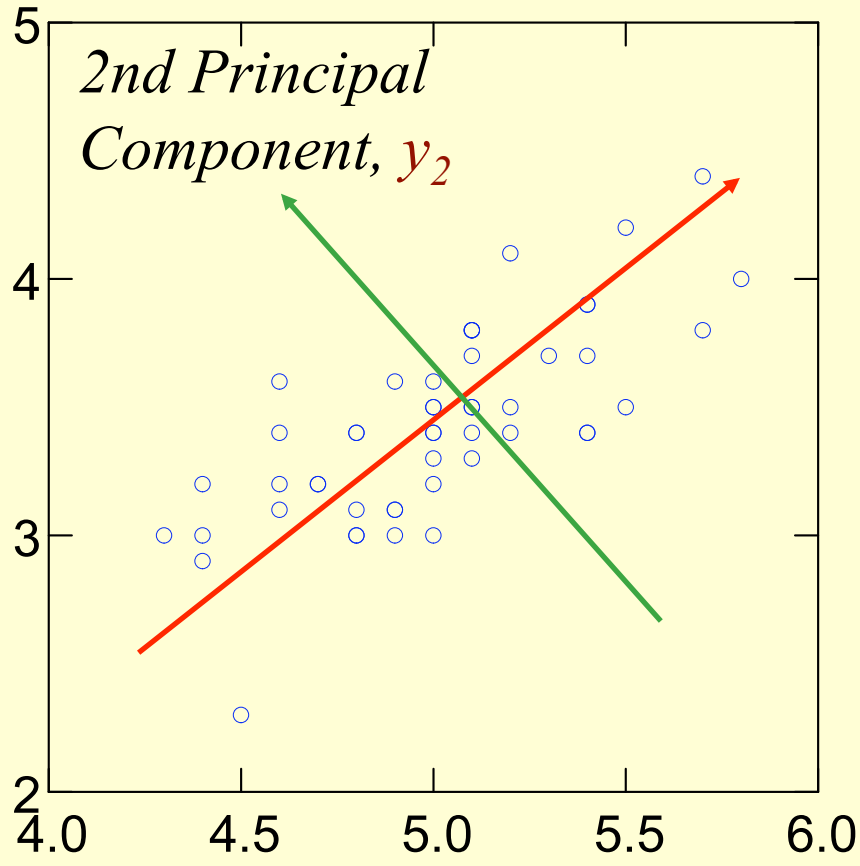
y_1 explains as much as possible of original variance in data set

y_2 explains as much as possible of remaining variance

etc.

Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

Principal Components Analysis



Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

Principal Components Analysis

- Rotates multivariate dataset into a new configuration which is easier to interpret

- Purposes
  - simplify data
  - look at relationships between variables
  - look at patterns of units

Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

Principal Components Analysis

- Uses:
  - Correlation matrix, or
  - Covariance matrix when variables in same units (morphometrics, etc.)

Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

Principal Components Analysis

$\{a_{11}, a_{12}, \ldots, a_{1k}\}$ is 1st **Eigenvector** of correlation/covariance matrix, and **coefficients** of first principal component

$\{a_{21}, a_{22}, \ldots, a_{2k}\}$ is 2nd **Eigenvector** of correlation/covariance matrix, and **coefficients** of 2nd principal component

…

$\{a_{k1}, a_{k2}, \ldots, a_{kk}\}$ is *k*th **Eigenvector** of correlation/covariance matrix, and **coefficients** of *k*th principal component

Digression #1: Where do you get covar matrix?

$\{a_{11}, a_{12}, \ldots, a_{1k}\}$ is 1st **Eigenvector** of correlation/covariance matrix, and **coefficients** of first principal component

$\{a_{21}, a_{22}, \ldots, a_{2k}\}$ is 2nd **Eigenvector** of correlation/covariance matrix, and **coefficients** of 2nd principal component

…

$\{a_{k1}, a_{k2}, \ldots, a_{kk}\}$ is *k*th **Eigenvector** of correlation/covariance matrix, and **coefficients** of *k*th principal component

Variance

- A random variable fluctuating about its mean value.

$$\delta x = x - \langle x \rangle$$

$$\left\langle (\delta x)^2 \right\rangle = \left\langle x^2 \right\rangle - \langle x \rangle^2$$

- Average of the square of the fluctuations.

Adapted from hebb.mit.edu/courses/9.641/lectures/pca.ppt

Covariance

- Pair of random variables, each fluctuating about its mean value.

$$\delta x_1 = x_1 - \langle x_1 \rangle$$

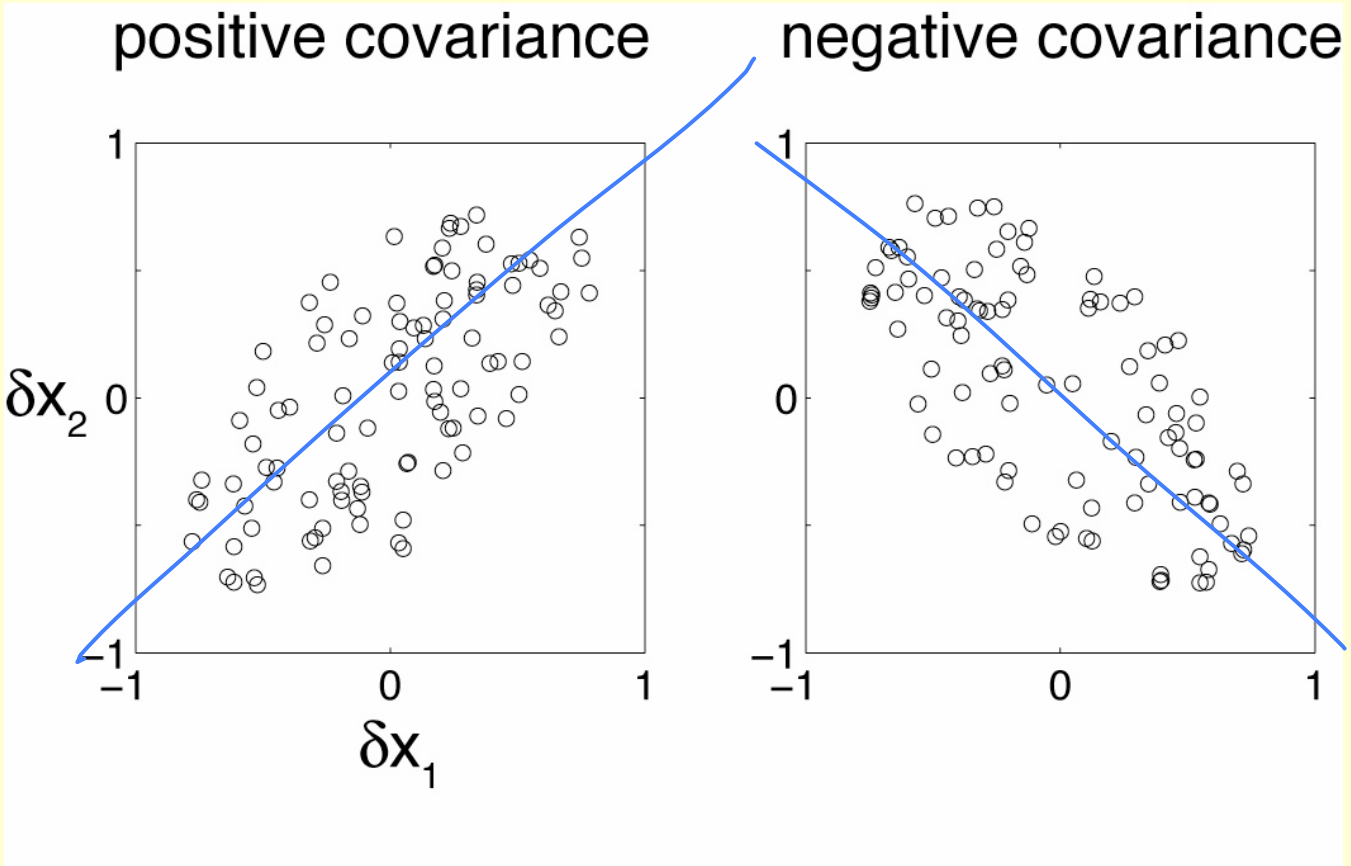
$$\delta x_2 = x_2 - \langle x_2 \rangle$$

$$\langle \delta x_1 \delta x_2 \rangle = \langle x_1 x_2 \rangle - \langle x_1 \rangle \langle x_2 \rangle$$

- Average of product of fluctuations.

Adapted from hebb.mit.edu/courses/9.641/lectures/pca.ppt

Covariance examples



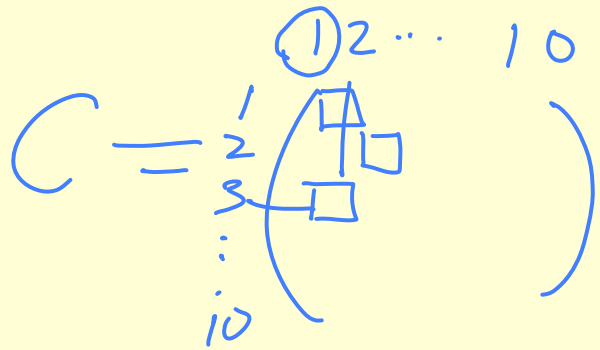
Adapted from hebb.mit.edu/courses/9.641/lectures/pca.ppt

Covariance matrix

- N random variables
- NxN symmetric matrix

$$C_{ij} = \langle x_i x_j \rangle - \langle x_i \rangle \langle x_j \rangle$$

- Diagonal elements are variances



$A = \begin{pmatrix} & \\ & \end{pmatrix}$

AA^T

A^TA

Adapted from hebb.mit.edu/courses/9.641/lectures/pca.ppt

Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

Principal Components Analysis

$\{a_{11}, a_{12}, \ldots, a_{1k}\}$ is 1st **Eigenvector** of correlation/covariance matrix, and **coefficients** of first principal component

$\{a_{21}, a_{22}, \ldots, a_{2k}\}$ is 2nd **Eigenvector** of correlation/covariance matrix, and **coefficients** of 2nd principal component

…

$\{a_{k1}, a_{k2}, \ldots, a_{kk}\}$ is *k*th **Eigenvector** of correlation/covariance matrix, and **coefficients** of *k*th principal component

Digression #2: Brief Review of Eigenvectors

$\{a_{11}, a_{12}, \ldots, a_{1k}\}$ is 1st **Eigenvector** of correlation/covariance matrix, and **coefficients** of first principal component

$\{a_{21}, a_{22}, \ldots, a_{2k}\}$ is 2nd **Eigenvector** of correlation/covariance matrix, and **coefficients** of 2nd principal component

…

$\{a_{k1}, a_{k2}, \ldots, a_{kk}\}$ is ***k*th Eigenvector** of correlation/covariance matrix, and **coefficients** of *k*th principal component

eigenvalue problem

- The eigenvalue problem is any problem having the following form:

$$\mathbf{A} \cdot \mathbf{v} = \lambda \cdot \mathbf{v}$$

A: n x n matrix

v: n x 1 non-zero vector

λ: scalar

Any value of λ for which this equation has a solution is called the eigenvalue of A and vector v which corresponds to this value is called the eigenvector of A.

[from BIIS_05lecture7.ppt]

Adapted from http://www.cs.rit.edu/~rsg/BIIS_05lecture7.ppt

eigenvalue problem

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$$\mathbf{A} \cdot \mathbf{v} = \lambda \cdot \mathbf{v}$$

Therefore, (3,2) is an eigenvector of the square matrix **A** and 4 is an eigenvalue of **A**

Given matrix A, how can we calculate the eigenvector and eigenvalues for A?

[from BIIS_05lecture7.ppt]

Adapted from http://www.cs.rit.edu/~rsg/BIIS_05lecture7.ppt

Principal Components Analysis

So, **principal components** are given by:

$$y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k$$
$$y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k$$
$$\ldots$$
$$y_k = a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kk}x_k$$

x_j's are standardized if *correlation matrix* is used (mean 0.0, SD 1.0)

Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

Principal Components Analysis

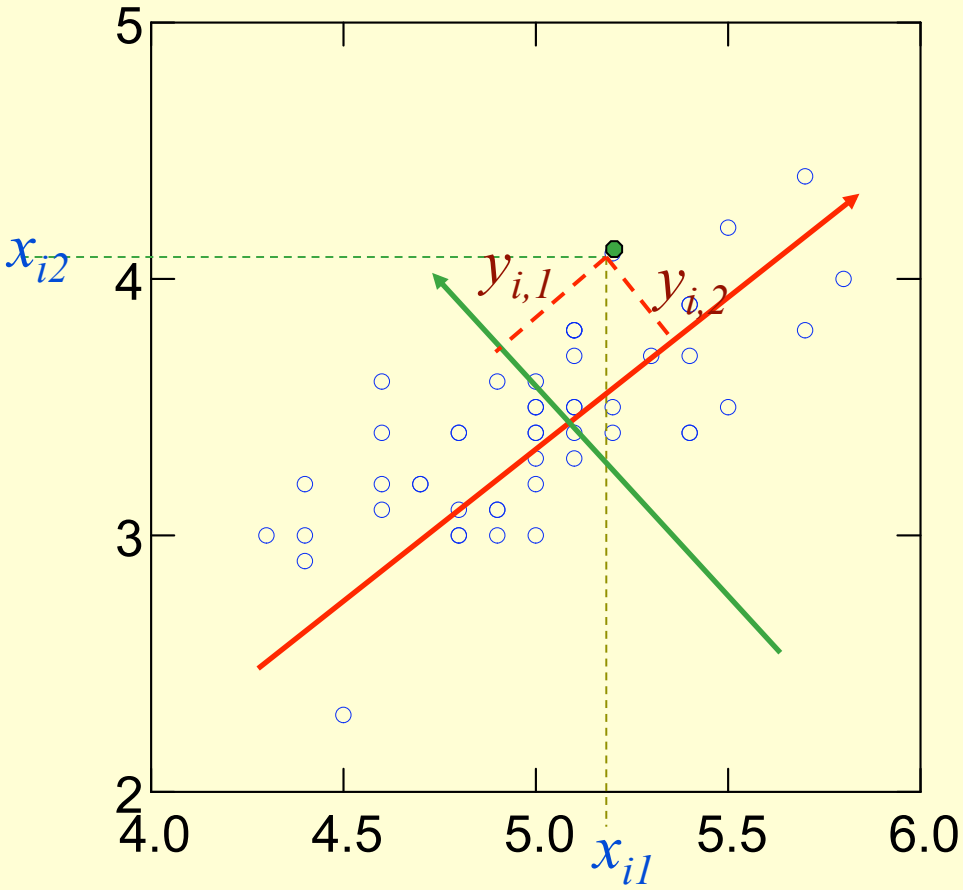
Score of *i*th unit on *j*th principal component

$$y_{i,j} = a_{j1}x_{i1} + a_{j2}x_{i2} + \ldots + a_{jk}x_{ik}$$

Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

PCA Scores



Principal Components Analysis

Amount of **variance accounted for** by:

1st principal component, λ_1, 1st **eigenvalue**

2nd principal component, λ_2, 2nd **eigenvalue**

...

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4 \geq \ldots$$

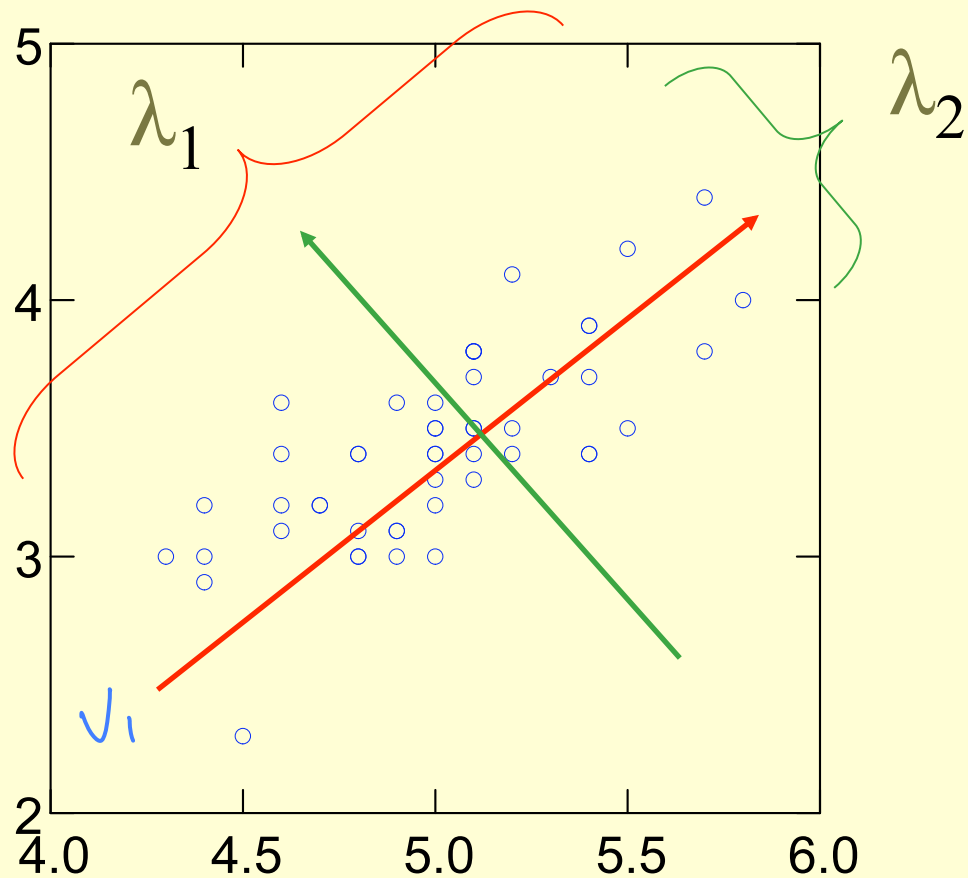
Average $\lambda_j = 1$ (correlation matrix)

Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

Principal Components Analysis: Eigenvalues

$A \downarrow C \downarrow DIAG$

$\vec{v}_i \quad \lambda_i$



Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

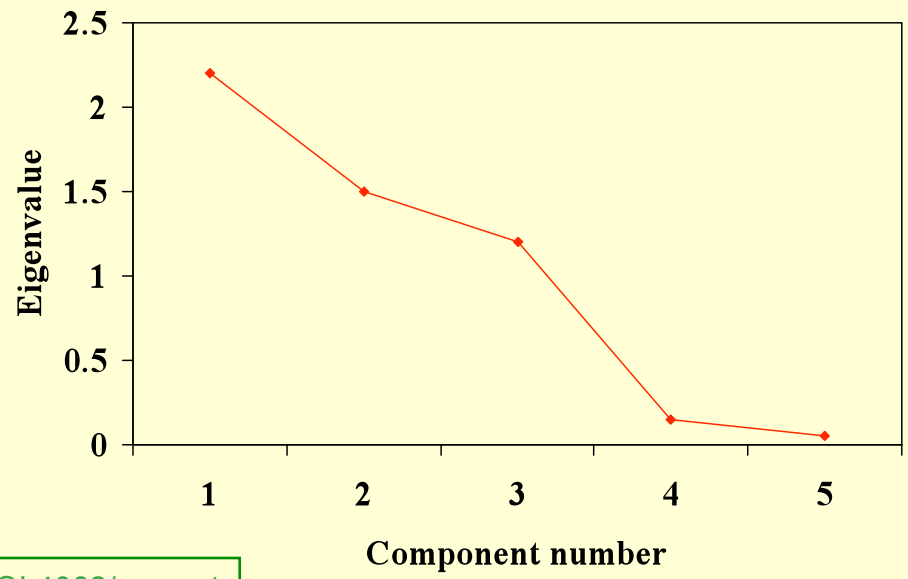
PCA: Terminology

- *j*th **principal component** is *j*th eigenvector of correlation/covariance matrix
- **coefficients**, a_{jk}, are elements of eigenvectors and relate original variables (standardized if using correlation matrix) to components
- **scores** are values of units on components (produced using coefficients)
- **amount of variance accounted for** by component is given by eigenvalue, λ_j
- **proportion of variance accounted for** by component is given by $\lambda_j / \Sigma \lambda_j$
- **loading** of *k*th original variable on *j*th

Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

How many components to use?

- If $\lambda_j < 1$ then component explains less variance than original variable (correlation matrix)
- Use 2 components (or 3) for visual ease
- Scree diagram:



Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

Principal Components Analysis on:

- *Covariance Matrix:*
  - Variables must be in same units
  - Emphasizes variables with most variance
  - Mean eigenvalue ≠1.0
  - Useful in morphometrics, a few other cases
- *Correlation Matrix:*
  - Variables are standardized (mean 0.0, SD 1.0)
  - Variables can be in different units
  - All variables have same impact on analysis
  - Mean eigenvalue = 1.0

Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

PCA: Potential Problems

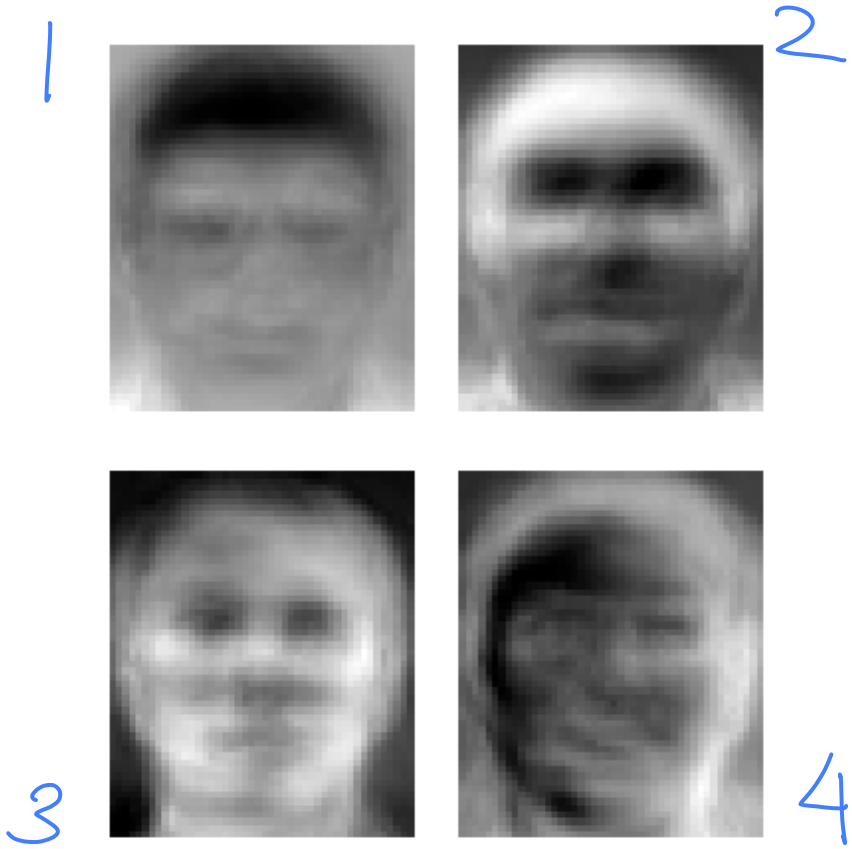
- Lack of Independence
  - **NO PROBLEM**
- Lack of Normality
  - Normality desirable but not essential
- Lack of Precision
  - Precision desirable but not essential
- Many Zeroes in Data Matrix SPARSE
  - **Problem** (use Correspondence Analysis)

Adapted from http://myweb.dal.ca/~hwhitehe/BIOL4062/pca.ppt

PCA applications -Eigenfaces

Adapted from http://www.cs.rit.edu/~rsg/BIIS_05lecture7.ppt

- the principal eigenface looks like a bland androgynous average human face



http://en.wikipedia.org/wiki/Image:Eigenfaces.png

Eigenfaces – Face Recognition

- When properly weighted, eigenfaces can be summed together to create an approximate gray-scale rendering of a human face.
- Remarkably few eigenvector terms are needed to give a fair likeness of most people's faces
- Hence eigenfaces provide a means of applying data compression to faces for identification purposes.

Adapted from http://www.cs.rit.edu/~rsg/BIIS_05lecture7.ppt

SVD

Puts together slides prepared by Brandon Xia with images from Alter et al. and Kluger et al.

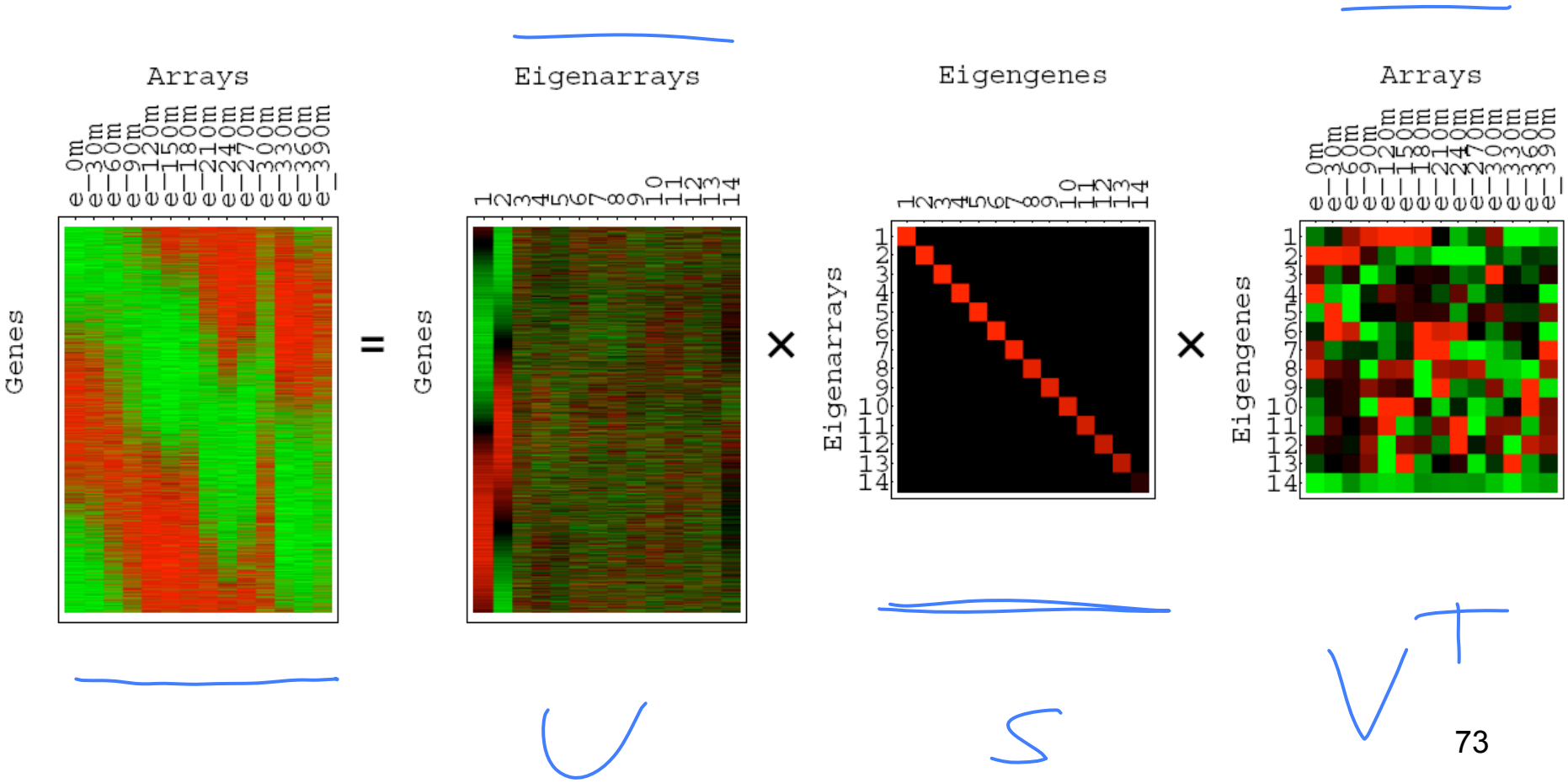
SVD

DATA

- $\boxed{A} = USV^T$
- A (m by n) is **any** rectangular matrix
  (m rows and n columns)
- U (m by n) is an "orthogonal" matrix
- S (n by n) is a diagonal matrix
- V (n by n) is another orthogonal matrix

- Such decomposition always exists
- All matrices are real; m ≥ n

SVD for microarray data (Alter et al, PNAS 2000)



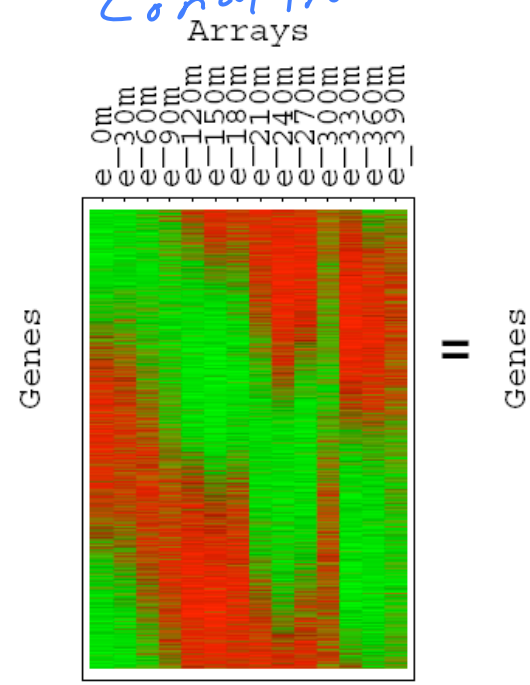
A^T

$$A = USV^T$$

1,000 people genes

10 conditions

- A is any rectangular matrix (m ≥ n)
- #1,000 10D Row space: vector subspace generated by the row vectors of A
- Column space: vector subspace generated by the column vectors of A
  - 10,000D #10 The dimension of the row & column space is the rank of the matrix A: r (≤ n)
- A is a linear transformation that maps vector x in row space into vector Ax in column space

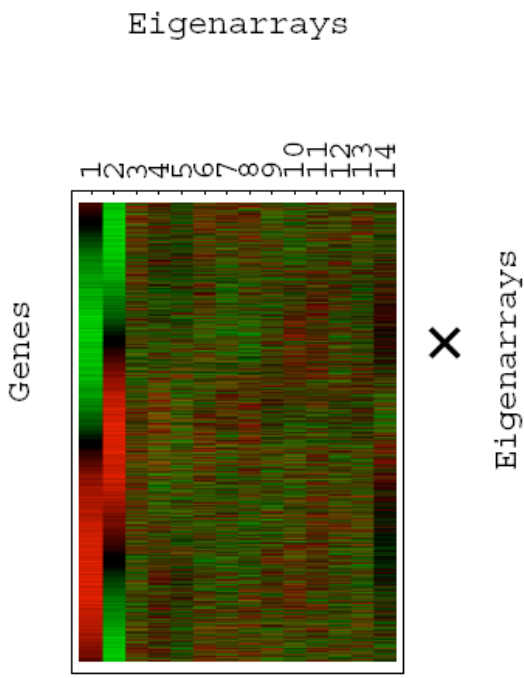


$A = USV^T$

- U is an "orthogonal" matrix (m ≥ n)
- Column vectors of U form an orthonormal basis for the column space of A: $U^T U = I$

$$U = \begin{pmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ | & | & & | \end{pmatrix}$$

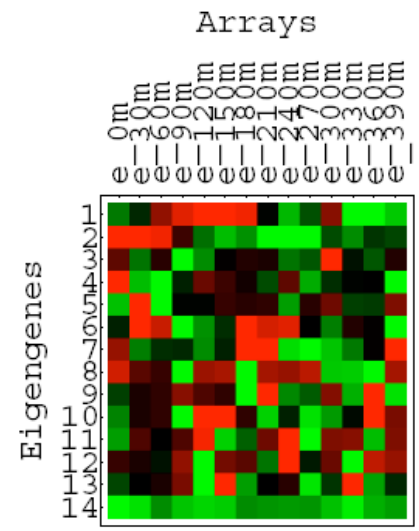
- $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ in U are eigenvectors of AA^T
  - $AA^T = USV^T VSU^T = US^2 U^T$
  - "Left singular vectors"



$A = USV^T$

- V is an orthogonal matrix (n by n)
- Column vectors of V form an orthonormal basis for the row space of A: $V^TV = VV^T = I$

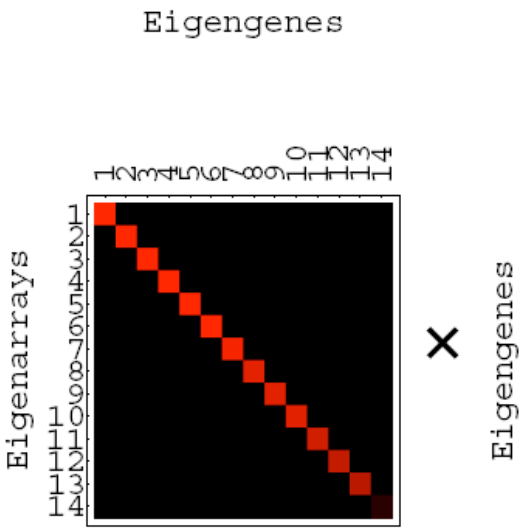
$$V = \begin{pmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \\ | & | & & | \end{pmatrix}$$



- $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ in V are eigenvectors of A^TA
  - $A^TA = VSU^T USV^T = VS^2 V^T$
  - "Right singular vectors"

$$A = USV^T$$

- S is a diagonal matrix (n by n) of non-negative singular values
- Typically sorted from largest to smallest
- Singular values are the non-negative square root of corresponding eigenvalues of A^TA and AA^T

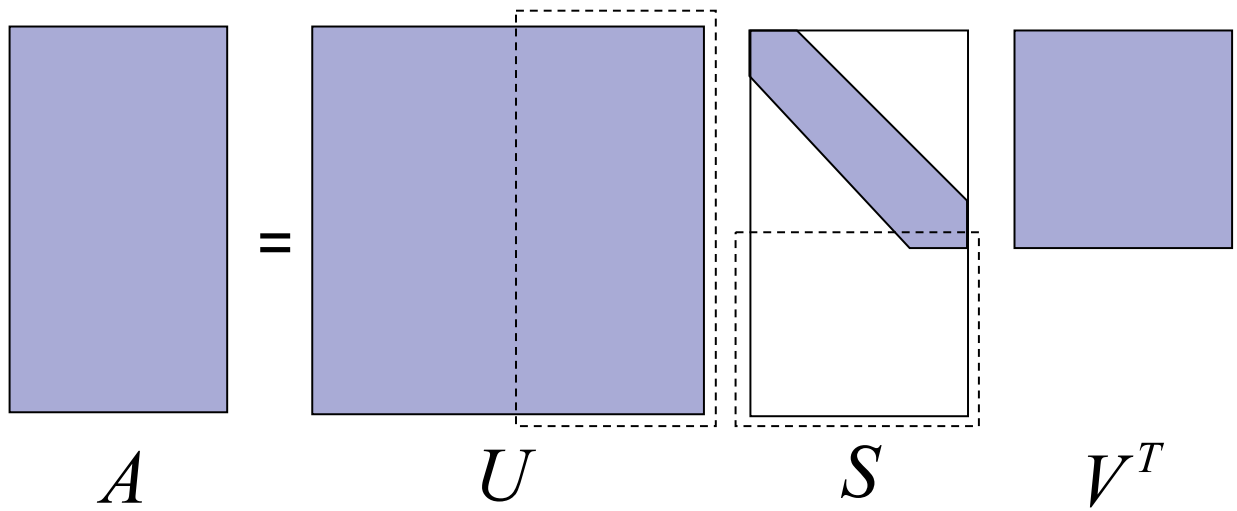


$$AV = US$$

- Means each $A\boldsymbol{v}_i = s_i\boldsymbol{u}_i$
- Remember A is a linear map from row space to column space
- Here, A maps an orthonormal basis $\{\boldsymbol{v}_i\}$ in row space into an orthonormal basis $\{\boldsymbol{u}_i\}$ in column space
- Each component of u_i is the projection of a row onto the vector v_i

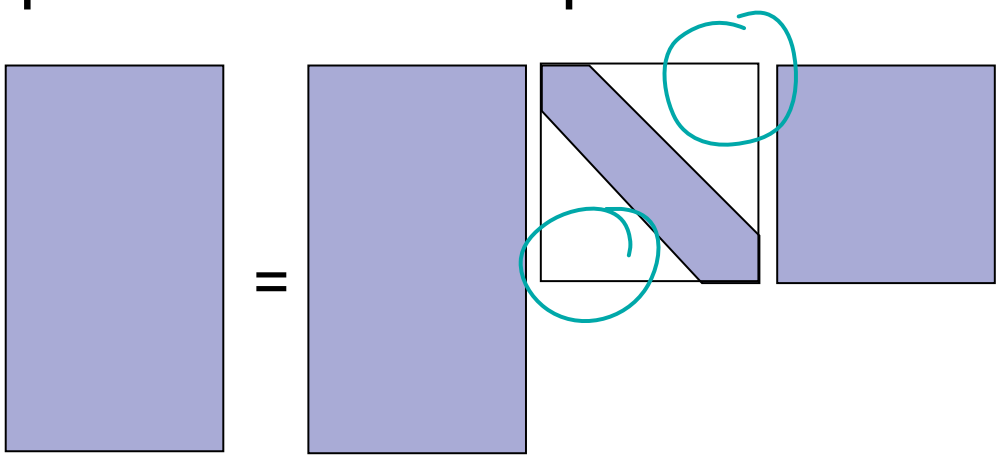
Full SVD

- We can complete U to a full orthogonal matrix and pad S by zeros accordingly



Reduced SVD

- For rectangular matrices, we have two forms of SVD. The reduced SVD looks like this:
  - The columns of U are orthonormal
  - Cheaper form for computation and storage



A U S V^T

SVD of A (m by n): recap

- $A = USV^T =$ (big-"orthogonal")(diagonal)(sq-orthogonal)

- $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ in U are eigenvectors of AA^T
- $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ in V are eigenvectors of A^TA
- $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$ in S are nonnegative singular values of A

- $AV = US$ means each $A\boldsymbol{v}_i = s_i\boldsymbol{u}_i$
- "Every A is diagonalized by 2 orthogonal matrices"

SVD as sum of rank-1 matrices

- $A = USV^T$
- $A = s_1\boldsymbol{u}_1\boldsymbol{v}_1^T + s_2\boldsymbol{u}_2\boldsymbol{v}_2^T + \ldots + s_n\boldsymbol{u}_n\boldsymbol{v}_n^T$
- $s_1 \geq s_2 \geq \ldots \geq s_n \geq 0$
- What is the rank-r matrix $\hat{A}$ that best approximates A ?
  - Minimize $\sum_{i=1}^{m}\sum_{j=1}^{n}\left(\hat{A}_{ij} - A_{ij}\right)^2$
- $\hat{A} = s_1\boldsymbol{u}_1\boldsymbol{v}_1^T + s_2\boldsymbol{u}_2\boldsymbol{v}_2^T + \ldots + s_r\boldsymbol{u}_r\boldsymbol{v}_r^T$
- Very useful for matrix approximation

LSQ

$\vec{u}_1 \cdot \vec{v}_1$

$\vec{u}_1 \vec{v}^T =$

RANK r

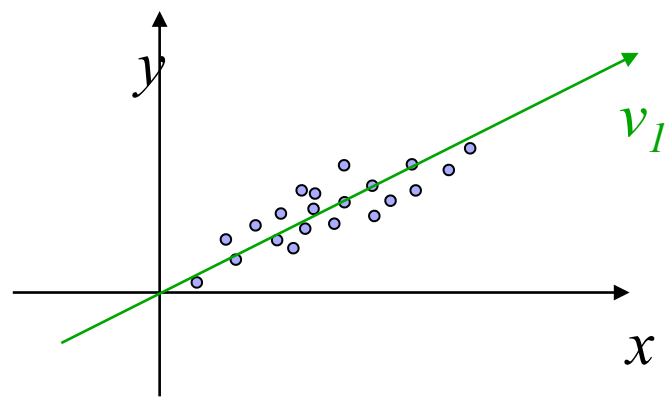
Examples of (almost) rank-1 matrices

- Steady states with fluctuations $\begin{pmatrix} 101 & 103 & 102 \\ 302 & 300 & 301 \\ 203 & 204 & 203 \\ 401 & 402 & 404 \end{pmatrix}$
- Array artifacts? $\begin{pmatrix} 101 & 303 & 202 \\ 102 & 300 & 201 \\ 103 & 304 & 203 \\ 101 & 302 & 204 \end{pmatrix}$
- Signals? $\begin{pmatrix} 1 & 2 & -1 \\ 2 & 4 & -2 \\ -1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

$A v_i = s u_i$

Geometry of SVD in row space

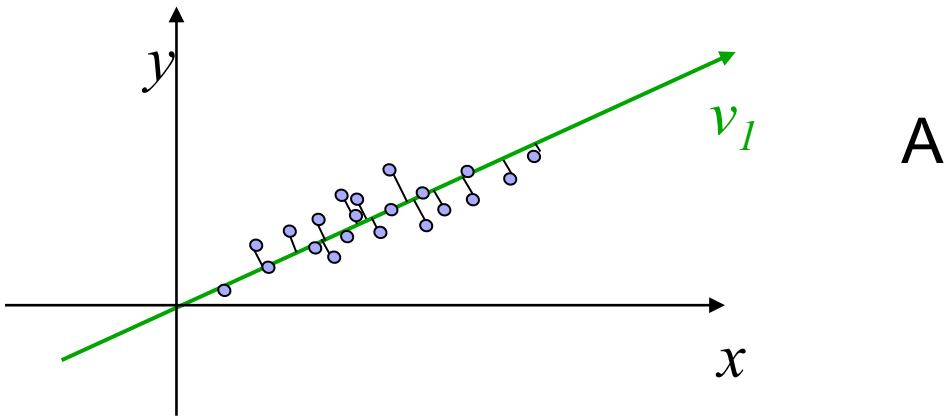
- A as a collection of m row vectors (points) in the row space of A
- $s_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T$ is the best rank-1 matrix approximation for A
- Geometrically: $\boldsymbol{v}_1$ is the direction of the best approximating rank-1 subspace that goes through origin
- $s_1 \boldsymbol{u}_1$ gives coordinates for row vectors in rank-1 subspace
- $\boldsymbol{v}_1$ Gives coordinates for row space basis vectors in rank-1 subspace



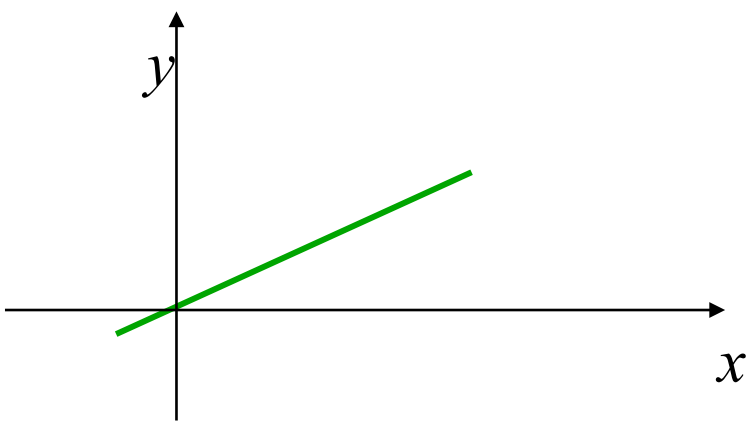
$$\boxed{A\,\mathbf{v_i} = s_i \mathbf{u_i}}$$

$$\boxed{I\,\mathbf{v_i} = \mathbf{v_i}}$$

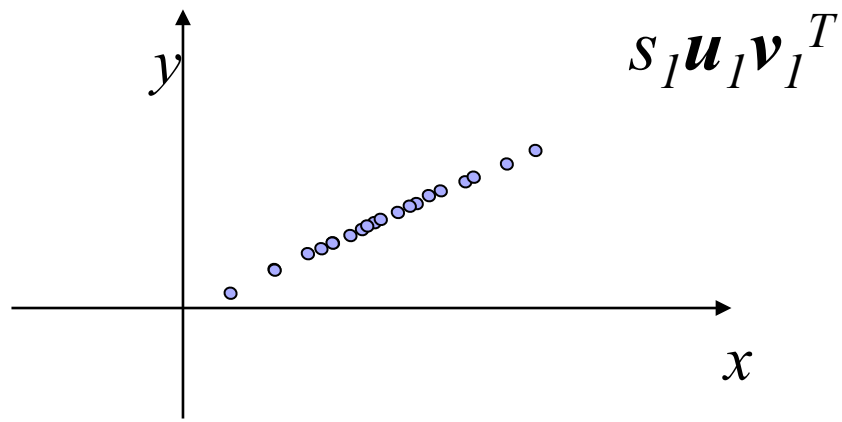
Geometry of SVD in row space



A



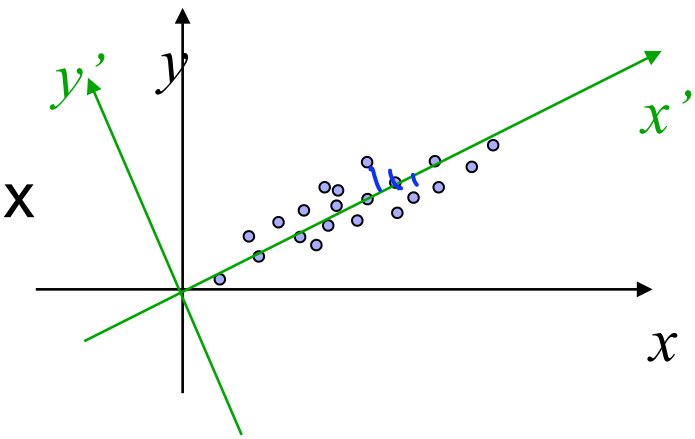
This line segment that goes through origin approximates the original data set



The projected data set approximates the original data set

Geometry of SVD in row space

- A as a collection of m row vectors (points) in the row space of A
- $s_1\boldsymbol{u}_1\boldsymbol{v}_1^T + s_2\boldsymbol{u}_2\boldsymbol{v}_2^T$ is the best rank-2 matrix approximation for A
- Geometrically: $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are the directions of the best approximating rank-2 subspace that goes through origin
- $s_1\boldsymbol{u}_1$ and $s_2\boldsymbol{u}_2$ gives coordinates for row vectors in rank-2 subspace
- $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ gives coordinates for row space basis vectors in rank-2 subspace

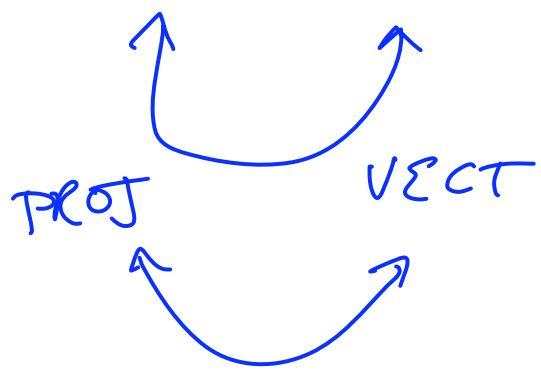


$$A\,\mathbf{v_i} = s_i\mathbf{u_i}$$

$$I\,\mathbf{v_i} = \mathbf{v_i}$$

What about geometry of SVD in column space?

- $A = USV^T$
- $A^T = VSU^T$
- The column space of A becomes the row space of A^T
- The same as before, except that U and V are switched



Geometry of SVD in row and column spaces

- Row space
  - $s_i \boldsymbol{u}_i$ gives coordinates for row vectors along unit vector $\boldsymbol{v}_i$
  - $\boldsymbol{v}_i$ gives coordinates for row space basis vectors along unit vector $\boldsymbol{v}_i$
- Column space
  - $s_i \boldsymbol{v}_i$ gives coordinates for column vectors along unit vector $\boldsymbol{u}_i$
  - $\boldsymbol{u}_i$ gives coordinates for column space basis vectors along unit vector $\boldsymbol{u}_i$
- Along the directions $\boldsymbol{v}_i$ and $\boldsymbol{u}_i$, these two spaces look pretty much the same!
  - Up to scale factors s_i
  - Switch row/column vectors and row/column space basis vectors
  - **Biplot....**

$$A\,\mathbf{v_i} = s_i \mathbf{u_i}$$

$$I\,\mathbf{v_i} = \mathbf{v_i}$$

$$A^T\,\mathbf{u_i} = s_i \mathbf{v_i}$$

$$I\,\mathbf{u_i} = \mathbf{u_i}$$

Biplot

- A biplot is a two-dimensional representation of a data matrix showing a point for each of the n observation vectors (rows of the data matrix) along with a point for each of the p variables (columns of the data matrix).
  - The prefix 'bi' refers to the two kinds of points; not to the dimensionality of the plot. The method presented here could, in fact, be generalized to a threedimensional (or higher-order) biplot. Biplots were introduced by Gabriel (1971) and have been discussed at length by Gower and Hand (1996). We applied the biplot procedure to the following toy data matrix to illustrate how a biplot can be generated and interpreted. See the figure on the next page.
- Here we have three variables (transcription factors) and ten observations (genomic bins). We can obtain a two-dimensional plot of the observations by plotting the first two principal components of the TF-TF correlation matrix R1.
  - We can then add a representation of the three variables to the plot of principal components to obtain a biplot. This shows each of the genomic bins as points and the axes as linear combination of the factors.
- The great advantage of a biplot is that its components can be interpreted very easily. First, correlations among the variables are related to the angles between the lines, or more specifically, to the cosines of these angles. An acute angle between two lines (representing two TFs) indicates a positive correlation between the two corresponding variables, while obtuse angles indicate negative correlation.
  - Angle of 0 or 180 degrees indicates perfect positive or negative correlation, respectively. A pair of orthogonal lines represents a correlation of zero. The distances between the points (representing genomic bins) correspond to the similarities between the observation profiles. Two observations that are relatively similar across all the variables will fall relatively close to each other within the two-dimensional space used for the biplot. The value or score for any observation on any variable is related to the perpendicular projection form the point to the line.
- Refs
  - Gabriel, K. R. (1971), "The Biplot Graphical Display of Matrices with Application to Principal Component Analysis," Biometrika, 58, 453–467.
  - Gower, J. C., and Hand, D. J. (1996), Biplots, London: Chapman & Hall.

Biplot Ex

A

TF

Genomic bin	a	b	c
1	21	16	28
2	14	18	25
3	14	17	22
4	14	19	33
5	17	23	28
6	20	14	34
7	22	21	30
8	15	18	22
9	18	13	36
10	24	10	32

Data matrix

Variable (column) standardization

TF

Genomic bin	a	b	c
1	0.84	-0.23	-0.20
2	-1.06	0.29	-0.82
3	-1.06	0.03	-1.43
4	-1.06	0.55	0.82
5	-0.24	1.59	-0.20
6	0.57	-0.75	1.02
7	1.11	1.07	0.20
8	-0.78	0.29	-1.43
9	0.03	-1.01	1.43
10	1.65	-1.80	0.61

Standardized data matrix

Correlating factors

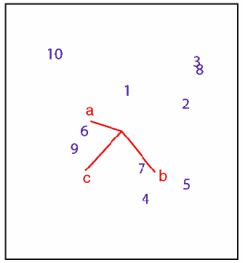
TF

TF	a	b	c
a	1.00	-0.44	0.48
b	-0.44	1.00	-0.40
c	0.48	-0.40	1.00

Correlation matrix R_1

PCA

3D scatterplot



Transpose

Genomic bin

Transcription factor	1	2	3	4	5	6	7	8	9	10
a	21	14	14	14	17	20	22	15	18	24
b	16	18	17	19	23	14	21	18	13	10
c	28	25	22	33	28	34	30	22	36	32

Data matrix (transposed)

A

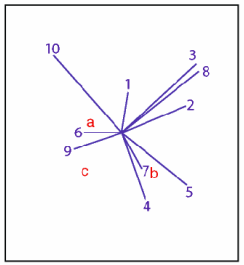
Variable (column) standardization

Genomic bin

TF	1	2	3	4	5	6	7	8	9	10
a	-0.11	-0.90	-0.91	-0.81	-1.03	-0.26	-0.47	-0.95	-0.36	0.18
b	-0.94	-0.18	-0.16	-0.30	0.06	-0.84	-0.68	-0.09	-0.77	-1.08
c	1.05	1.08	1.07	1.12	0.97	1.10	1.15	1.04	1.13	0.90

Standardized data matrix (transposed)

10D scatterplot *



PCA *

Correlating bins

Genomic bin

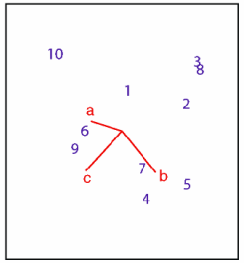
Genomic bin	1	2	3	4	5	6	7	8	9	10
1	1.00	0.70	0.69	0.77	0.54	0.99	0.95	0.65	0.98	0.97
2	0.70	1.00	1.00	0.99	0.98	0.79	0.89	1.00	0.84	0.50
3	0.69	1.00	1.00	0.99	0.98	0.78	0.89	1.00	0.83	0.49
4	0.77	0.99	0.99	1.00	0.95	0.85	0.94	0.98	0.89	0.59
5	0.54	0.98	0.98	0.95	1.00	0.64	0.78	0.99	0.71	0.31
6	0.99	0.79	0.78	0.85	0.64	1.00	0.98	0.74	1.00	0.93
7	0.95	0.89	0.89	0.94	0.78	0.98	1.00	0.86	0.99	0.84
8	0.65	1.00	1.00	0.98	0.99	0.74	0.86	1.00	0.80	0.43
9	0.98	0.84	0.83	0.89	0.71	1.00	0.99	0.80	1.00	0.89
10	0.97	0.50	0.49	0.59	0.31	0.93	0.84	0.43	0.89	1.00

Correlation matrix R_2

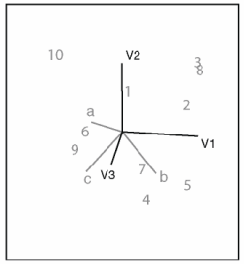
Biplot Ex #2

TF	a	b	c
a	1.00	-0.44	0.48
b	-0.44	1.00	-0.40
c	0.48	-0.40	1.00

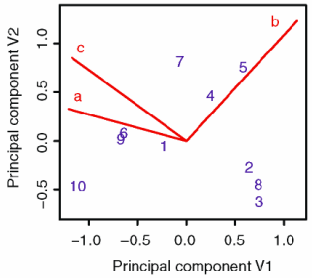
Correlation matrix R_1



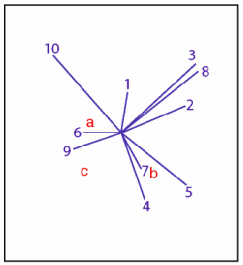
PCA *



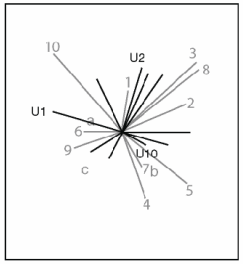
Projection *



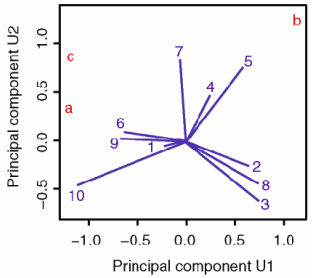
The same rank-2 approximation of the original data matrix



PCA *



Projection *



Genomic bin

2	3	4	5	6	7	8	9	10
.70	0.69	0.77	0.54	0.99	0.95	0.65	0.98	0.97
.00	1.00	0.99	0.98	0.79	0.89	1.00	0.84	0.50
.00	1.00	0.99	0.98	0.78	0.89	1.00	0.83	0.49
.99	0.99	1.00	0.95	0.85	0.94	0.98	0.89	0.59
.98	0.98	0.95	1.00	0.64	0.78	0.99	0.71	0.31
.79	0.78	0.85	0.64	1.00	0.98	0.74	1.00	0.93
.89	0.89	0.94	0.78	0.98	1.00	0.86	0.99	0.84
.00	1.00	0.98	0.99	0.74	0.86	1.00	0.80	0.43
.84	0.83	0.89	0.71	1.00	0.99	0.80	1.00	0.89
.50	0.49	0.59	0.31	0.93	0.84	0.43	0.89	1.00

Correlation matrix R_2

* 10D scatterplots are used here for illustrative purpose only.
PCA: the correlation matrix is eigen-decomposed; then the principal components are added to the original space.
Projection: the points and axes in the original space are projected onto the plane defined by the top two principal components.

Biplot Ex #3

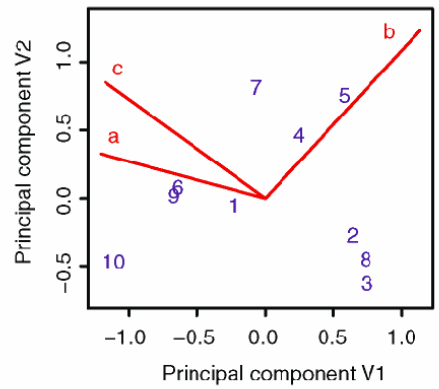
$$A\,\mathbf{v_i} = s_i\mathbf{u_i}$$

$$A^T\,\mathbf{u_i} = s_i\mathbf{v_i}$$

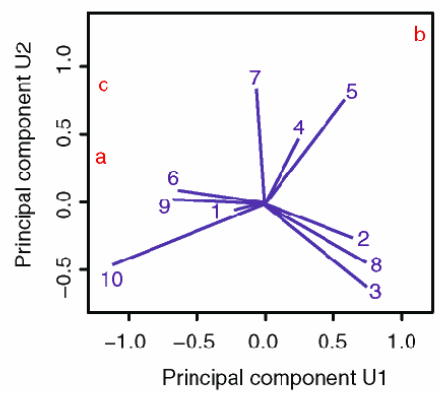
Assuming s=1,

$A\mathbf{v}_i = \mathbf{u}_i$

$A^T\mathbf{u}_i = \mathbf{v}_i$

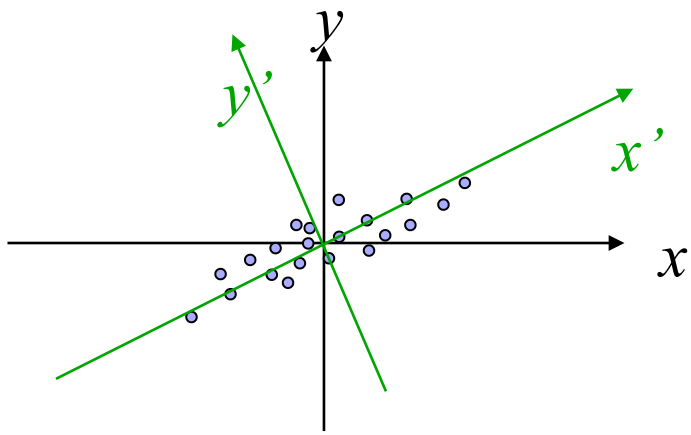


The same rank-2 approximation of the original data matrix

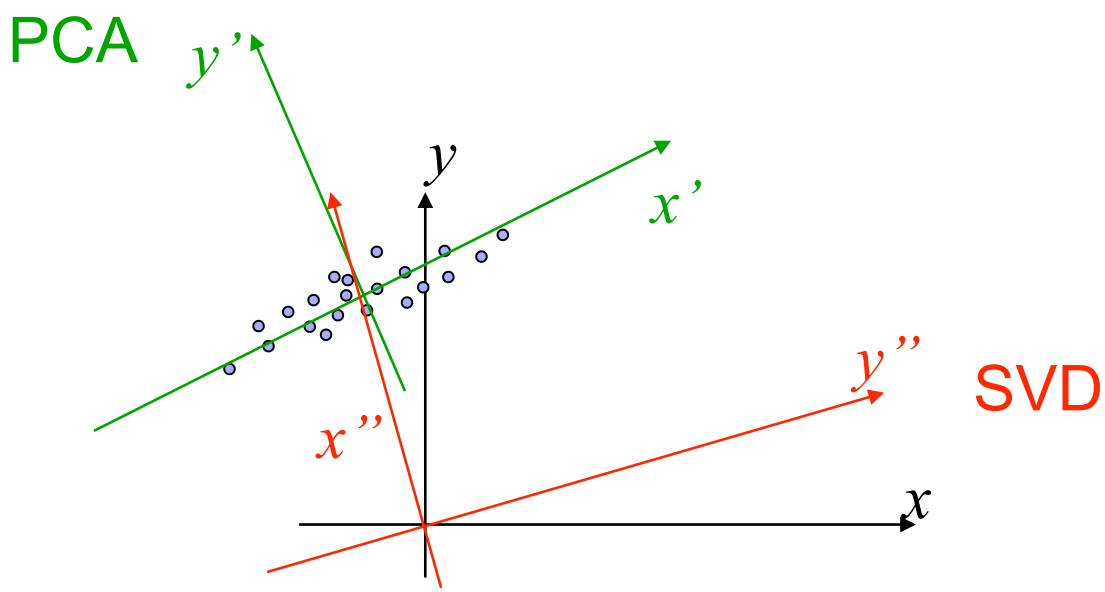


When is SVD = PCA?

- Centered data



When is SVD different from PCA?



Translation is not a linear operation, as it moves the origin !

Additional Points

Time Complexity Issues with SVD

AA^T

A^TA

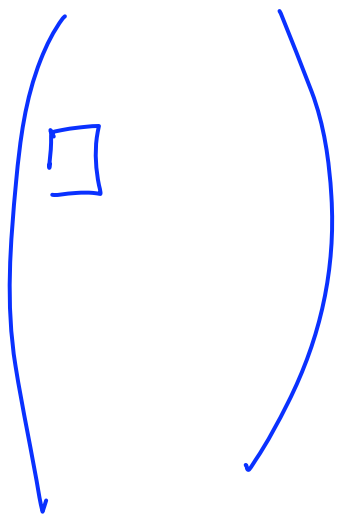
- SVD CUBIC COMPLEXITY

Application of SVD to text mining

DOCS

TERMS

A =



- SPARSE

LATENT SEMANTIC INDEXING

Conclusion

- SVD is the “absolute high point of linear algebra”
- SVD is difficult to compute; but once we have it, we have many things
- SVD finds the best approximating subspace, using linear transformation
- Simple SVD cannot handle translation, non-linear transformation, separation of labeled data, etc.
- Good for exploratory analysis; but once we know what we look for, use appropriate tools and model the structure of data explicitly!