Pre-processing the data

Scaling in case the input variables are on different scale

- Recommended to give equal weights to all variables.
  - Just think about the euclidean distance

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$

Larger values will drive the distance (think about gene expression)
…and you don’t want this

Scaling in case the input variables are on different scale

- Recommended to give equal weight to all variable.
  - Just think about linear regression

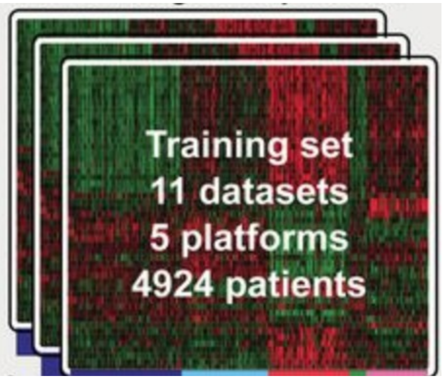
$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j .$$

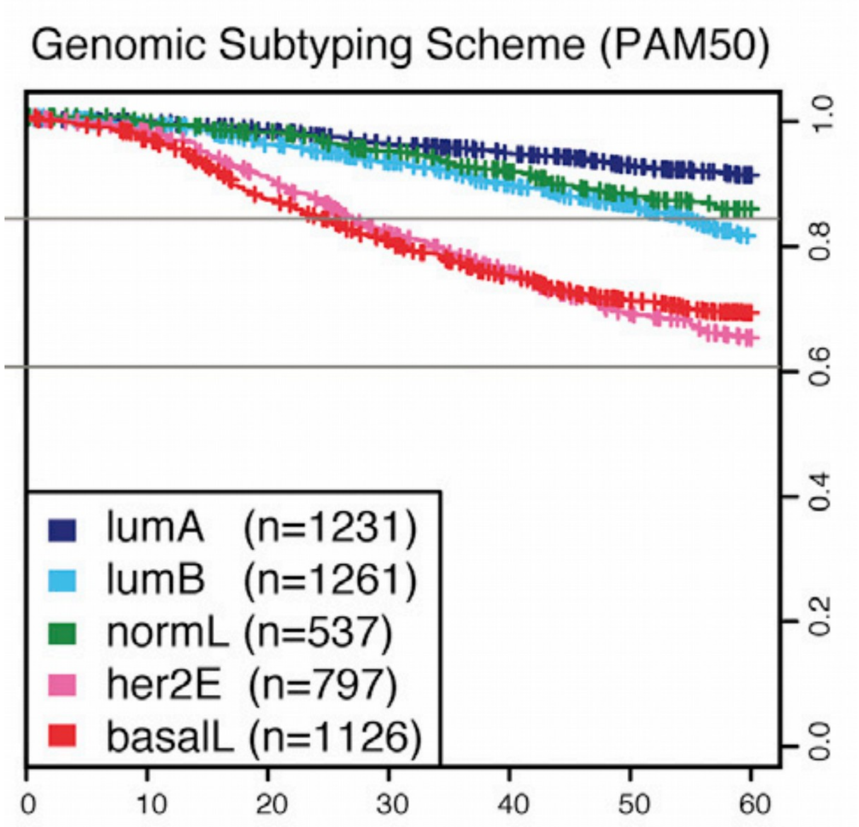
Coeffients would be different highly express versus lowly express genes

…but this can introduce come problems

- Scaling or centering assumes that the mean across different datasets would be similar ie the mean in the training versus test and to other future datasets have to be the same….
- We have shown it is not always the case and that subtle modifications to a dataset can change the results. True in breast cancer gene expression datasets at least….

Example with breast cancer subtypes





PAM50 uses a gene centering pre-processing step….

It assumes all datasets would be equa ie have roughly the same composition

Paquet et al. JNCI 2015

Not all breast cance datasets have the same composition

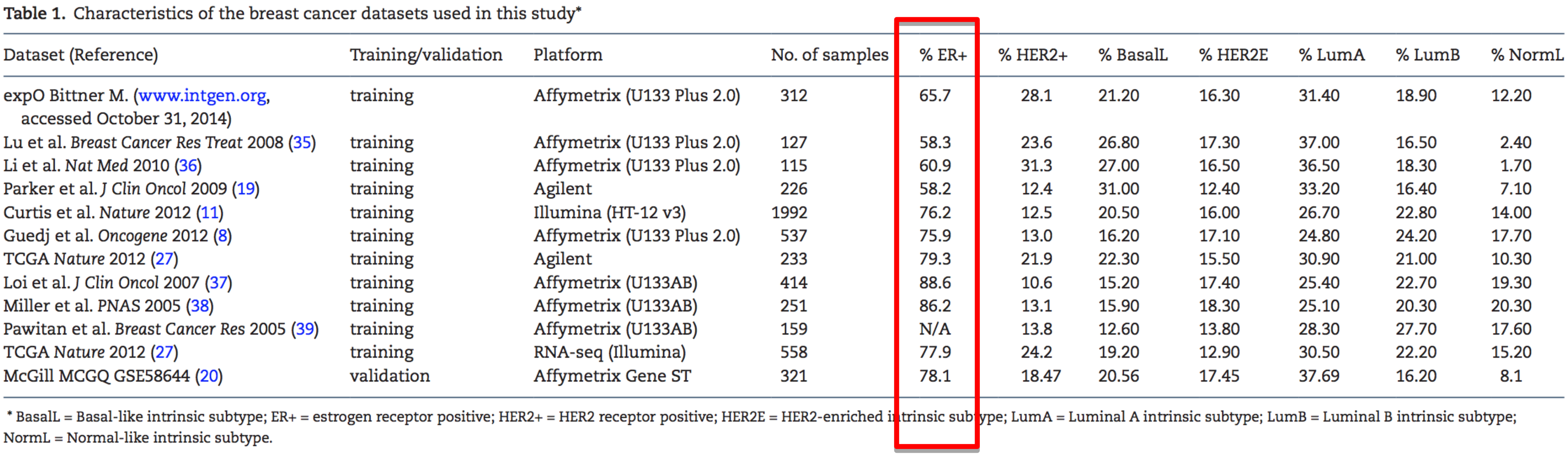


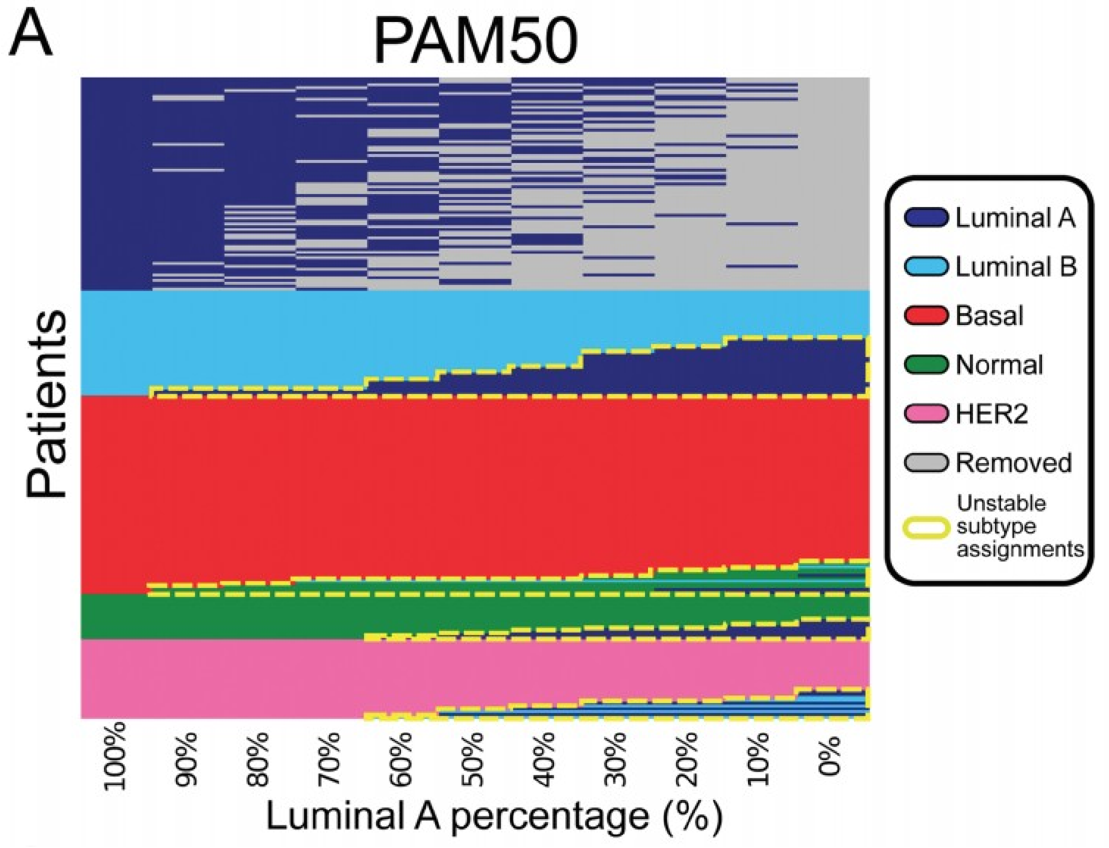
Table 1. Characteristics of the breast cancer datasets used in this study*

Dataset (Reference)	Training/validation	Platform	No. of samples	% ER+	% HER2+	% BasalL	% HER2E	% LumA	% LumB	% NormL
expO Bittner M. (www.intgen.org, accessed October 31, 2014)	training	Affymetrix (U133 Plus 2.0)	312	65.7	28.1	21.20	16.30	31.40	18.90	12.20
Lu et al. *Breast Cancer Res Treat* 2008 (35)	training	Affymetrix (U133 Plus 2.0)	127	58.3	23.6	26.80	17.30	37.00	16.50	2.40
Li et al. *Nat Med* 2010 (36)	training	Affymetrix (U133 Plus 2.0)	115	60.9	31.3	27.00	16.50	36.50	18.30	1.70
Parker et al. *J Clin Oncol* 2009 (19)	training	Agilent	226	58.2	12.4	31.00	12.40	33.20	16.40	7.10
Curtis et al. *Nature* 2012 (11)	training	Illumina (HT-12 v3)	1992	76.2	12.5	20.50	16.00	26.70	22.80	14.00
Guedj et al. *Oncogene* 2012 (8)	training	Affymetrix (U133 Plus 2.0)	537	75.9	13.0	16.20	17.10	24.80	24.20	17.70
TCGA *Nature* 2012 (27)	training	Agilent	233	79.3	21.9	22.30	15.50	30.90	21.00	10.30
Loi et al. *J Clin Oncol* 2007 (37)	training	Affymetrix (U133AB)	414	88.6	10.6	15.20	17.40	25.40	22.70	19.30
Miller et al. PNAS 2005 (38)	training	Affymetrix (U133AB)	251	86.2	13.1	15.90	18.30	25.10	20.30	20.30
Pawitan et al. *Breast Cancer Res* 2005 (39)	training	Affymetrix (U133AB)	159	N/A	13.8	12.60	13.80	28.30	27.70	17.60
TCGA *Nature* 2012 (27)	training	RNA-seq (Illumina)	558	77.9	24.2	19.20	12.90	30.50	22.20	15.20
McGill MCGQ GSE58644 (20)	validation	Affymetrix Gene ST	321	78.1	18.47	20.56	17.45	37.69	16.20	8.1

* BasalL = Basal-like intrinsic subtype; ER+ = estrogen receptor positive; HER2+ = HER2 receptor positive; HER2E = HER2-enriched intrinsic subtype; LumA = Luminal A intrinsic subtype; LumB = Luminal B intrinsic subtype; NormL = Normal-like intrinsic subtype.

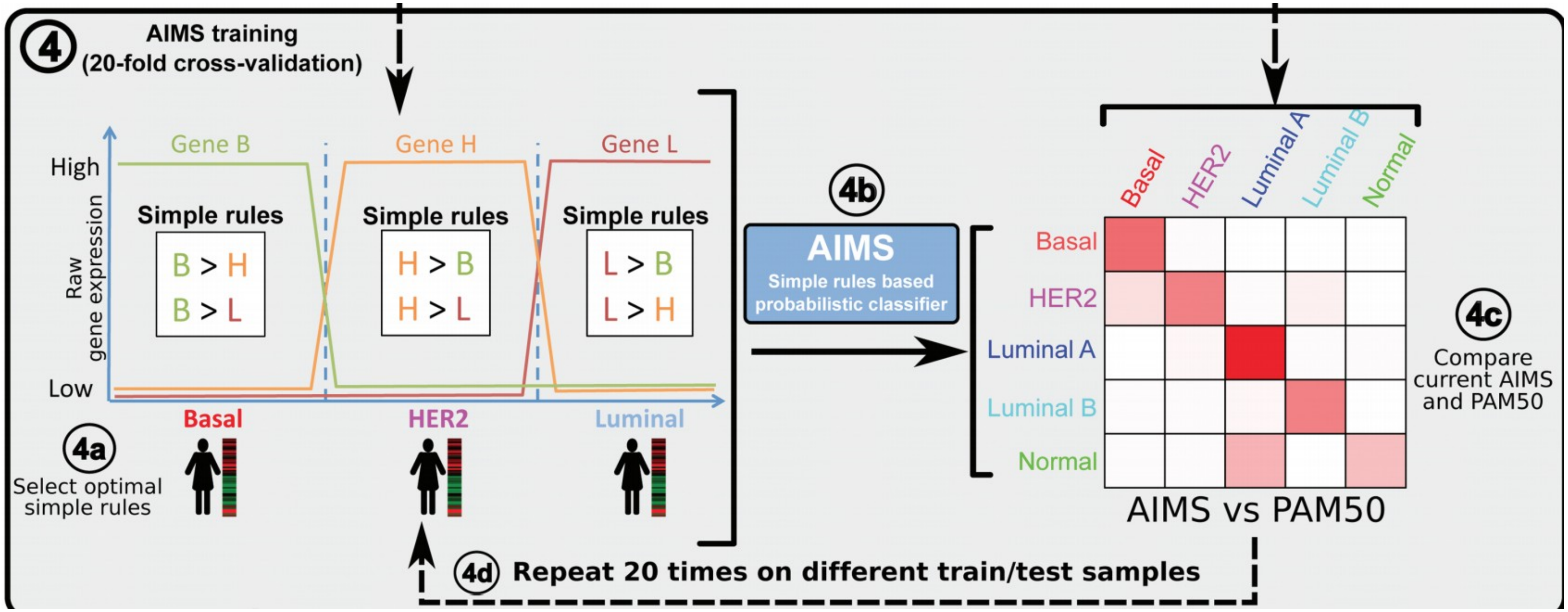
Paquet et al. JNCI 2015

What happen if we artificially change the composition of the dataset?



Paquet et al. JNCI 2015

How did we solve this?



We decided to go for simple binary feature rules estimated from “raw” data instead of requiring gene centering.

Paquet et al. JNCI 2015

Take home message

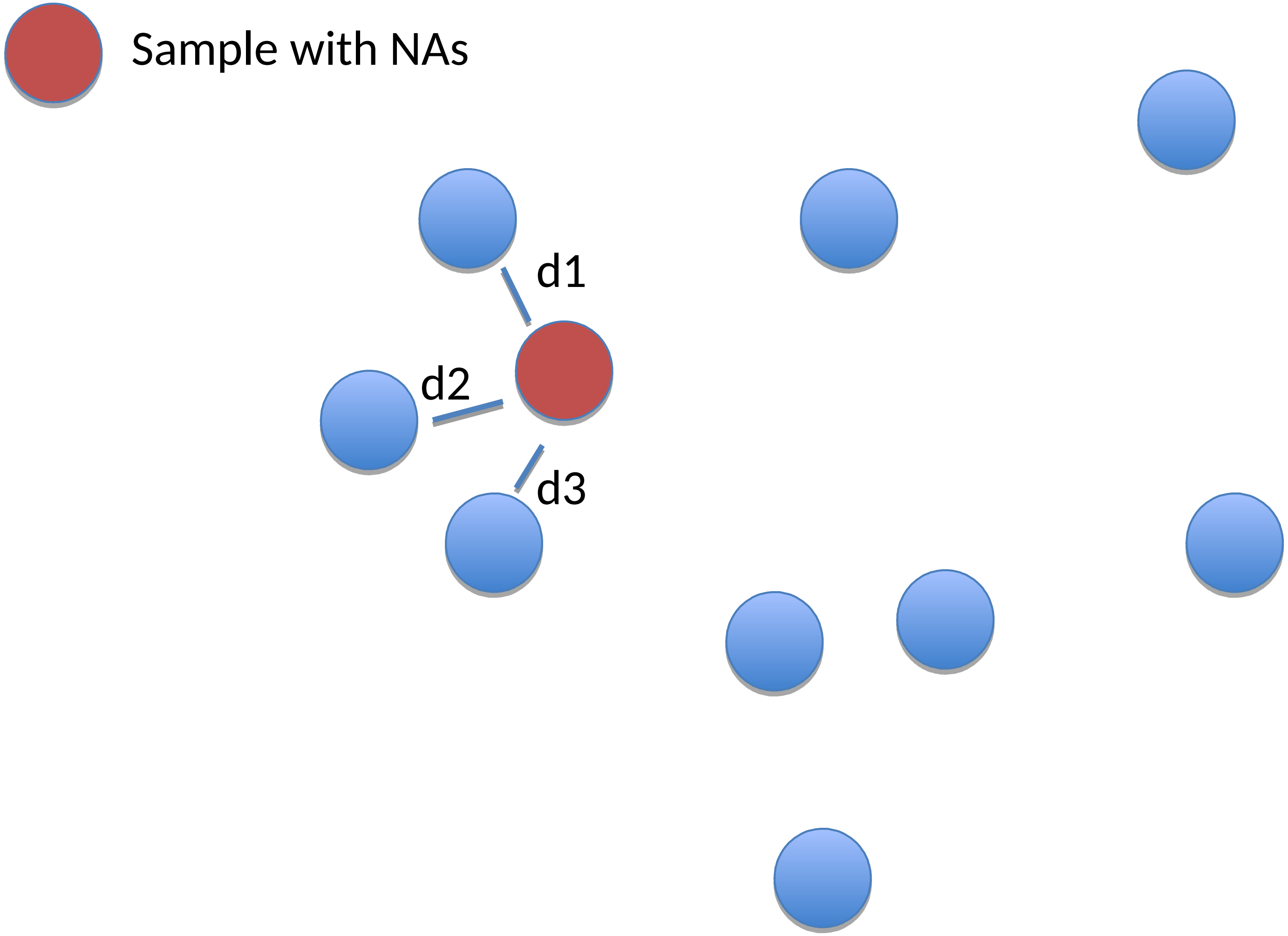
- Sometime pre-processing is important BUT
- It also introduces strong assumption on the future composition of your datasets
- You need to think about this when training your models

Imputation

What to do when you have missing data?

- Throw away the samples with NA
  - In case you don’t have a lot of samples with NA this is a good option
- Throw away the variables with NA
  - If the variable is mostly NA then it is fine, the variable was not informative anyway
- Do some imputation
  - Example. Use a knn based approach. Find the k closest samples using knn and non-NA values and impute the NA with the mean of the k-nearest neighbors.

knnImpute



K=3

1) d = dist(a,b) not using NA
2) Average the NA values from other samples

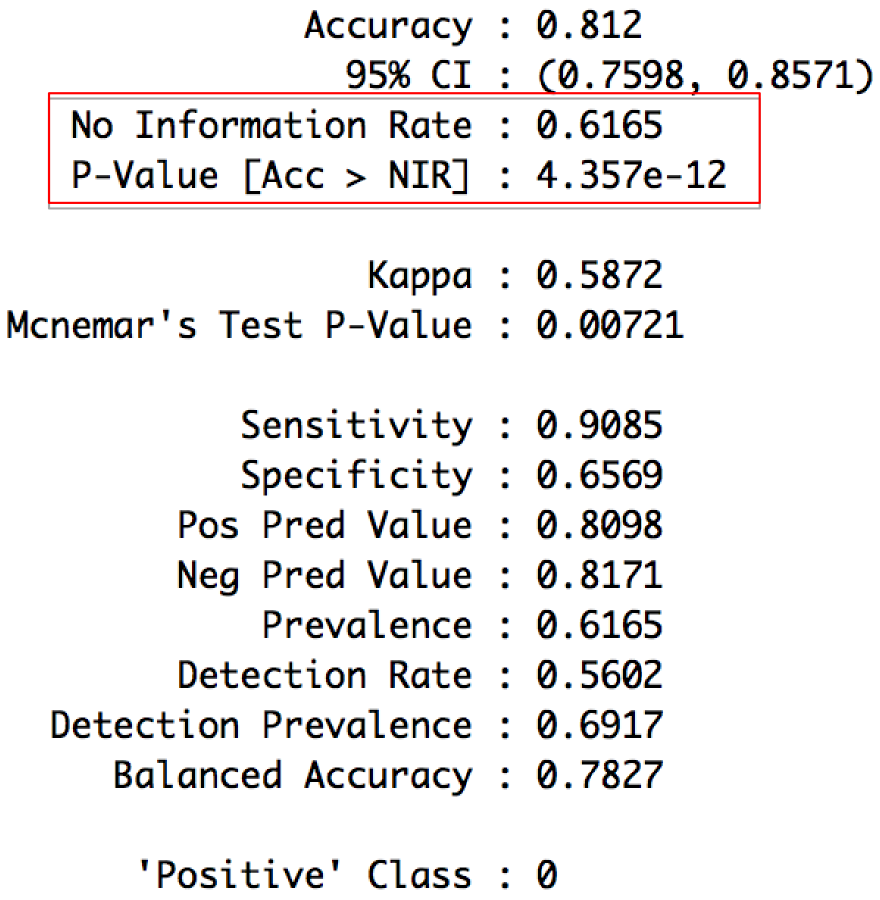
Class imbalance

With high class imbalance we could have the "fealing" of performance

- Example
  - 80% patients are of class responders
  - 20% patients are of class non-responders
    - Random prior would classify all patients as responders
    - You need to be careful when working with strong imbalance.
    - Look at several metrics sensitivity and specificity + accuracy. Maybe also Matthew's correlation coefficient (less sensitive to imbalance):

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Example (caret computeCon)



```
              Accuracy : 0.812
                95% CI : (0.7598, 0.8571)
   No Information Rate : 0.6165
   P-Value [Acc > NIR] : 4.357e-12

                 Kappa : 0.5872
Mcnemar's Test P-Value : 0.00721

           Sensitivity : 0.9085
           Specificity : 0.6569
        Pos Pred Value : 0.8098
        Neg Pred Value : 0.8171
            Prevalence : 0.6165
        Detection Rate : 0.5602
  Detection Prevalence : 0.6917
     Balanced Accuracy : 0.7827

      'Positive' Class : 0
```

Features selection
P >> N
genomics

P >> N

- Case where number of features are way higher than the number of samples
  - P >> N
- 3 strategies :
  - Select features (how many? -> Cross-validation)
    - What about correlated features?
    - Use you favorite approaches (t-test, wilcox-test, fold change, etc)
  - Dimension reduction [generalization ?]
    - PCA
  - Regularization approaches
    - Ridge (L2-norm), lasso (L1-norm), elastic net (mixing L2 and L1)

Regularization : Ridge, lasso, elastic net

Ridge(L2-norm)

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \left\{ \boxed{\sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\Big)^2} + \boxed{\lambda \sum_{j=1}^{p} \beta_j^2} \right\}.$$

Lasso (L1-norm)

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \boxed{\frac{1}{2} \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\Big)^2} + \boxed{\lambda \sum_{j=1}^{p} |\beta_j|} \right\}.$$

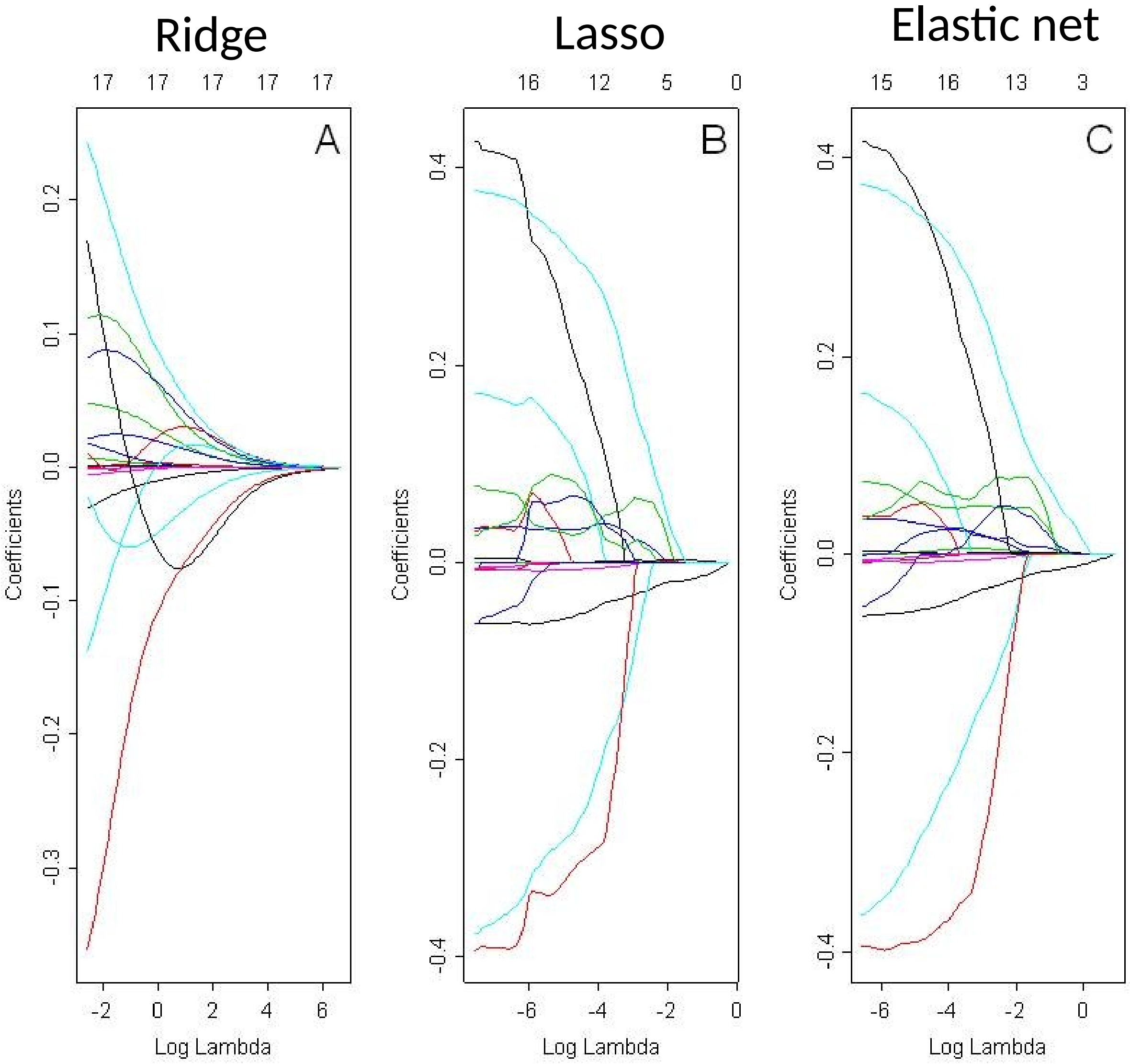
Elastic net

$$\boxed{\lambda \sum_{j=1}^{p} \left(\alpha\beta_j^2 + (1-\alpha)|\beta_j|\right),}$$

Combine both

The elements ot statistical learning

Lasso and elastic net would set coefficients to 0 “selecting features” while optimizing



Ridge

Lasso

Elastic net

Lasso and elastic can drive coefficients to zero, but this is not the case for ridge

Different regularizations, different properties (number of features)

- Ridge would not select features ie set coefficients to 0
- Lasso would do feature selection [p >> n]

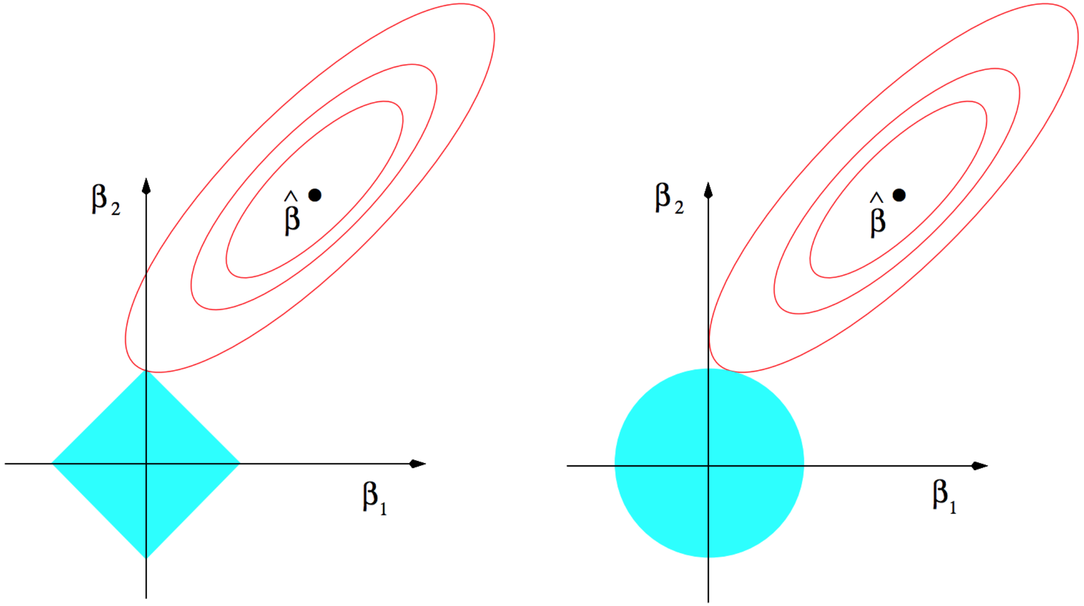


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$, respectively, while the red ellipses are the contours of the least squares error function.*

$$\boxed{\lambda \sum_{j=1}^{p} |\beta_j|}$$

$$\boxed{\lambda \sum_{j=1}^{p} \beta_j^2}$$

Different regularizations, different properties (correlated features)

- Ridge regression would tend to give equal weigths to correlated features [robustness].
- Lasso would tend to select one of the correlated features randomly.

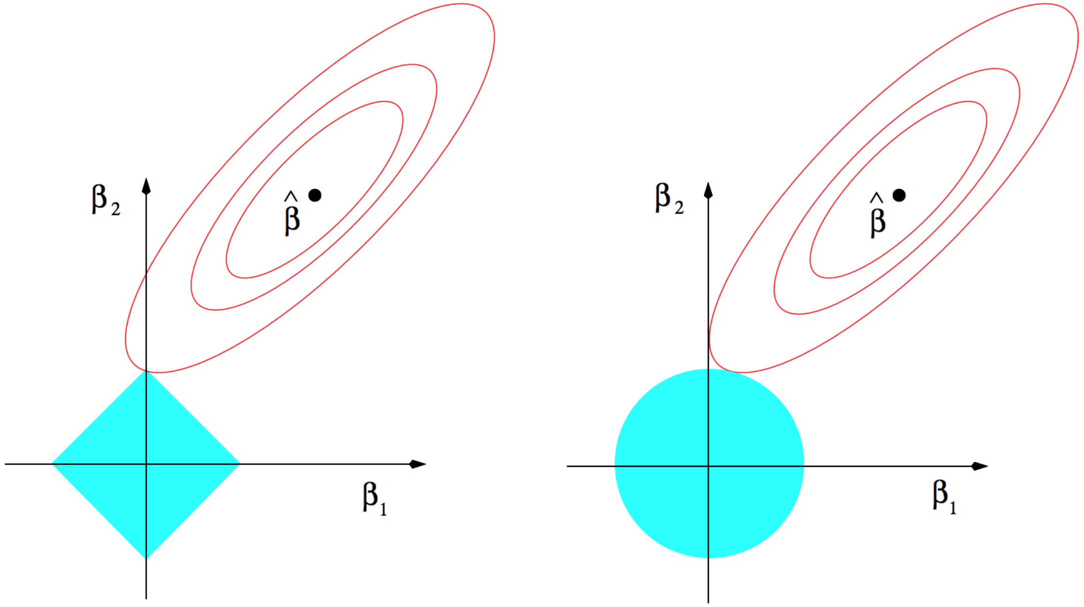


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

$$\boxed{\lambda \sum_{j=1}^{p} |\beta_j|}$$

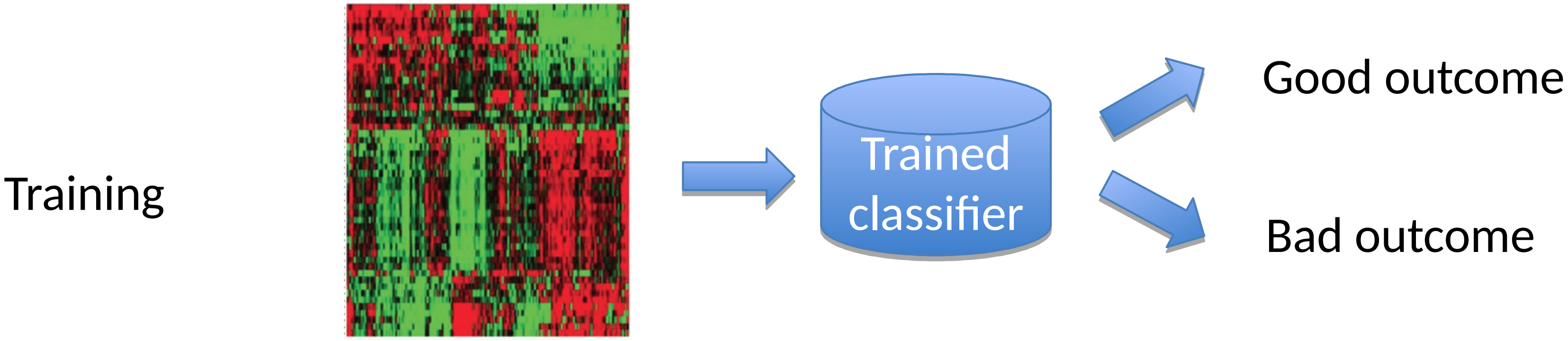
$$\boxed{\lambda \sum_{j=1}^{p} \beta_j^2}$$

Take home

- Regularization and shrinkage are important tools
- Select in function of application
- Keep in mind Occam’s razor (law of parsimony):
  - Keep it simple.
  - Simpler solutions should be prefered to more complex ones

MAQC-II
Best pratices to translate classifiers in the clinic

Goal of personalized medicine

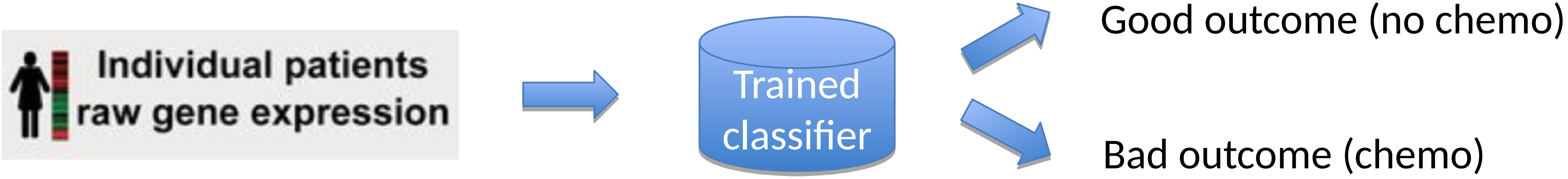


Training

Trained classifier

Good outcome

Bad outcome



Individual patients raw gene expression

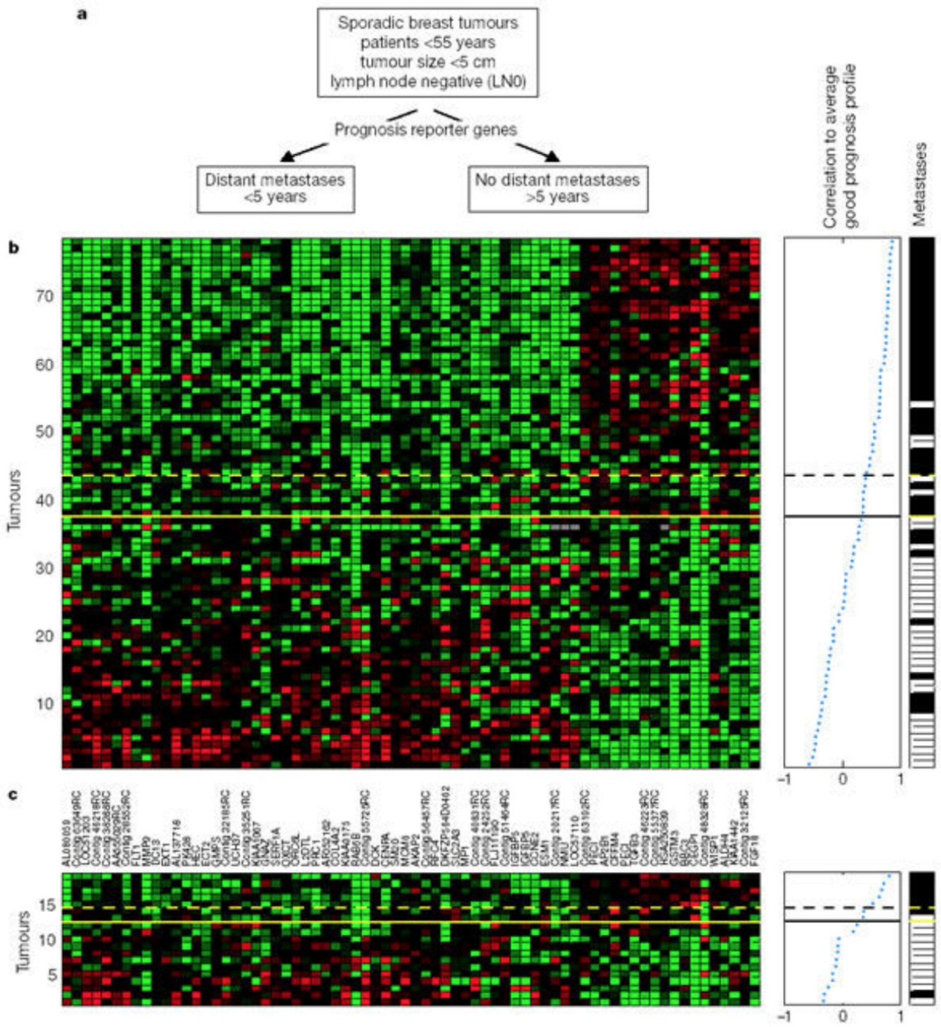
Trained classifier

Good outcome (no chemo)

Bad outcome (chemo)

One good example
Mammaprint (70-gene)

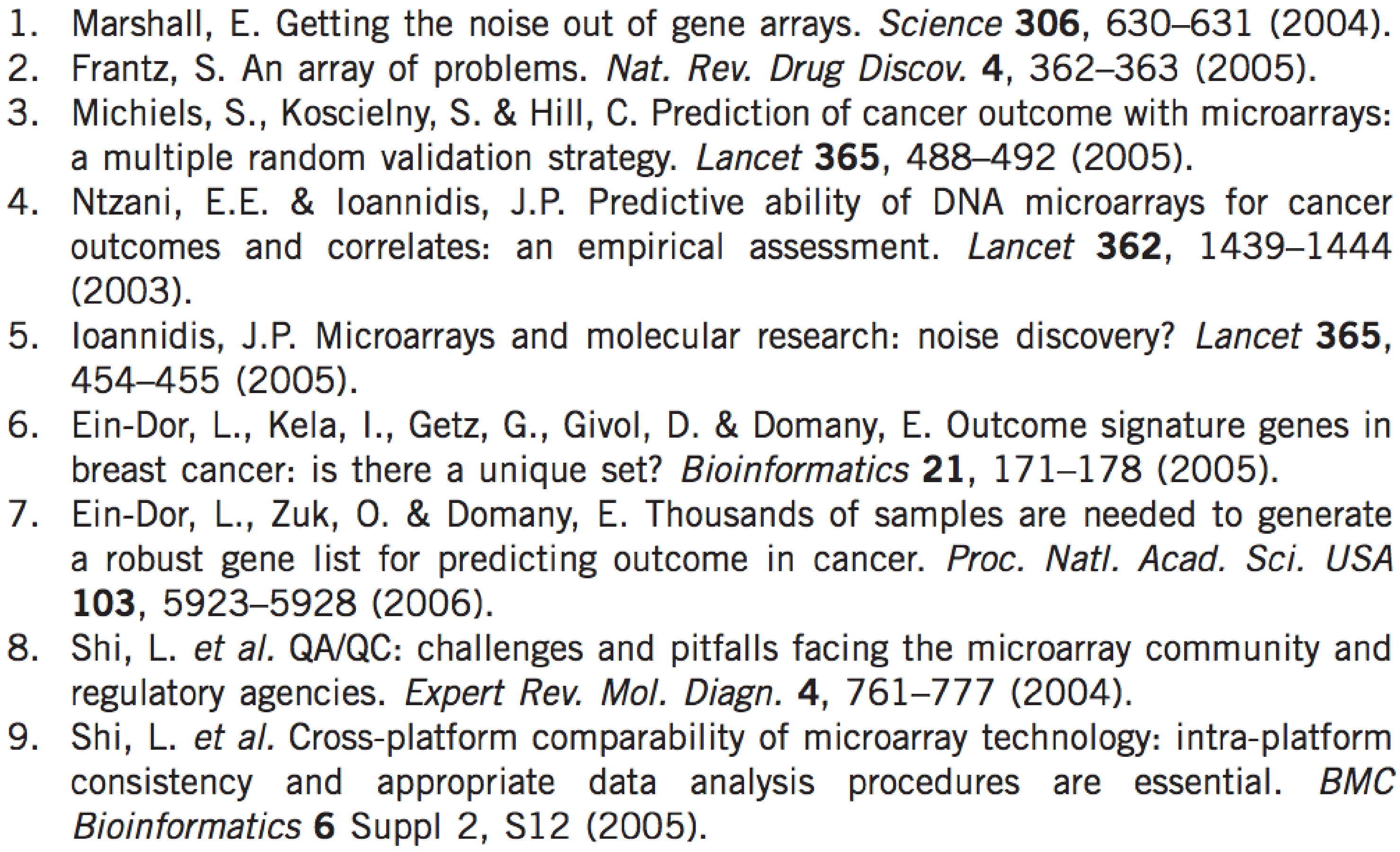
Figure 2: Supervised classification on prognosis signatures.



FDA approved in 2007

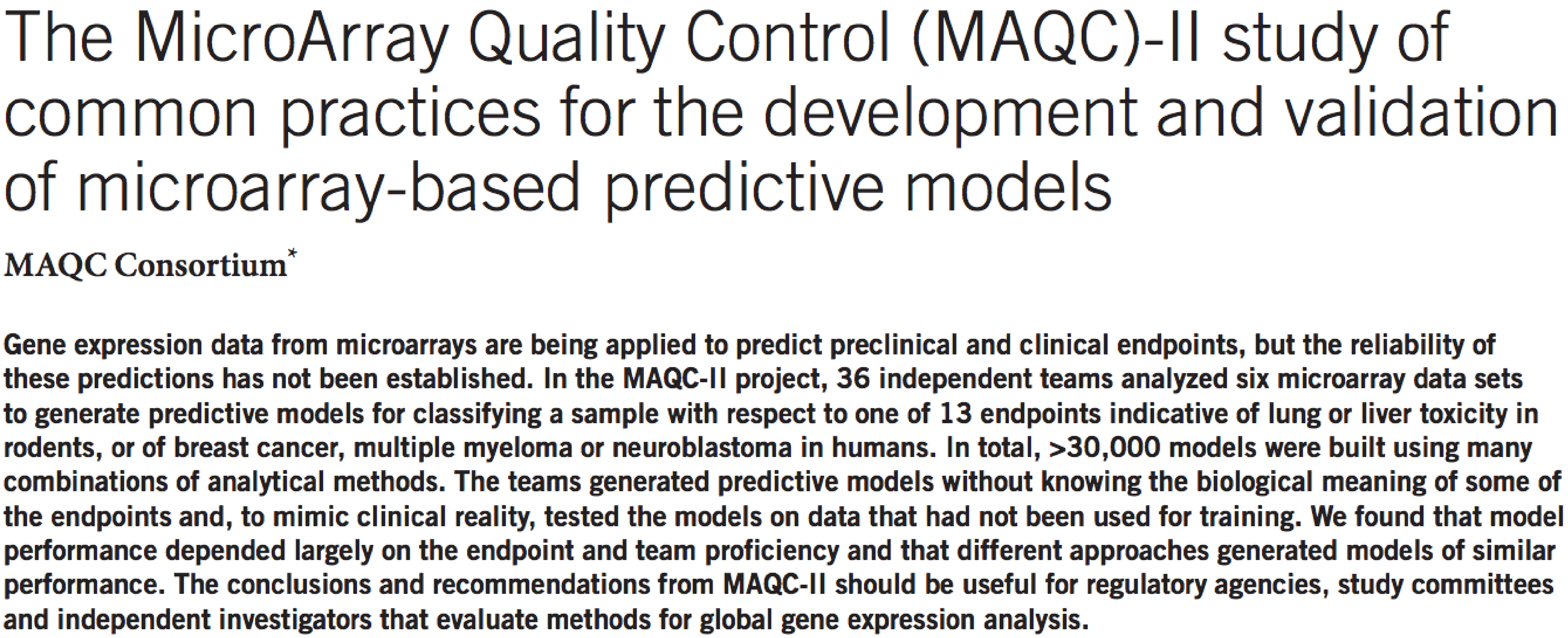
Van’t Veer et al. Nature 2002

Why?



1. Marshall, E. Getting the noise out of gene arrays. *Science* **306**, 630–631 (2004).
2. Frantz, S. An array of problems. *Nat. Rev. Drug Discov.* **4**, 362–363 (2005).
3. Michiels, S., Koscielny, S. & Hill, C. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet* **365**, 488–492 (2005).
4. Ntzani, E.E. & Ioannidis, J.P. Predictive ability of DNA microarrays for cancer outcomes and correlates: an empirical assessment. *Lancet* **362**, 1439–1444 (2003).
5. Ioannidis, J.P. Microarrays and molecular research: noise discovery? *Lancet* **365**, 454–455 (2005).
6. Ein-Dor, L., Kela, I., Getz, G., Givol, D. & Domany, E. Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics* **21**, 171–178 (2005).
7. Ein-Dor, L., Zuk, O. & Domany, E. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proc. Natl. Acad. Sci. USA* **103**, 5923–5928 (2006).
8. Shi, L. *et al.* QA/QC: challenges and pitfalls facing the microarray community and regulatory agencies. *Expert Rev. Mol. Diagn.* **4**, 761–777 (2004).
9. Shi, L. *et al.* Cross-platform comparability of microarray technology: intra-platform consistency and appropriate data analysis procedures are essential. *BMC Bioinformatics* **6** Suppl 2, S12 (2005).

THE MAQC II



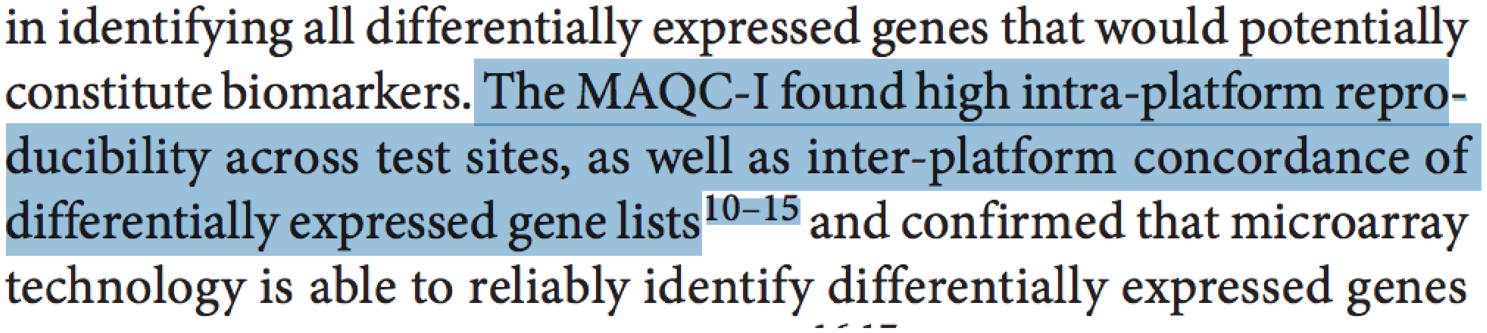
The MicroArray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models

MAQC Consortium*

Gene expression data from microarrays are being applied to predict preclinical and clinical endpoints, but the reliability of these predictions has not been established. In the MAQC-II project, 36 independent teams analyzed six microarray data sets to generate predictive models for classifying a sample with respect to one of 13 endpoints indicative of lung or liver toxicity in rodents, or of breast cancer, multiple myeloma or neuroblastoma in humans. In total, >30,000 models were built using many combinations of analytical methods. The teams generated predictive models without knowing the biological meaning of some of the endpoints and, to mimic clinical reality, tested the models on data that had not been used for training. We found that model performance depended largely on the endpoint and team proficiency and that different approaches generated models of similar performance. The conclusions and recommendations from MAQC-II should be useful for regulatory agencies, study committees and independent investigators that evaluate methods for global gene expression analysis.

What to do with classifiers in the clinic? FDA?

MAQC-I reliability of arrays

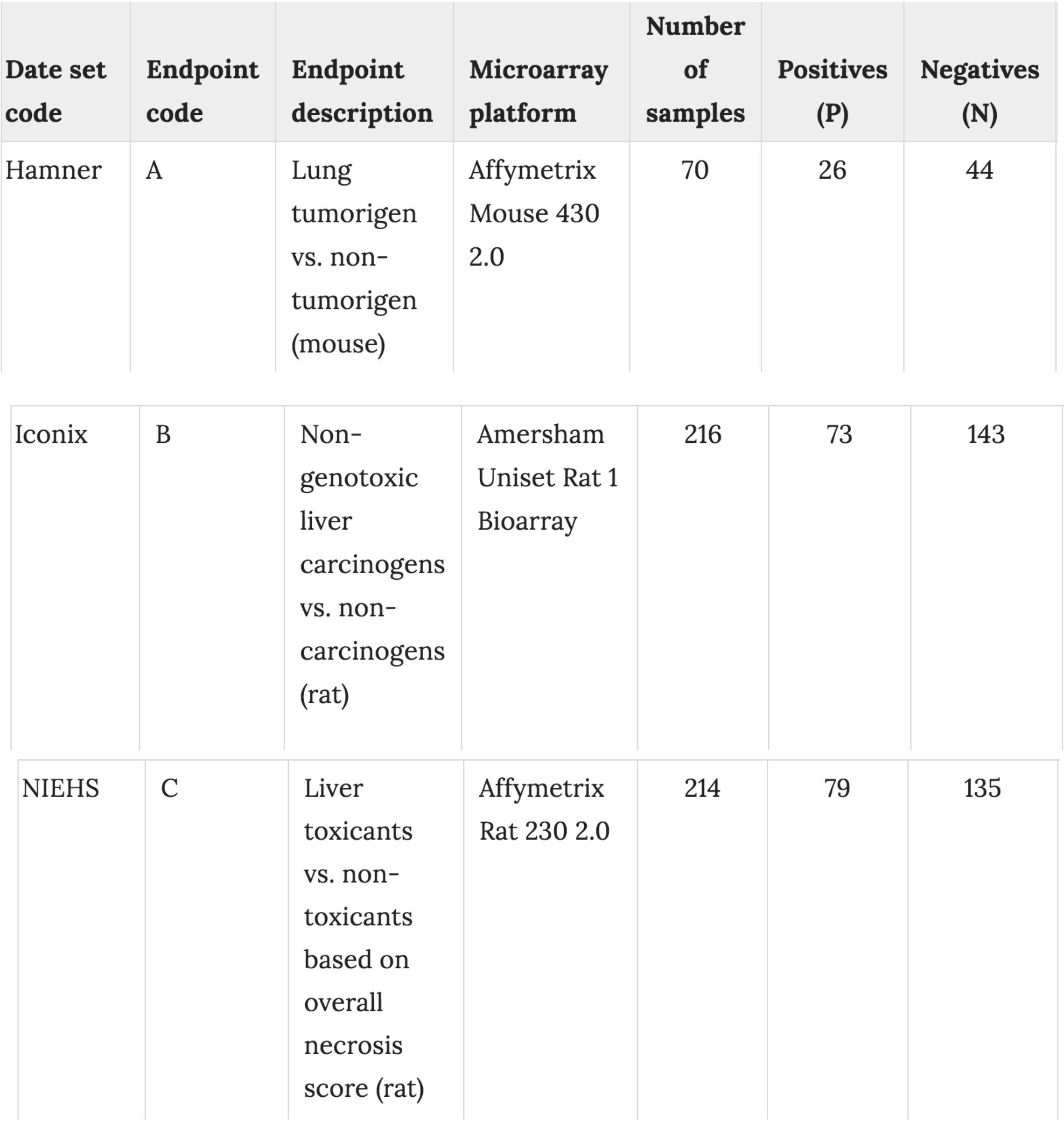


in identifying all differentially expressed genes that would potentially constitute biomarkers. The MAQC-I found high intra-platform reproducibility across test sites, as well as inter-platform concordance of differentially expressed gene lists[10–15] and confirmed that microarray technology is able to reliably identify differentially expressed genes

MAQC-II (challenge, 17 different teams)

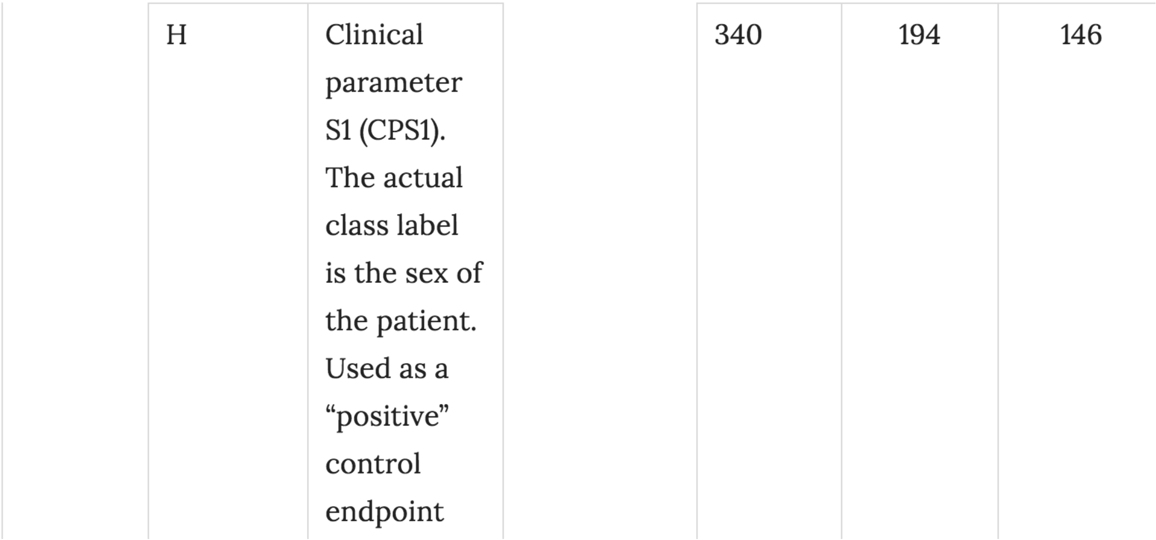
- Different teams applying machine learning supervised algorithms to predict different endpoints.
- Evaluate how good/different they are

Examples of datasets

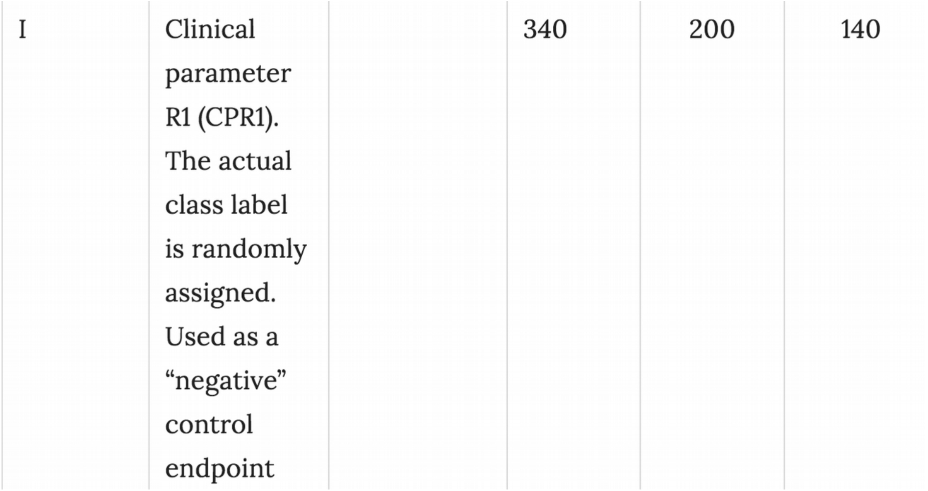


Date set code	Endpoint code	Endpoint description	Microarray platform	Number of samples	Positives (P)	Negatives (N)
Hamner	A	Lung tumorigen vs. non-tumorigen (mouse)	Affymetrix Mouse 430 2.0	70	26	44
Iconix	B	Non-genotoxic liver carcinogens vs. non-carcinogens (rat)	Amersham Uniset Rat 1 Bioarray	216	73	143
NIEHS	C	Liver toxicants vs. non-toxicants based on overall necrosis score (rat)	Affymetrix Rat 230 2.0	214	79	135

Other controls

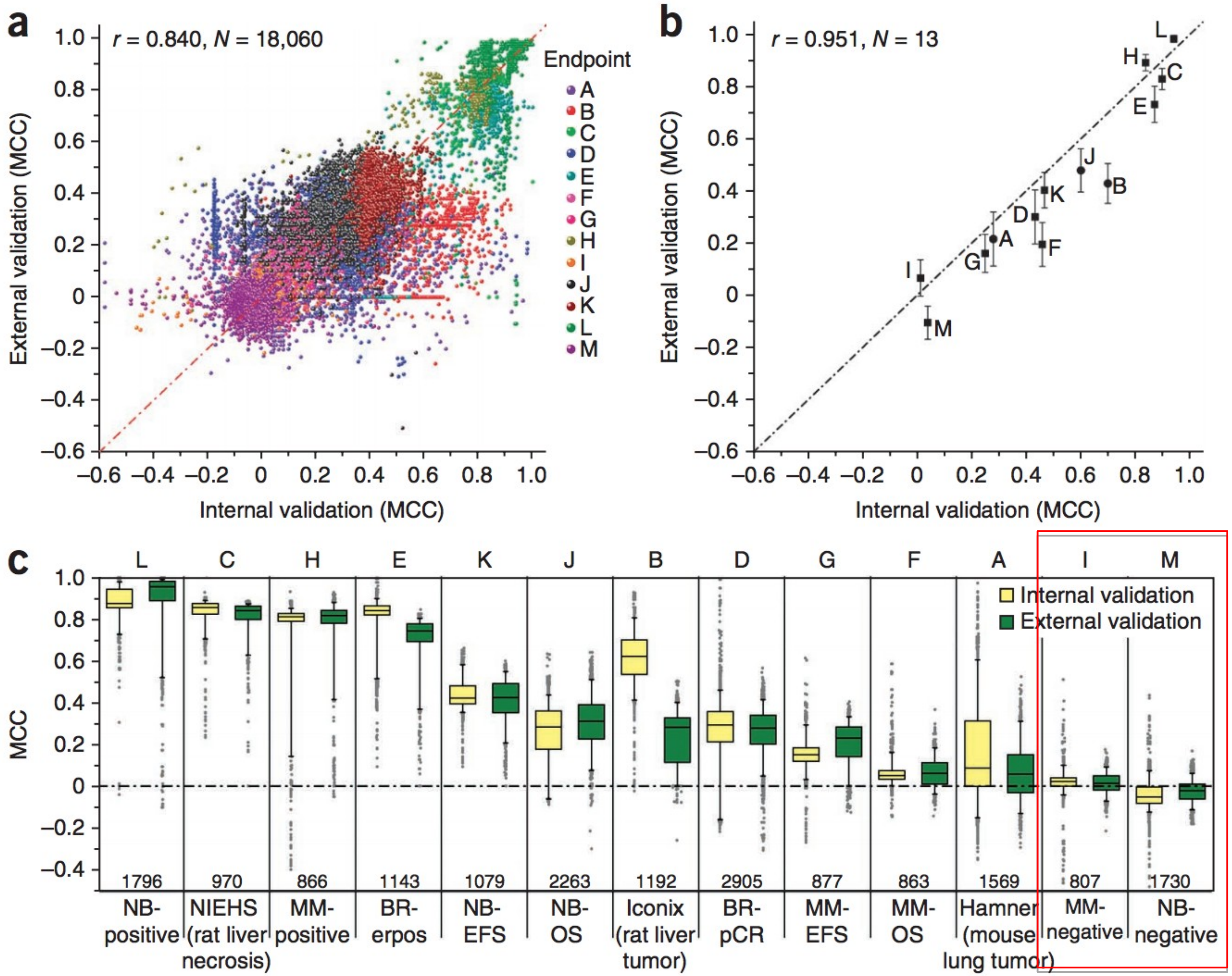


	H	Clinical parameter S1 (CPS1). The actual class label is the sex of the patient. Used as a “positive” control endpoint		340	194	146

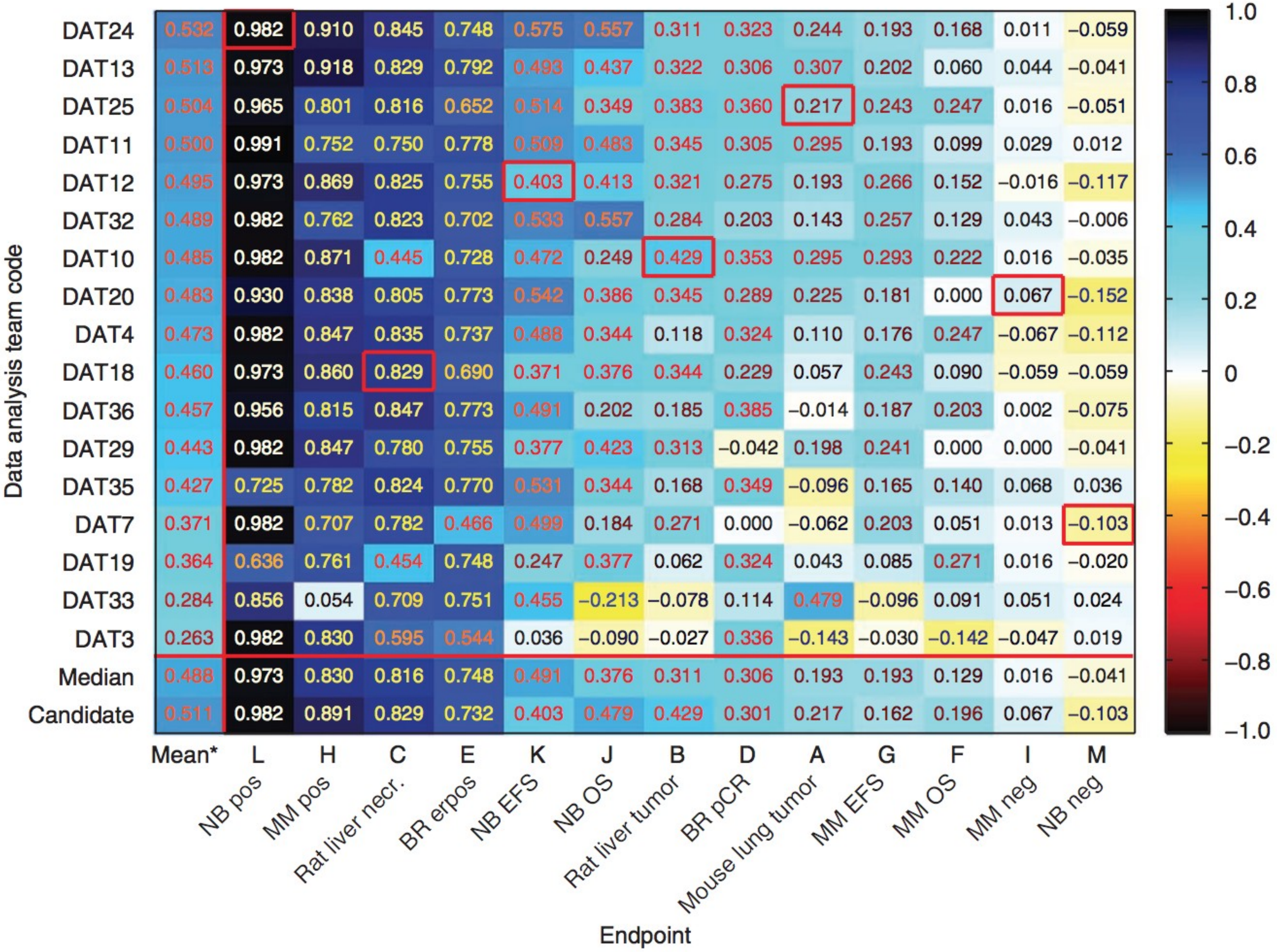


I	Clinical parameter R1 (CPR1). The actual class label is randomly assigned. Used as a “negative” control endpoint		340	200	140

Results [Performance depends on endpoint and can be estimated during training]



Results [Data analysis teams show different proficiency]



Data analysis team code	Mean*	L NB pos	H MM pos	C Rat liver necr.	E BR erpos	K NB EFS	J NB OS	B Rat liver tumor	D BR pCR	A Mouse lung tumor	G MM EFS	F MM OS	I MM neg	M NB neg
DAT24	0.532	0.982	0.910	0.845	0.748	0.575	0.557	0.311	0.323	0.244	0.193	0.168	0.011	−0.059
DAT13	0.513	0.973	0.918	0.829	0.792	0.493	0.437	0.322	0.306	0.307	0.202	0.060	0.044	−0.041
DAT25	0.504	0.965	0.801	0.816	0.652	0.514	0.349	0.383	0.360	0.217	0.243	0.247	0.016	−0.051
DAT11	0.500	0.991	0.752	0.750	0.778	0.509	0.483	0.345	0.305	0.295	0.193	0.099	0.029	0.012
DAT12	0.495	0.973	0.869	0.825	0.755	0.403	0.413	0.321	0.275	0.193	0.266	0.152	−0.016	−0.117
DAT32	0.489	0.982	0.762	0.823	0.702	0.533	0.557	0.284	0.203	0.143	0.257	0.129	0.043	−0.006
DAT10	0.485	0.982	0.871	0.445	0.728	0.472	0.249	0.429	0.353	0.295	0.293	0.222	0.016	−0.035
DAT20	0.483	0.930	0.838	0.805	0.773	0.542	0.386	0.345	0.289	0.225	0.181	0.000	0.067	−0.152
DAT4	0.473	0.982	0.847	0.835	0.737	0.488	0.344	0.118	0.324	0.110	0.176	0.247	−0.067	−0.112
DAT18	0.460	0.973	0.860	0.829	0.690	0.371	0.376	0.344	0.229	0.057	0.243	0.090	−0.059	−0.059
DAT36	0.457	0.956	0.815	0.847	0.773	0.491	0.202	0.185	0.385	−0.014	0.187	0.203	0.002	−0.075
DAT29	0.443	0.982	0.847	0.780	0.755	0.377	0.423	0.313	−0.042	0.198	0.241	0.000	0.000	−0.041
DAT35	0.427	0.725	0.782	0.824	0.770	0.531	0.344	0.168	0.349	−0.096	0.165	0.140	0.068	0.036
DAT7	0.371	0.982	0.707	0.782	0.466	0.499	0.184	0.271	0.000	−0.062	0.203	0.051	0.013	−0.103
DAT19	0.364	0.636	0.761	0.454	0.748	0.247	0.377	0.062	0.324	0.043	0.085	0.271	0.016	−0.020
DAT33	0.284	0.856	0.054	0.709	0.751	0.455	−0.213	−0.078	0.114	0.479	−0.096	0.091	0.051	0.024
DAT3	0.263	0.982	0.830	0.595	0.544	0.036	−0.090	−0.027	0.336	−0.143	−0.030	−0.142	−0.047	0.019
Median	0.488	0.973	0.830	0.816	0.748	0.491	0.376	0.311	0.306	0.193	0.193	0.129	0.016	−0.041
Candidate	0.511	0.982	0.891	0.829	0.732	0.403	0.479	0.429	0.301	0.217	0.162	0.196	0.067	−0.103

Take home message

- Hard problems are hard for everyone.
  - There is no magic approach. You are limited by the signal in your data

Kernel trick

Sometime data cannot be mapped using a linear hyperplane (eg. SVM)

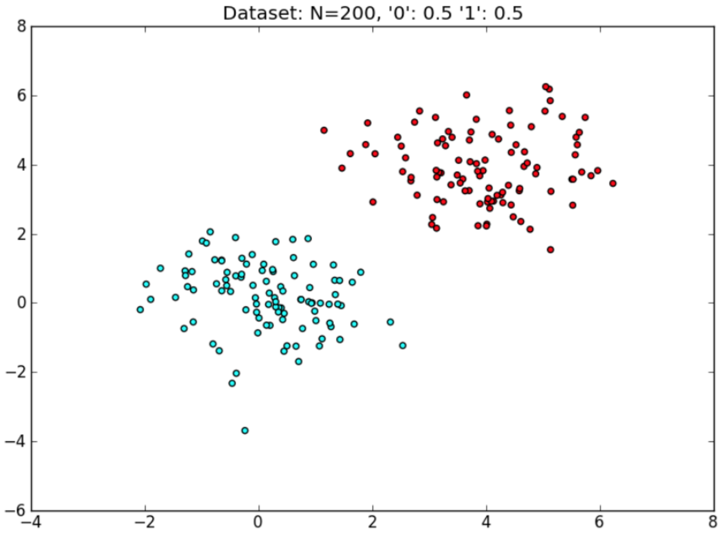

Dataset: N=200, '0': 0.5 '1': 0.5

Figure 1: A two-class, linearly separable dataset.

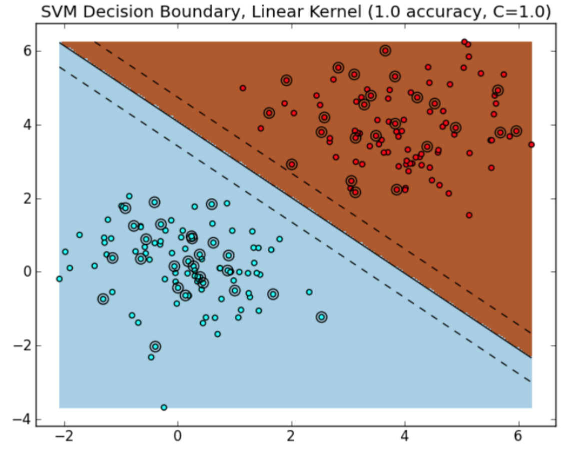

SVM Decision Boundary, Linear Kernel (1.0 accuracy, C=1.0)

Figure 2: The Decision Boundary of a Linear SVM on a linearly-separable dataset. The solid line is the boundary. The SVM is trained on 75% of the dataset, and evaluated on the remaining 25%. Circled data points are from the test set.

http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Sometime data cannot be mapped using a linear hyperplane (eg. SVM)

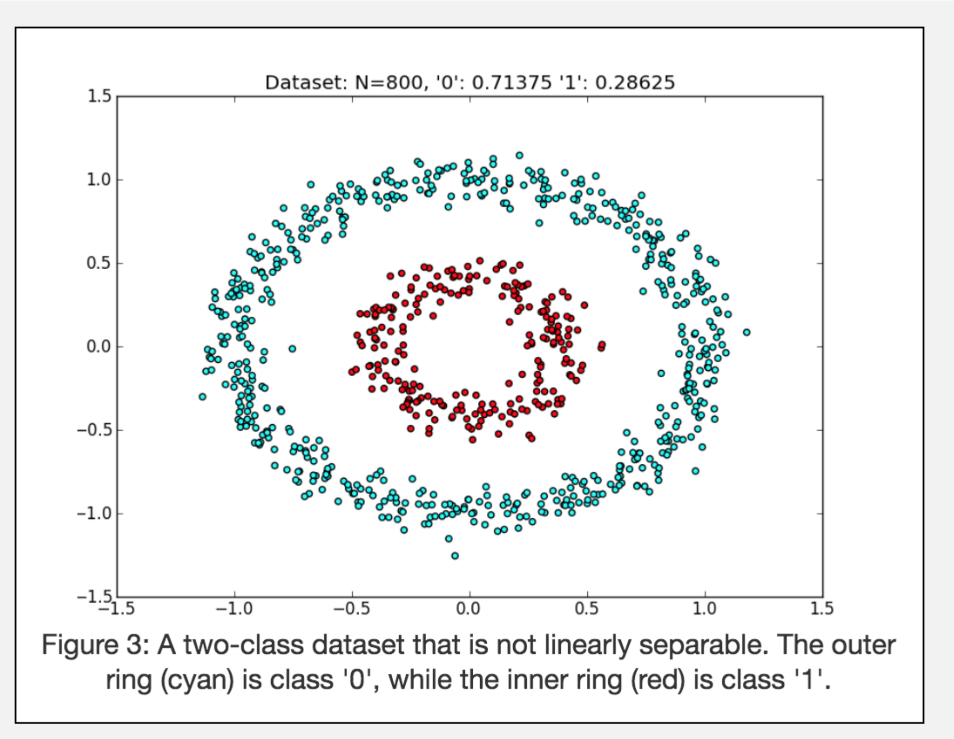


Figure 3: A two-class dataset that is not linearly separable. The outer ring (cyan) is class '0', while the inner ring (red) is class '1'.

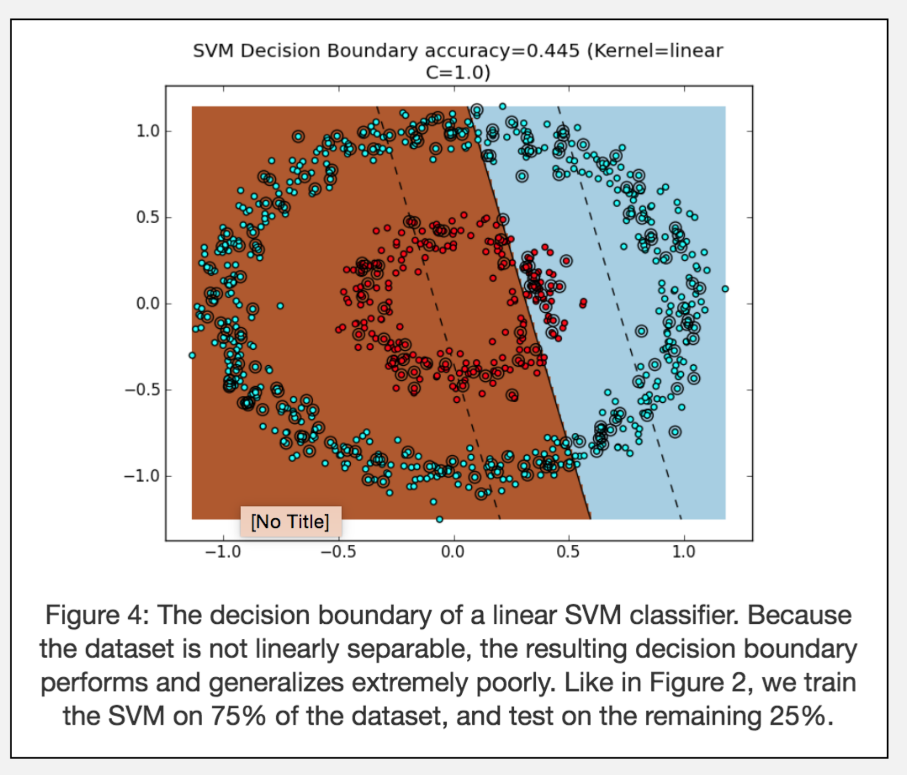


Figure 4: The decision boundary of a linear SVM classifier. Because the dataset is not linearly separable, the resulting decision boundary performs and generalizes extremely poorly. Like in Figure 2, we train the SVM on 75% of the dataset, and test on the remaining 25%.

http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Separable in higher dimension

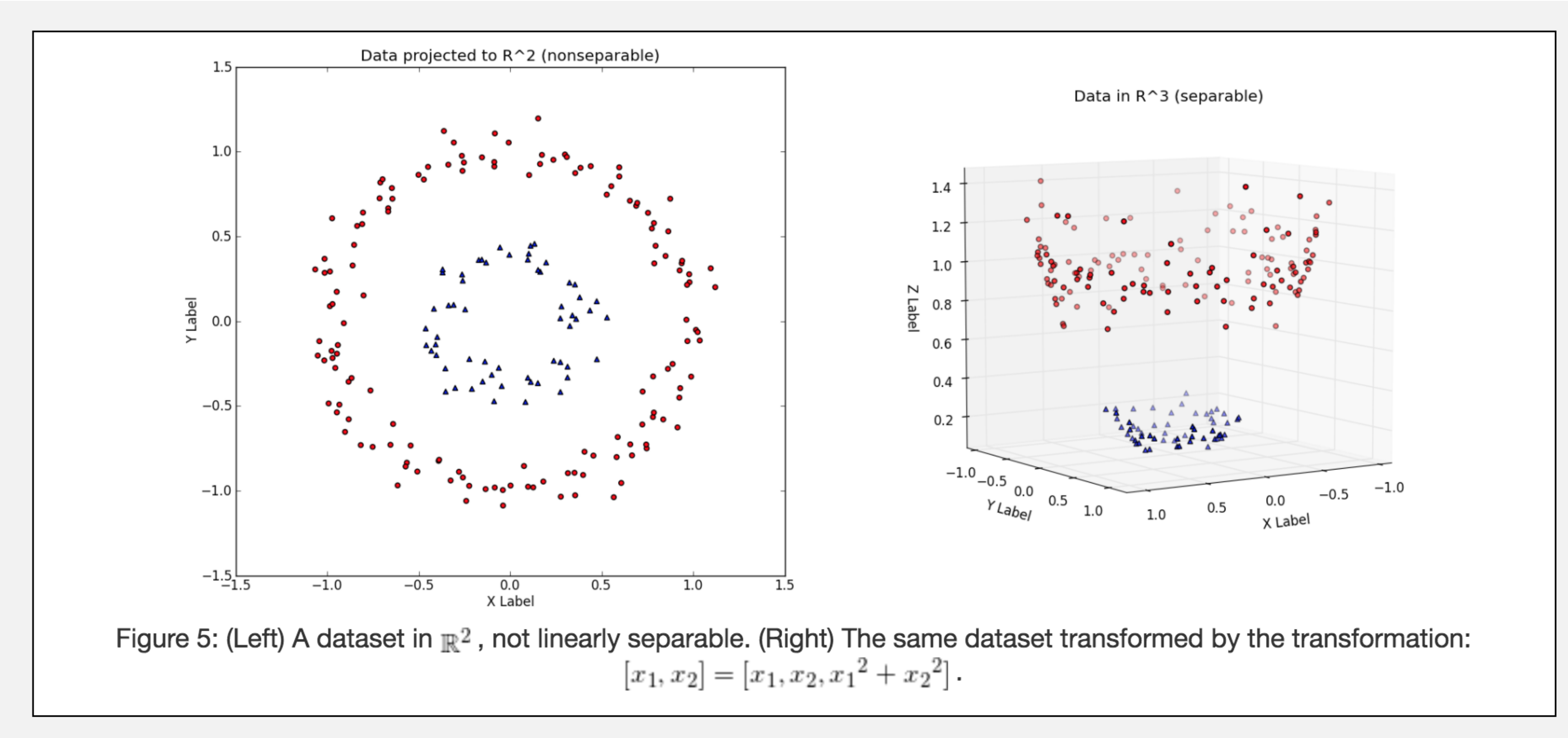


Figure 5: (Left) A dataset in $\mathbb{R}^2$, not linearly separable. (Right) The same dataset transformed by the transformation: $[x_1, x_2] = [x_1, x_2, {x_1}^2 + {x_2}^2]$.

http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Separable in higher dimension

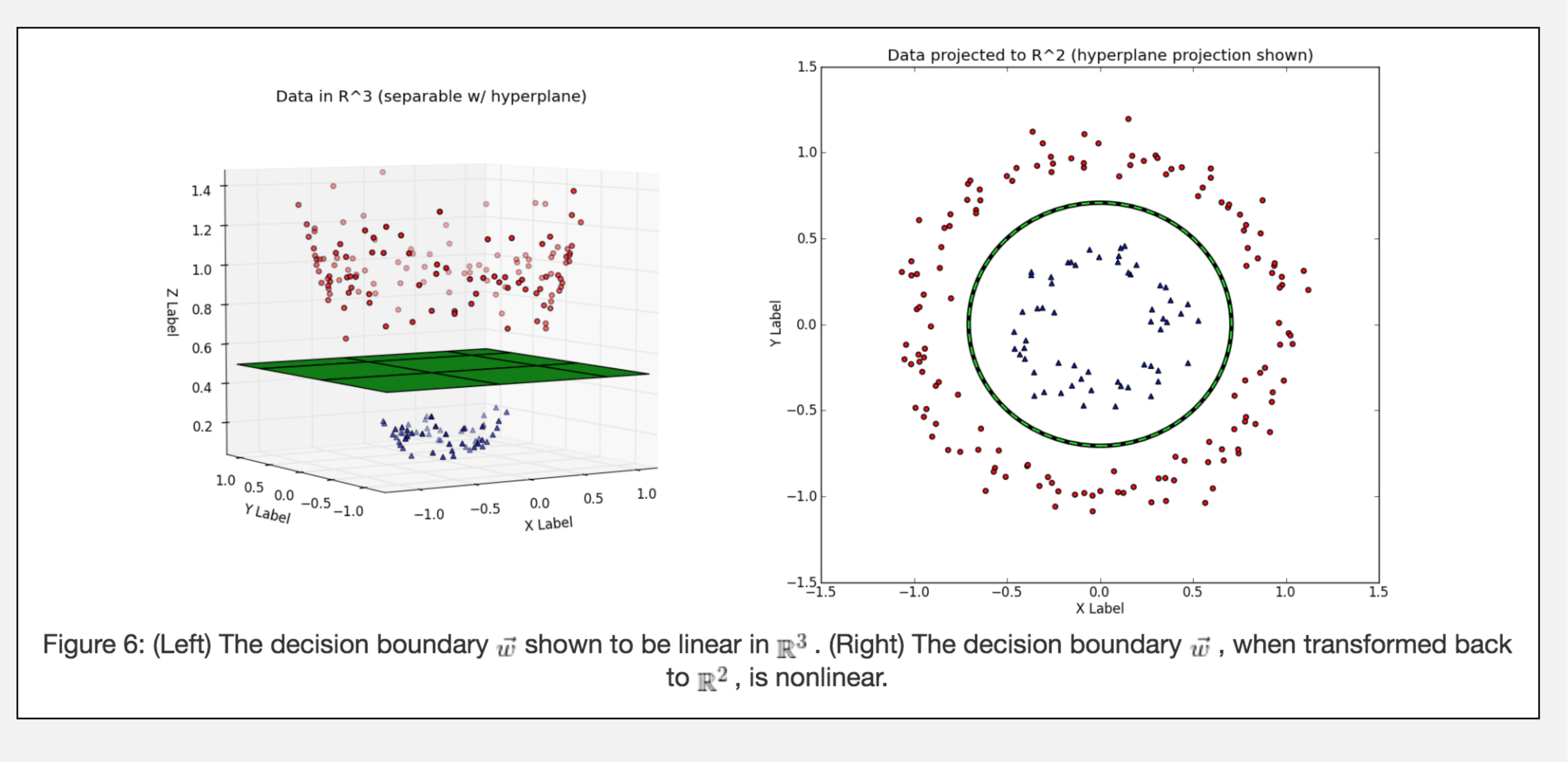


Figure 6: (Left) The decision boundary $\vec{w}$ shown to be linear in $\mathbb{R}^3$. (Right) The decision boundary $\vec{w}$, when transformed back to $\mathbb{R}^2$, is nonlinear.

http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Different kernels

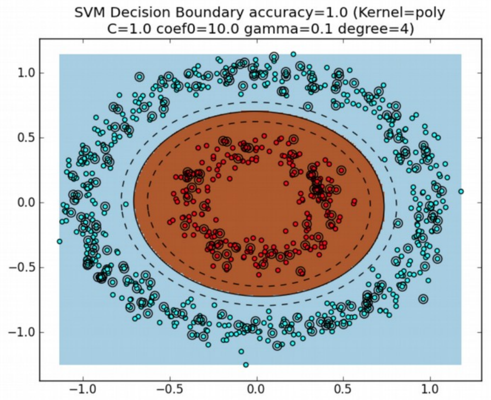


Figure 6: The decision boundary with a Polynomial kernel.

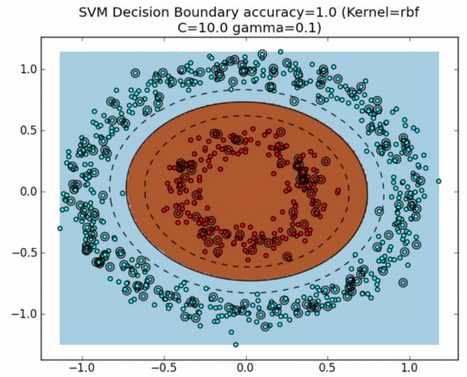


Figure 7: The decision boundary with a Radial Basis Function (RBF) kernel.

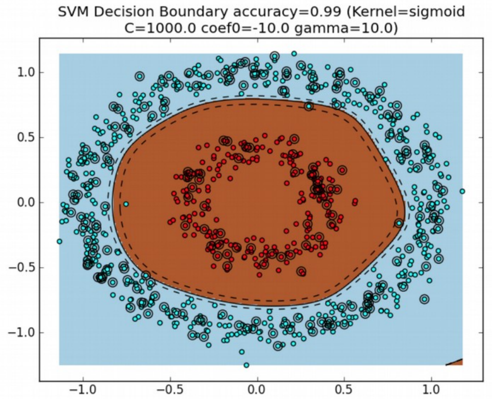


Figure 8: The decision boundary with a Sigmoid kernel.

linear:

u'*v

polynomial:

(gamma*u'*v + coef0)^degree

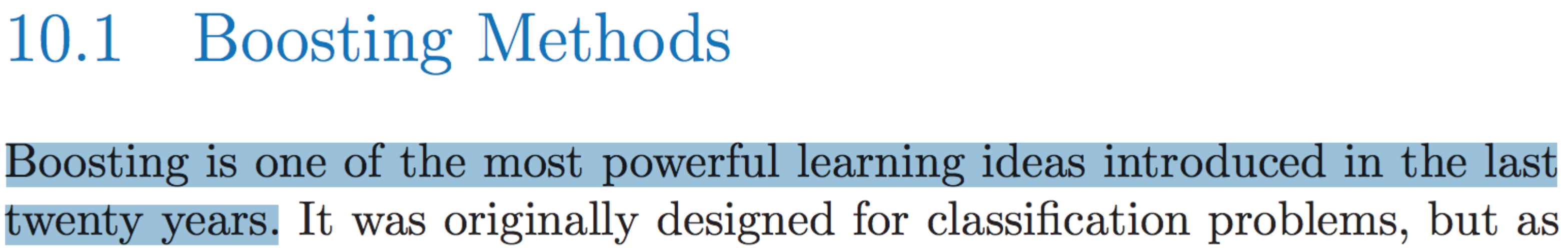
radial basis:

exp(-gamma*|u-v|^2)

sigmoid:

tanh(gamma*u'*v + coef0)

Boosting



10.1 Boosting Methods

Boosting is one of the most powerful learning ideas introduced in the last twenty years. It was originally designed for classification problems, but as

The elements of statistical learning

AdaBoost, Freund and Schapire 1997

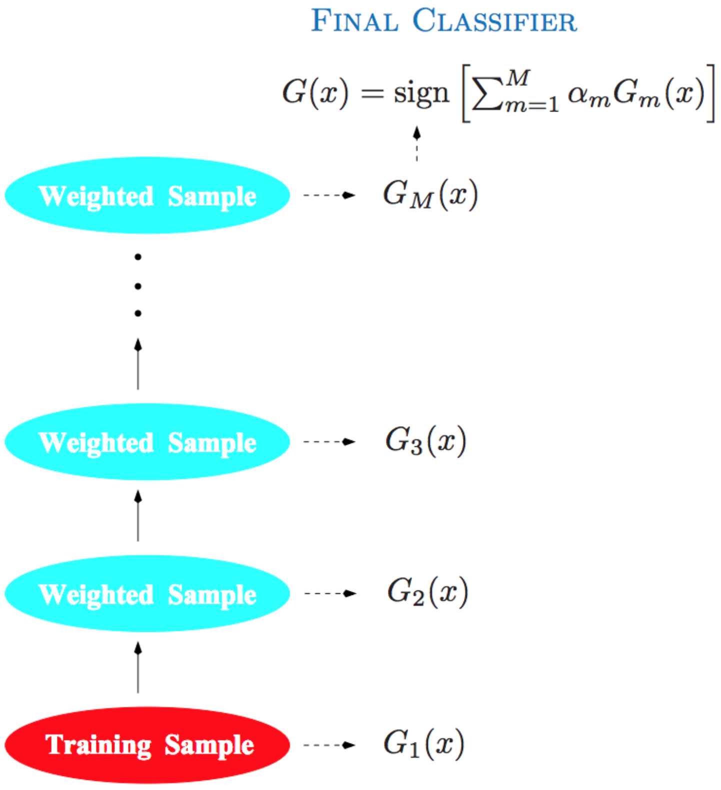
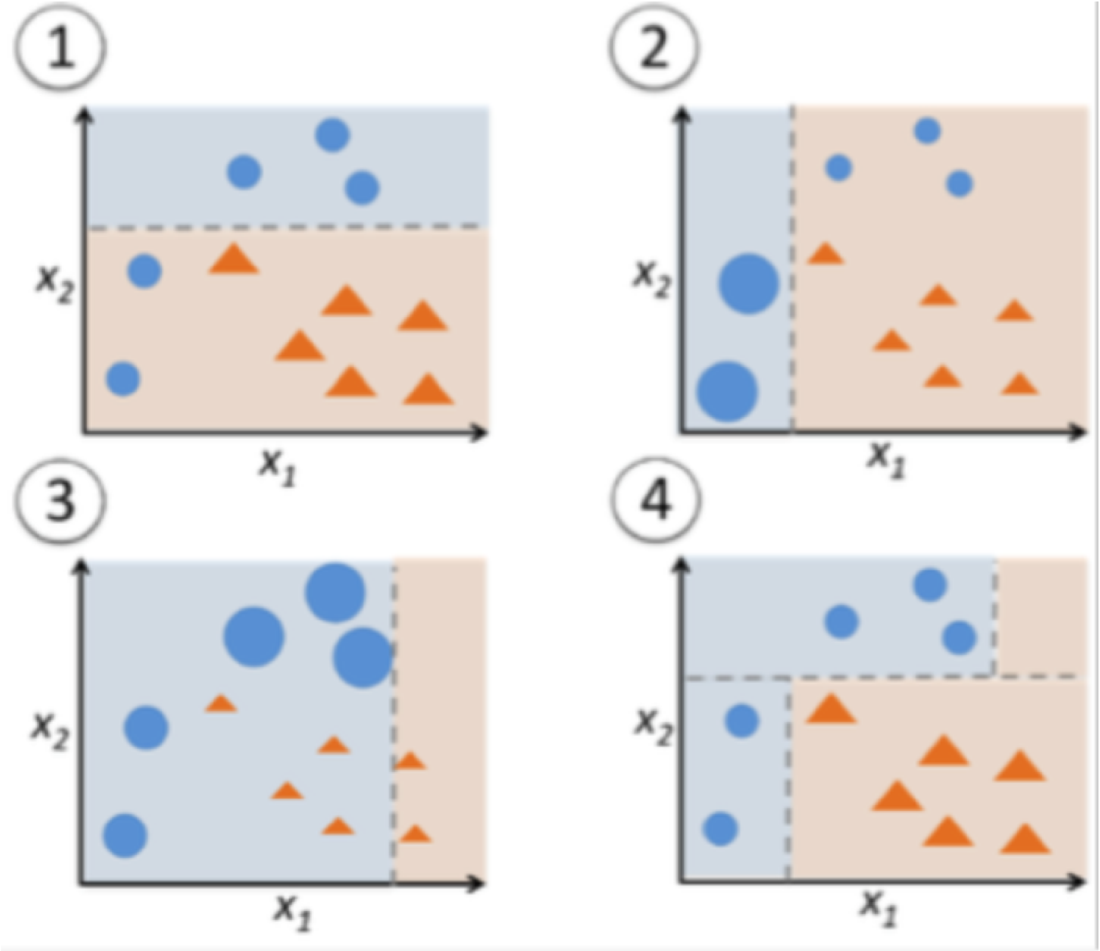


FIGURE 10.1. *Schematic of AdaBoost. Classifiers are trained on weighted versions of the dataset, and then combined to produce a final prediction.*

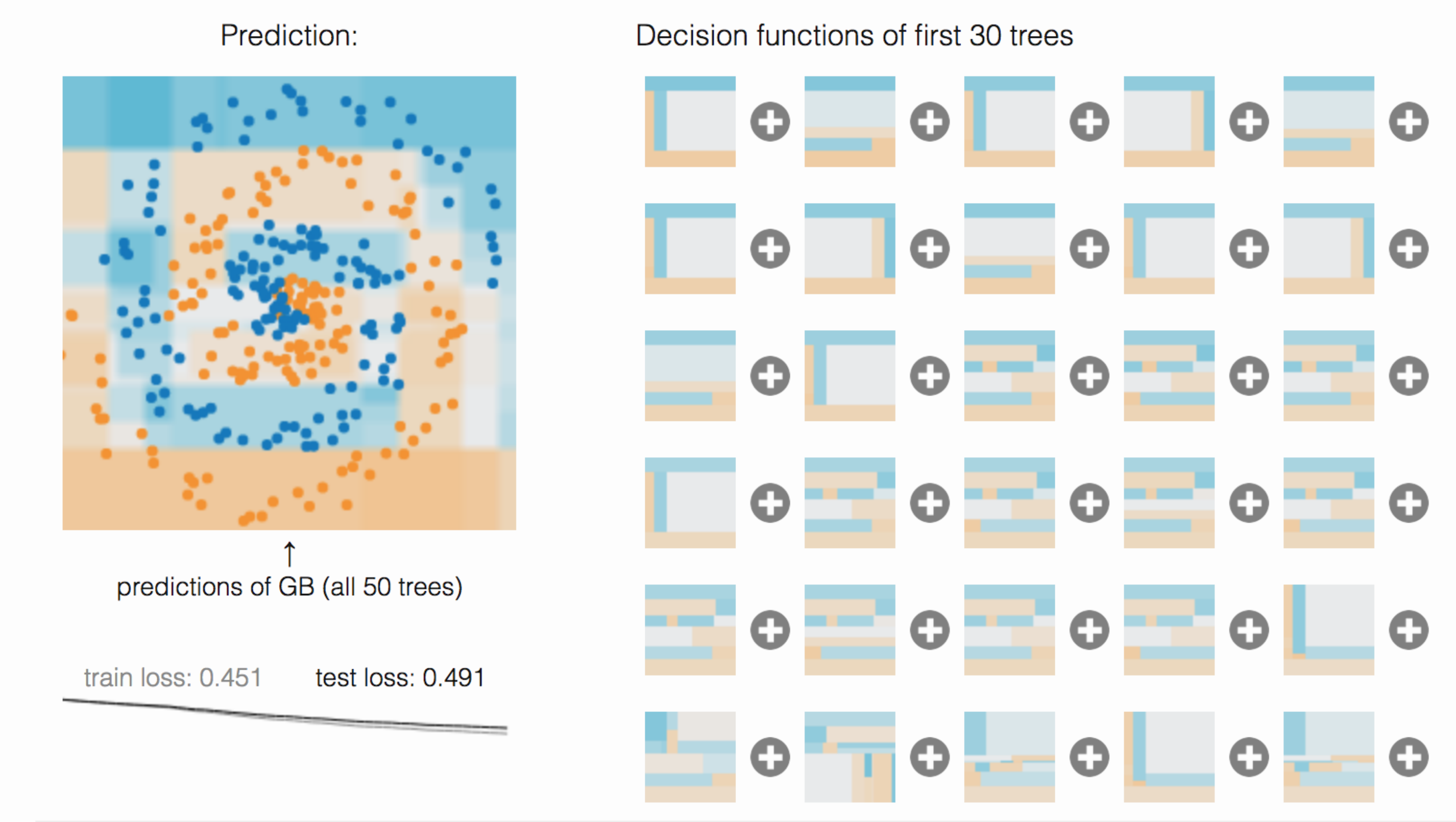
The elements of statistical learning

Example



https://sebastianraschka.com/faq/docs/bagging-boosting-rf.html

Gradient Boosting Models


Prediction:
Decision functions of first 30 trees
↑
predictions of GB (all 50 trees)
train loss: 0.451
test loss: 0.491

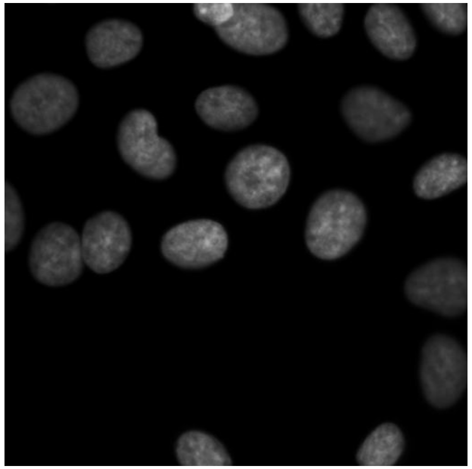
http://arogozhnikov.github.io/2016/07/05/gradient_boosting_playground.html

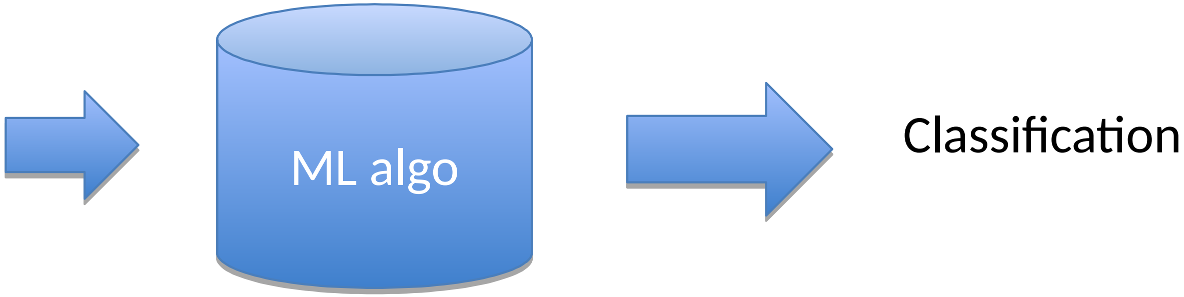
EXtreme Gradient Boosting (XGBoost)

- Currently one of the best performing method in Kaggle competition
- http://xgboost.readthedocs.io/en/latest/
- You should have a look

Image Analysis :
Mostly how do you extract features to feed your ML algorithm

ML base on images

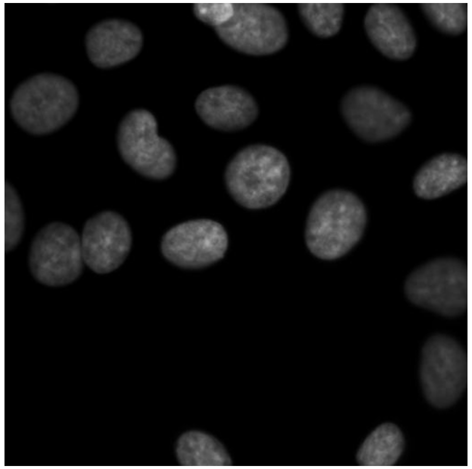


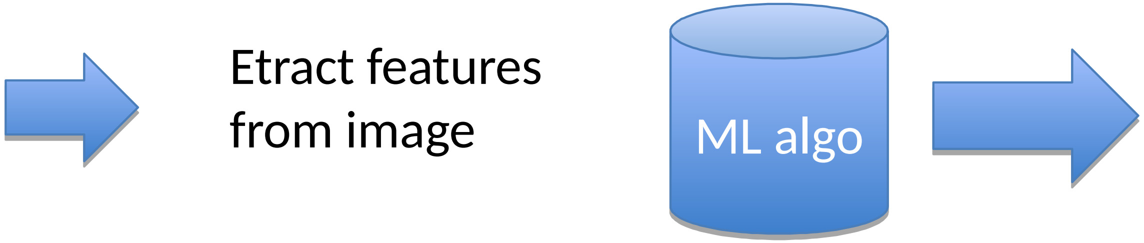


ML algo

Classification

ML base on images





Etract features from image

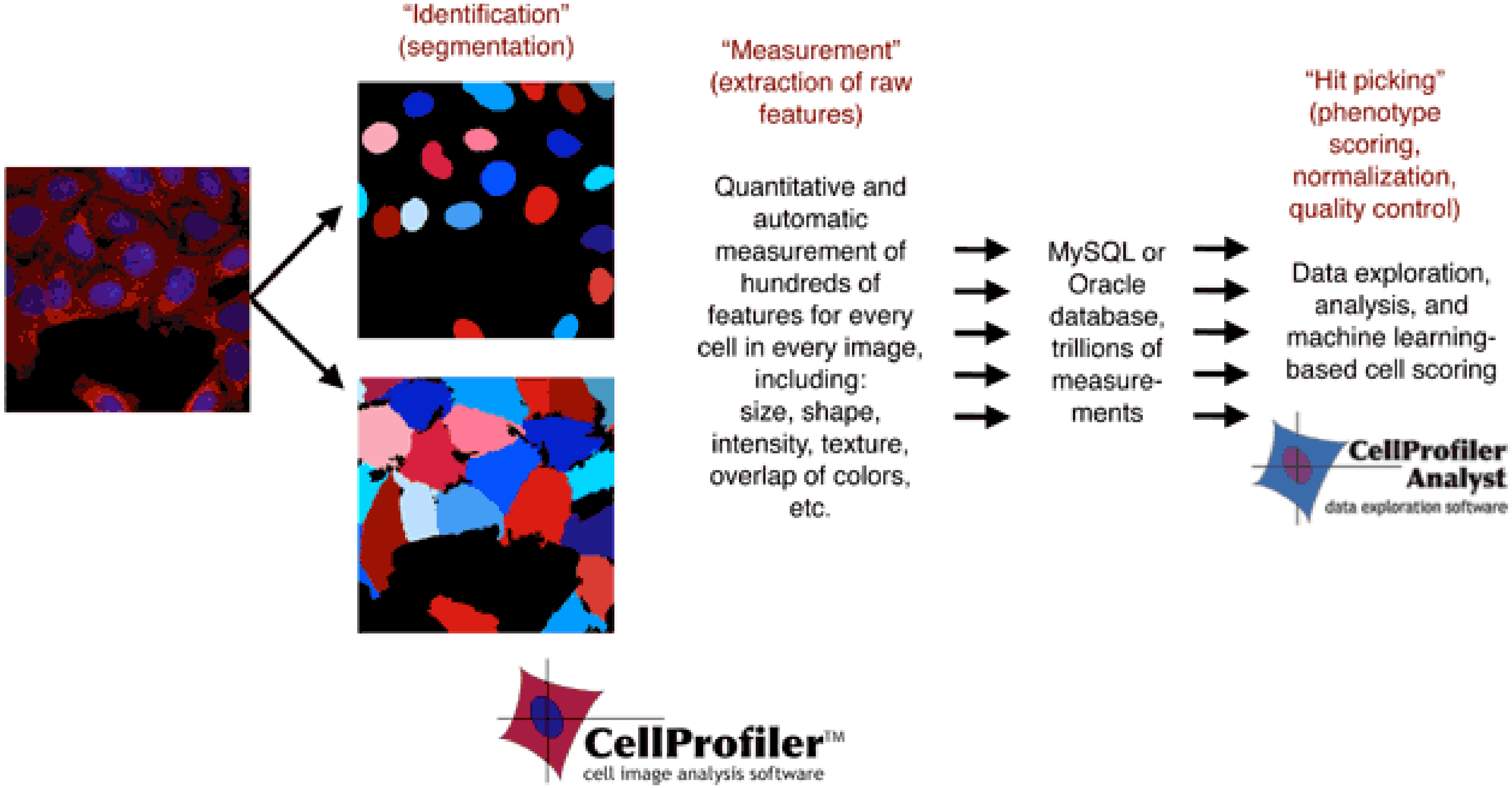
ML algo

Classification

Different tools to extract features

- Cell profiler
  - Mostly for cells
- Matlab
  - Powerful image processing toolbox. Not specific for systems biology. Might take time
- Ilastik
  - Machine learning for images
- Phenoripper
  - Segmentation free
- Directly in R:
  - EBImage
  - imageHTS

Cell Profiler



"Identification" (segmentation)

"Measurement" (extraction of raw features)

Quantitative and automatic measurement of hundreds of features for every cell in every image, including: size, shape, intensity, texture, overlap of colors, etc.

MySQL or Oracle database, trillions of measure-ments

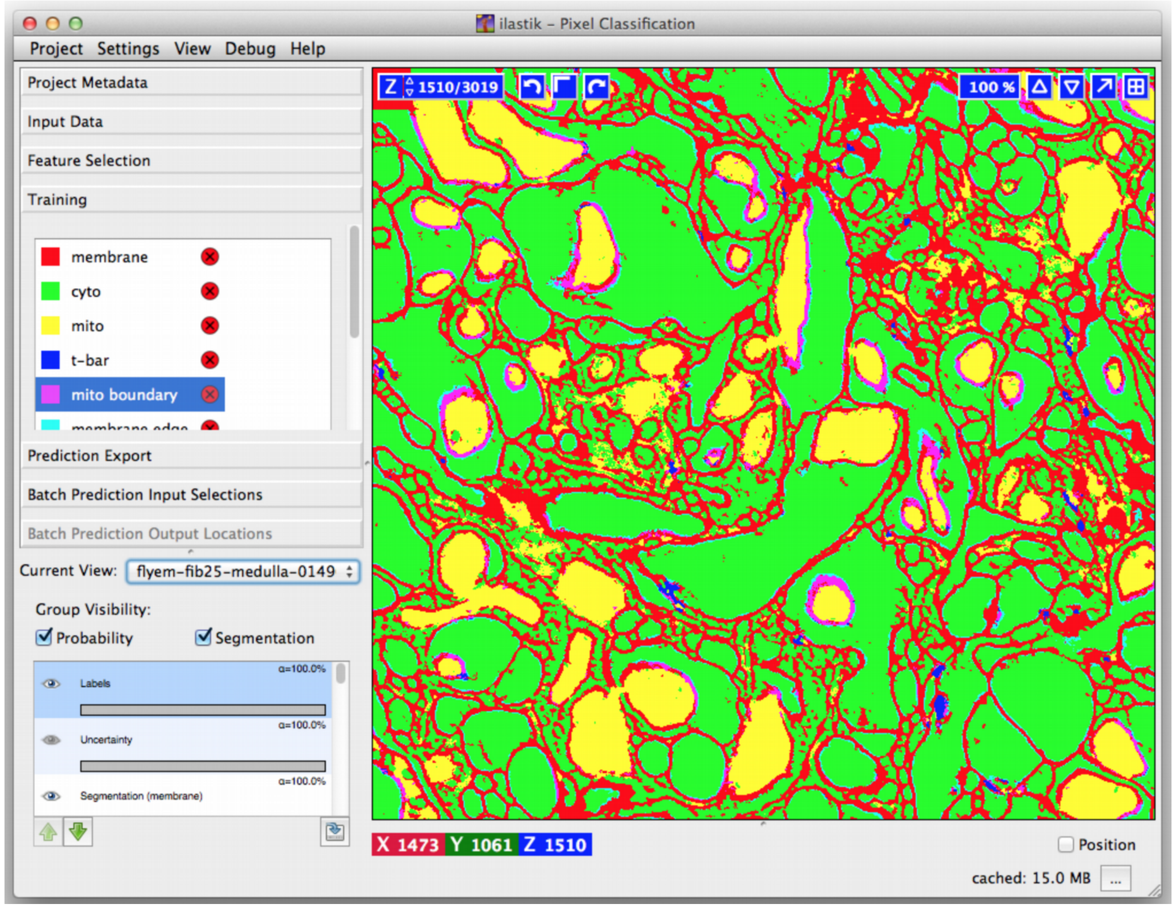
"Hit picking" (phenotype scoring, normalization, quality control)

Data exploration, analysis, and machine learning-based cell scoring

http://cellprofiler.org/

http://cellprofiler.org/cp-analyst/

ilastik



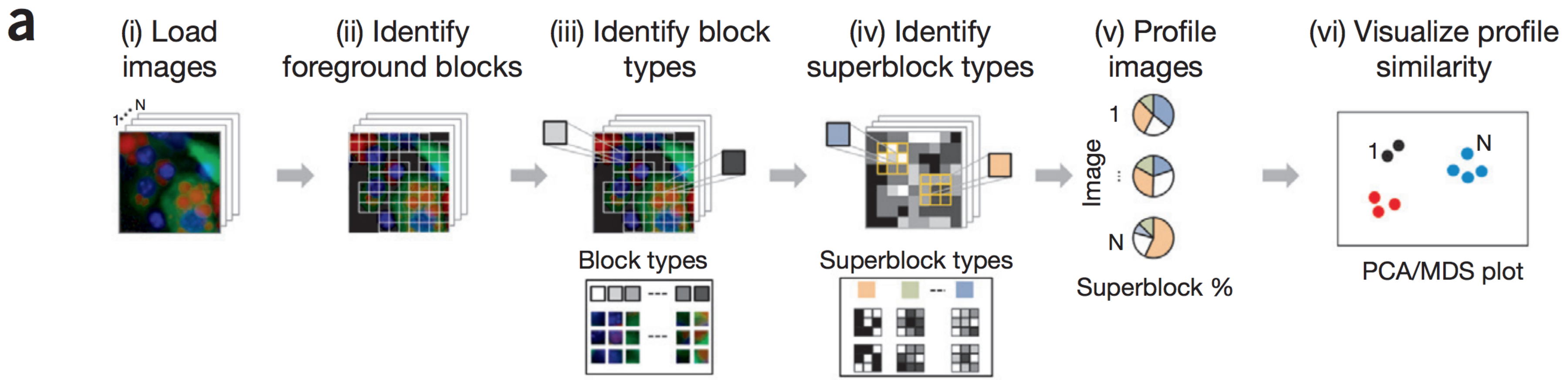
Current View: flyem-fib25-medulla-0149

☑ Probability ☑ Segmentation

http://ilastik.org/

PhenoRipper

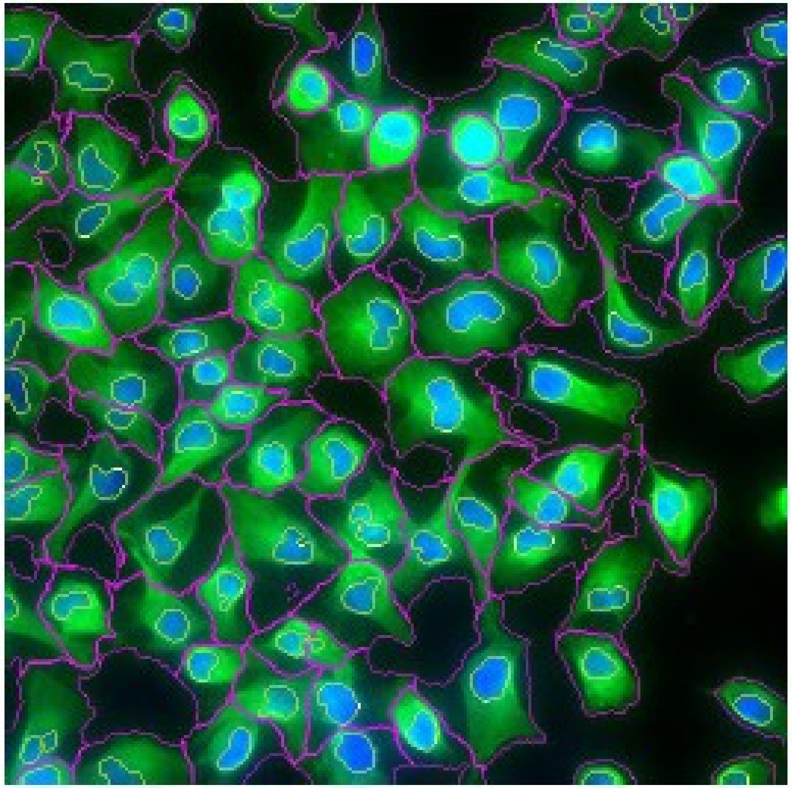
- Segmentation free image analysis
  - Just extract block features (composition in colors) an co-occurrence within 3 by 3 grids.

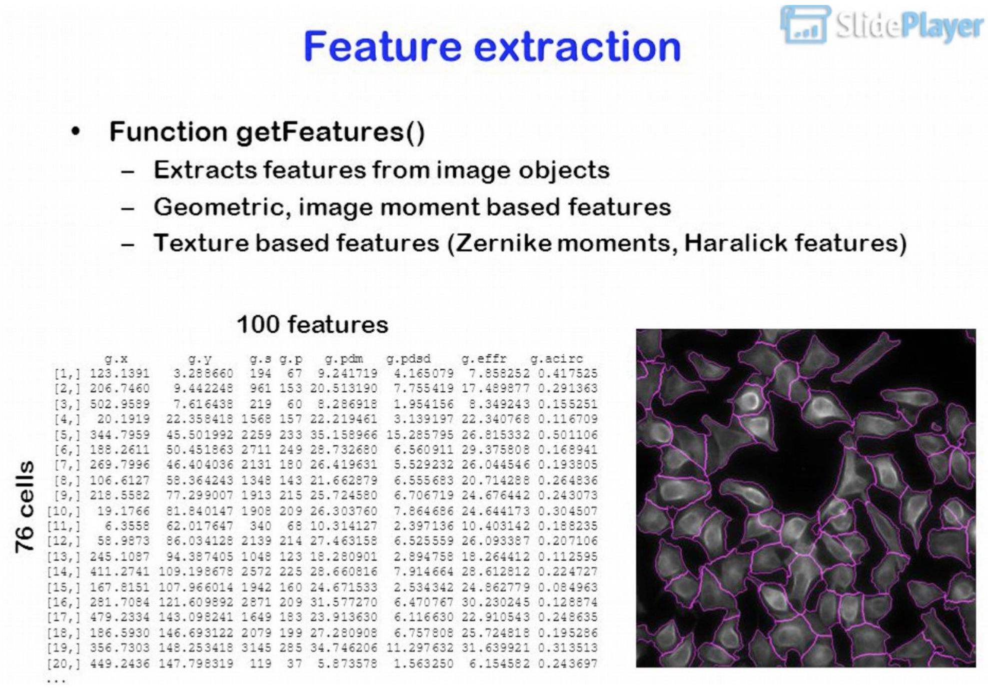

a
(i) Load images
1
N
(ii) Identify foreground blocks
(iii) Identify block types
Block types
(iv) Identify superblock types
Superblock types
(v) Profile images
Image
1
N
Superblock %
(vi) Visualize profile similarity
1
N
PCA/MDS plot

http://awlab.ucsf.edu/Web_Site/PhenoRipper/default.htm

EBImage

- Matlab “like” but in R





Feature extraction

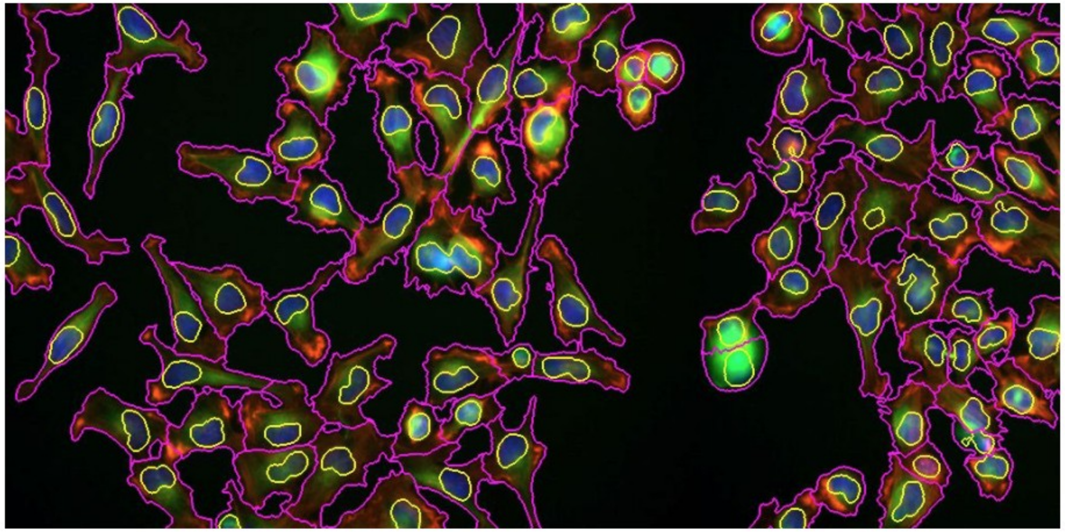
- Function getFeatures()
  - Extracts features from image objects
  - Geometric, image moment based features
  - Texture based features (Zernike moments, Haralick features)

100 features

76 cells

```
            g.x       g.y  g.s g.p     g.pdm    g.pdsd    g.effr   g.acirc
 [1,] 123.1391   3.288660  194  67  9.241719  4.165079  7.858252 0.417525
 [2,] 206.7460   9.442248  961 153 20.513190  7.755419 17.489877 0.291363
 [3,] 502.9589   7.616438  219  60  8.286918  1.954156  8.349243 0.155251
 [4,]  20.1919  22.358418 1568 157 22.219461  3.139197 22.340768 0.116709
 [5,] 344.7959  45.501992 2259 233 35.158966 15.285795 26.815332 0.501106
 [6,] 188.2611  50.451863 2711 249 28.732680  6.560911 29.375808 0.168941
 [7,] 269.7996  46.404036 2131 180 26.419631  5.529232 26.044546 0.193805
 [8,] 106.6127  58.364243 1348 143 21.662879  6.555683 20.714288 0.264836
 [9,] 218.5582  77.299007 1913 215 25.724580  6.706719 24.676442 0.243073
[10,]  19.1766  81.840147 1908 209 26.303760  7.864686 24.644173 0.304507
[11,]   6.3558  62.017647  340  68 10.314127  2.397136 10.403142 0.188235
[12,]  58.9873  86.034128 2139 214 27.463158  6.525559 26.093387 0.207106
[13,] 245.1087  94.387405 1048 123 18.280901  2.894758 18.264412 0.112595
[14,] 411.2741 109.198678 2572 225 28.660816  7.914664 28.612812 0.224727
[15,] 167.8151 107.966014 1942 160 24.671533  2.534342 24.862779 0.084963
[16,] 281.7084 121.609892 2871 209 31.577270  6.470767 30.230245 0.128874
[17,] 479.2334 143.098241 1649 183 23.913630  6.116630 22.910543 0.248635
[18,] 186.5930 146.693122 2079 199 27.280908  6.757808 25.724818 0.195286
[19,] 356.7303 148.253418 3145 285 34.746206 11.297632 31.639921 0.313513
[20,] 449.2436 147.798319  119  37  5.873578  1.563250  6.154582 0.243697
...
```

imageHTS



Segmentation + feature extraction

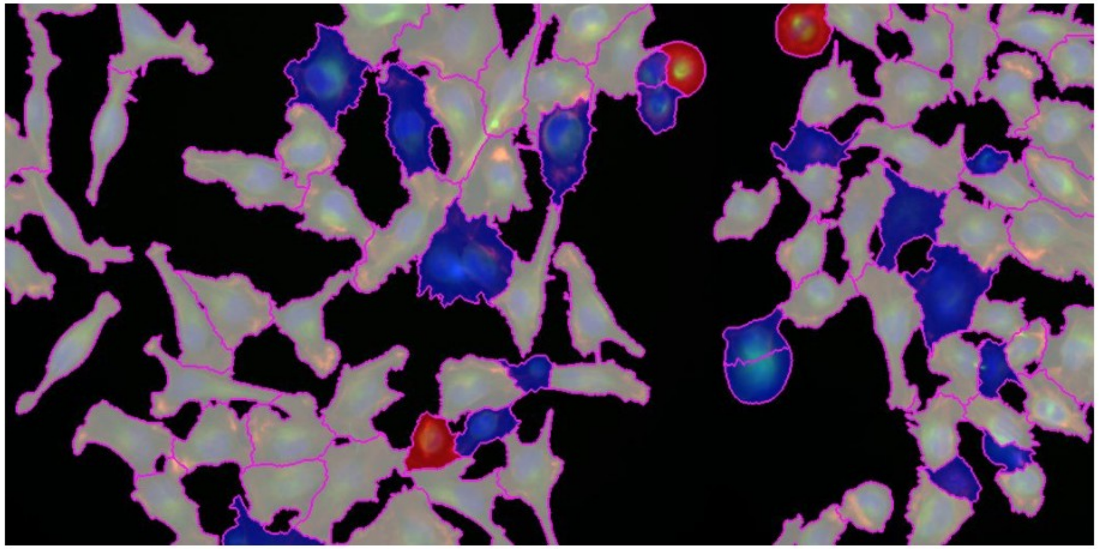
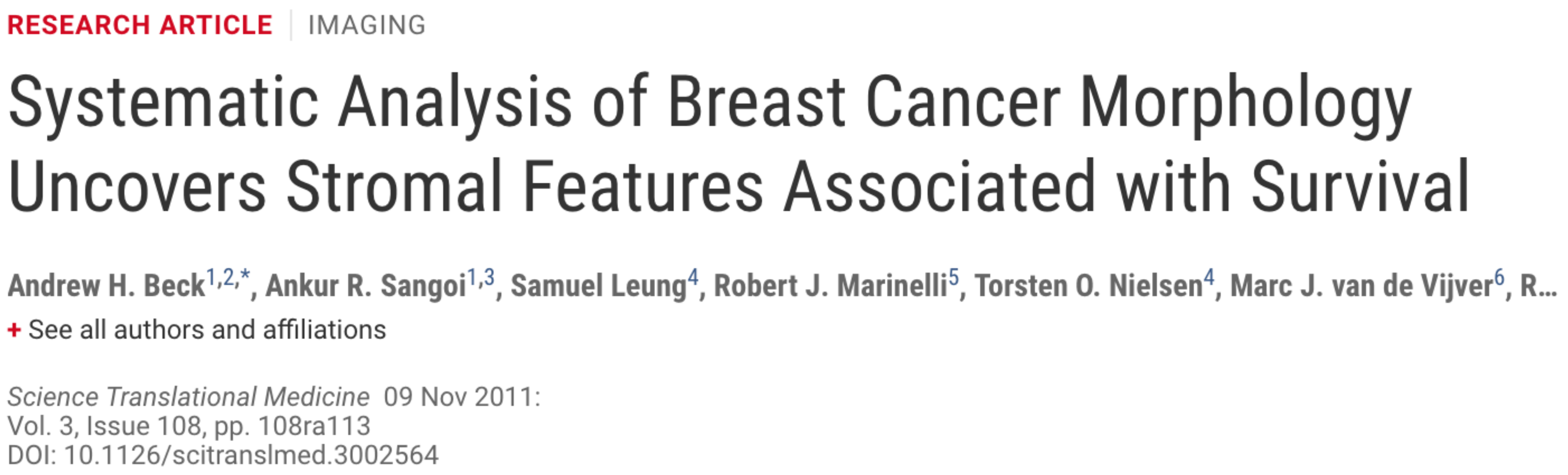


Figure 5: Predicted cell labels (grey: interphase, red: mitotic, blue: debris) in well '001-02-C03'

Can do some supervised learning
Example : SVM with radial kernel

Example in breast cancer C-Path



RESEARCH ARTICLE | IMAGING

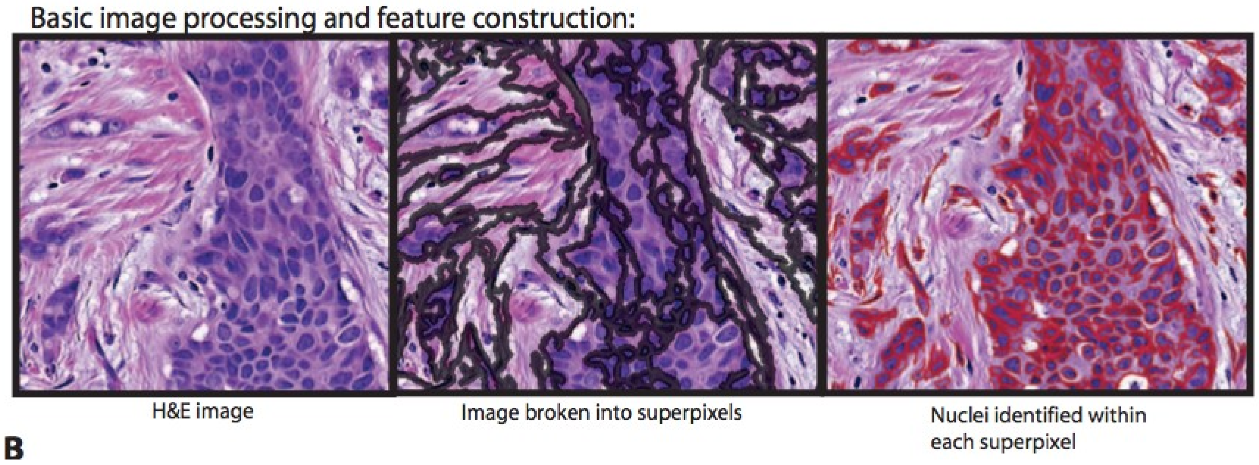
Systematic Analysis of Breast Cancer Morphology Uncovers Stromal Features Associated with Survival

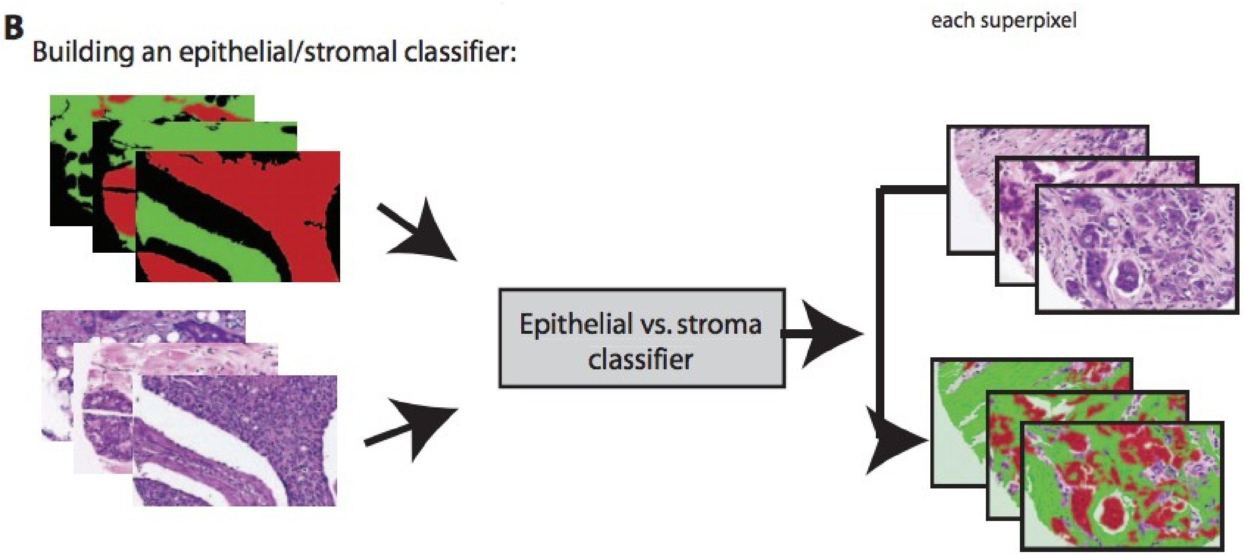
Andrew H. Beck[1,2,*], Ankur R. Sangoi[1,3], Samuel Leung[4], Robert J. Marinelli[5], Torsten O. Nielsen[4], Marc J. van de Vijver[6], R...

+ See all authors and affiliations

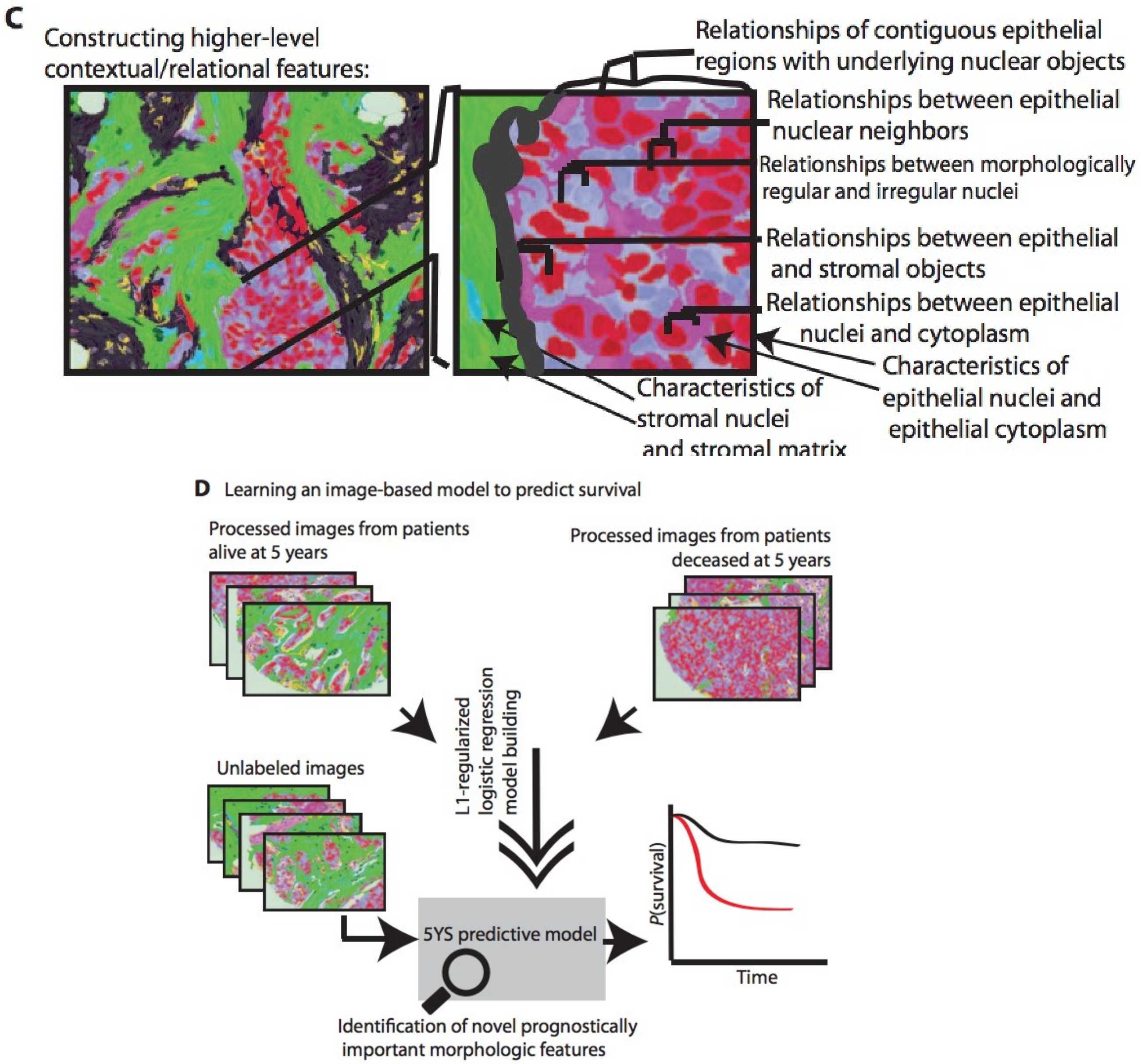
Science Translational Medicine 09 Nov 2011:
Vol. 3, Issue 108, pp. 108ra113
DOI: 10.1126/scitranslmed.3002564

C-path

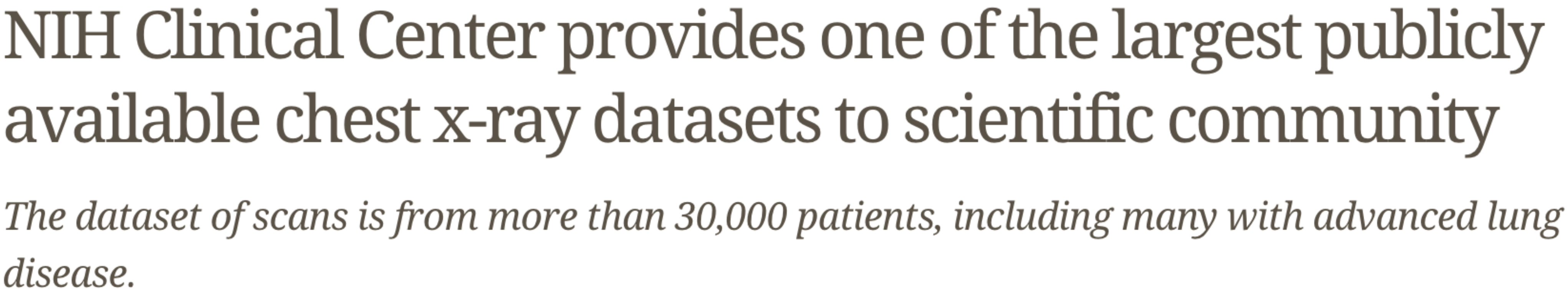

Basic image processing and feature construction:
H&E image
Image broken into superpixels
Nuclei identified within each superpixel
B


B
Building an epithelial/stromal classifier:
each superpixel
Epithelial vs. stroma classifier

C-path

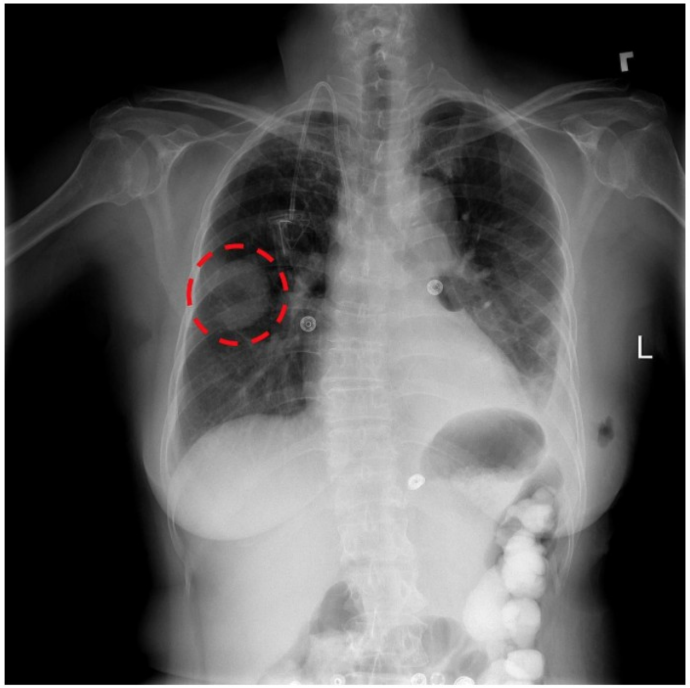

C
Constructing higher-level contextual/relational features:
Relationships of contiguous epithelial regions with underlying nuclear objects
Relationships between epithelial nuclear neighbors
Relationships between morphologically regular and irregular nuclei
Relationships between epithelial and stromal objects
Relationships between epithelial nuclei and cytoplasm
Characteristics of epithelial nuclei and epithelial cytoplasm
Characteristics of stromal nuclei and stromal matrix
D Learning an image-based model to predict survival
Processed images from patients alive at 5 years
Processed images from patients deceased at 5 years
L1-regularized logistic regression model building
Unlabeled images
5YS predictive model
Identification of novel prognostically important morphologic features
P(survival)
Time

Deep learning (Chest X-ray)



NIH Clinical Center provides one of the largest publicly available chest x-ray datasets to scientific community

The dataset of scans is from more than 30,000 patients, including many with advanced lung disease.



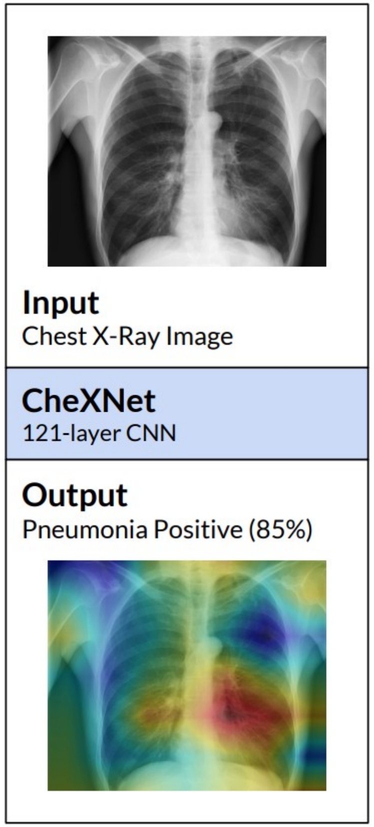
A chest x-ray identifies a lung mass.

https://nihcc.app.box.com/v/ChestXray-NIHCC

Dataset published in September 2017

CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning

Pranav Rajpurkar*[1] Jeremy Irvin*[1] Kaylie Zhu[1] Brandon Yang[1] Hershel Mehta[1]
Tony Duan[1] Daisy Ding[1] Aarti Bagul[1] Curtis Langlotz[2] Katie Shpanskaya[2]
Matthew P. Lungren[2] Andrew Y. Ng[1]



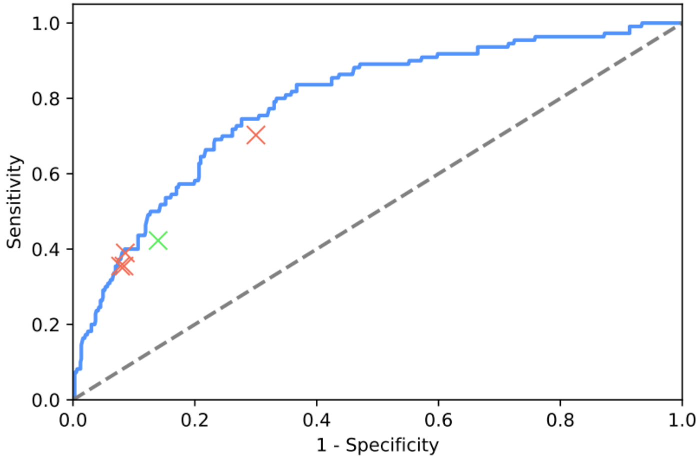
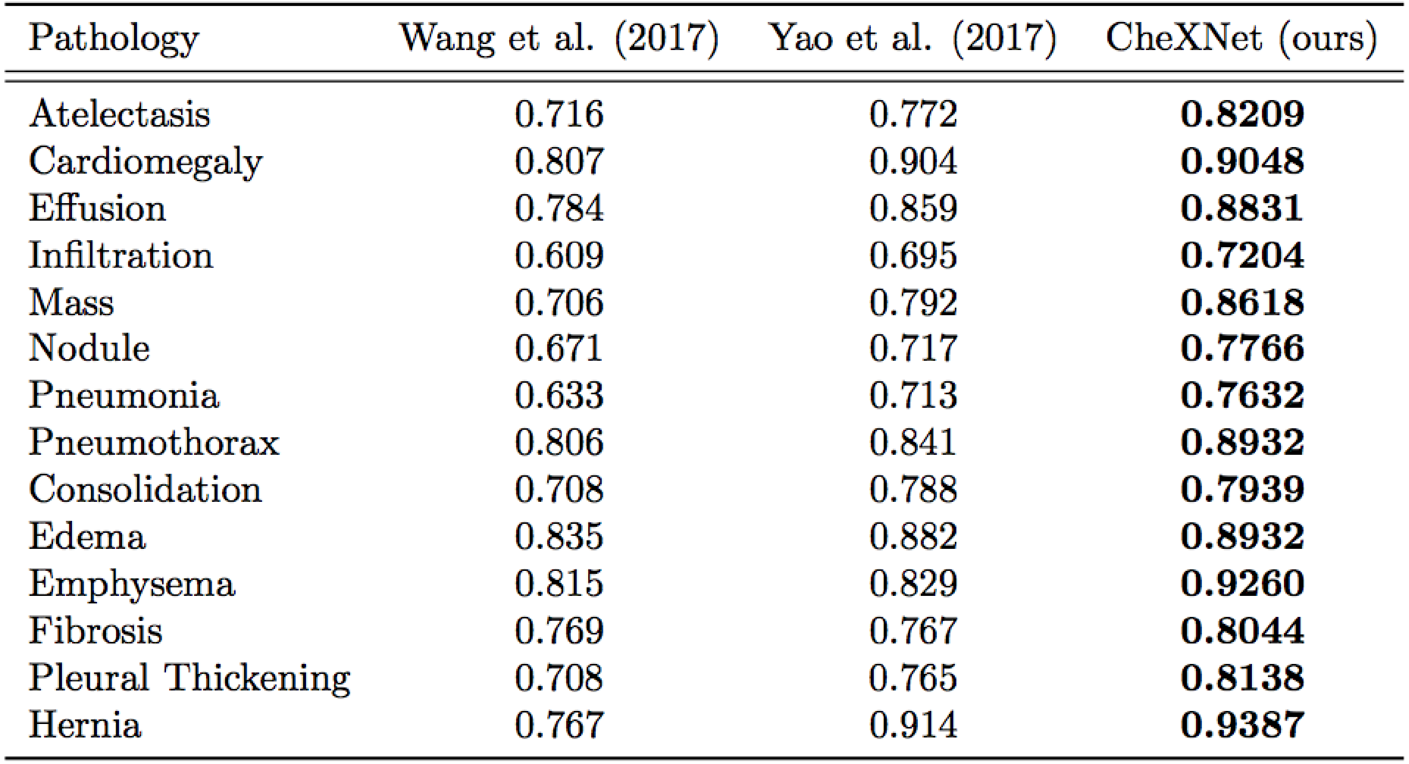
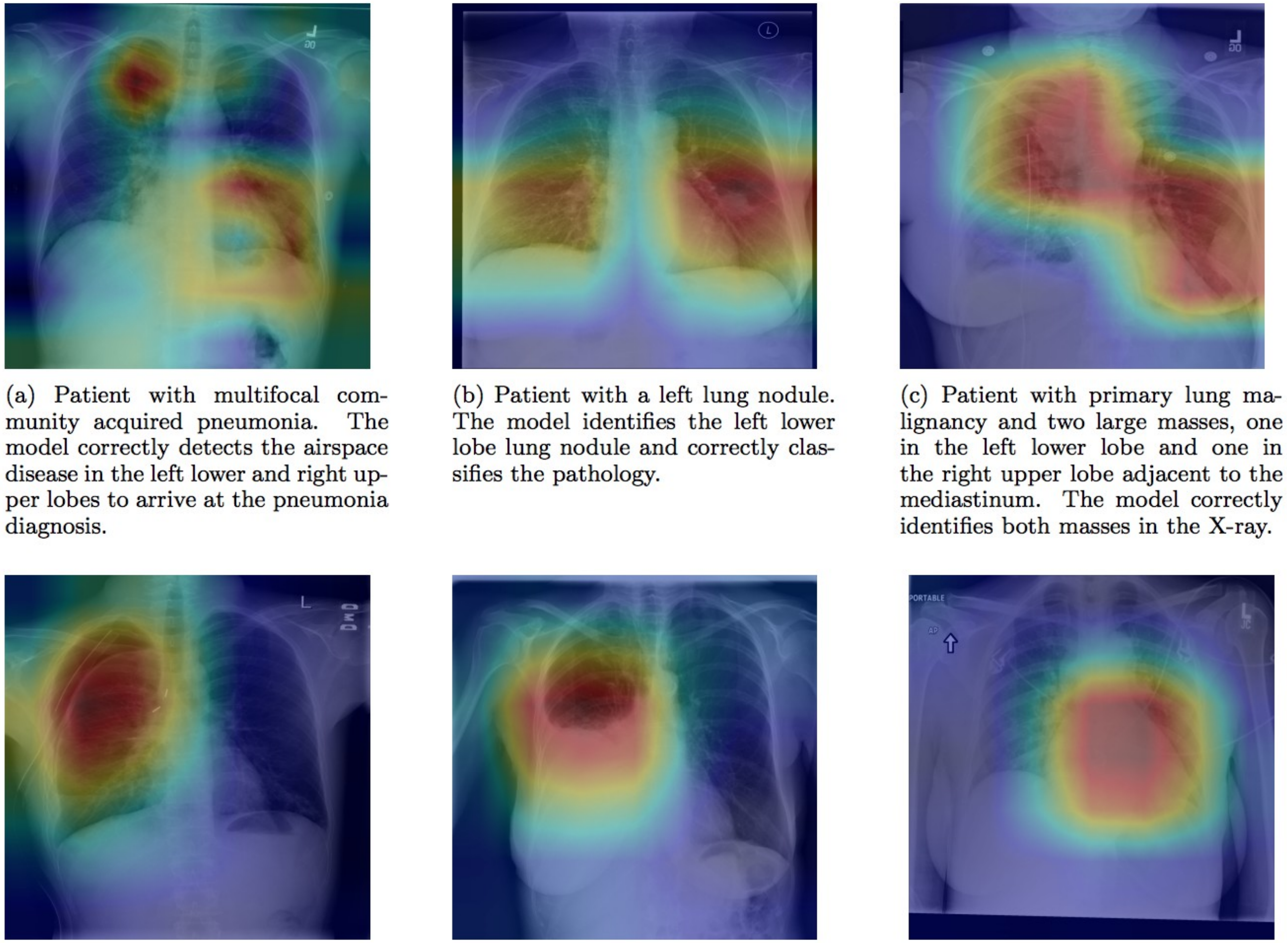


Figure 2. CheXNet outperforms the average of the radiologists at pneuomonia detection using X-ray images. ChexNet



Pathology	Wang et al. (2017)	Yao et al. (2017)	CheXNet (ours)
Atelectasis	0.716	0.772	**0.8209**
Cardiomegaly	0.807	0.904	**0.9048**
Effusion	0.784	0.859	**0.8831**
Infiltration	0.609	0.695	**0.7204**
Mass	0.706	0.792	**0.8618**
Nodule	0.671	0.717	**0.7766**
Pneumonia	0.633	0.713	**0.7632**
Pneumothorax	0.806	0.841	**0.8932**
Consolidation	0.708	0.788	**0.7939**
Edema	0.835	0.882	**0.8932**
Emphysema	0.815	0.829	**0.9260**
Fibrosis	0.769	0.767	**0.8044**
Pleural Thickening	0.708	0.765	**0.8138**
Hernia	0.767	0.914	**0.9387**

Table 1. CheXNet outperforms the best published results on all 14 pathologies in the ChestX-ray14 dataset. In detecting Mass, Nodule, Pneumonia, Pneumothorax, and Emphysema, CheXNet has a margin of >0.05 AUROC over previous state of the art results.



(a) Patient with multifocal community acquired pneumonia. The model correctly detects the airspace disease in the left lower and right upper lobes to arrive at the pneumonia diagnosis.

(b) Patient with a left lung nodule. The model identifies the left lower lobe lung nodule and correctly classifies the pathology.

(c) Patient with primary lung malignancy and two large masses, one in the left lower lobe and one in the right upper lobe adjacent to the mediastinum. The model correctly identifies both masses in the X-ray.

(d) Patient with a right-sided pneumothroax and chest tube. The model detects the abnormal lung to correctly predict the presence of pneumothorax (collapsed lung).

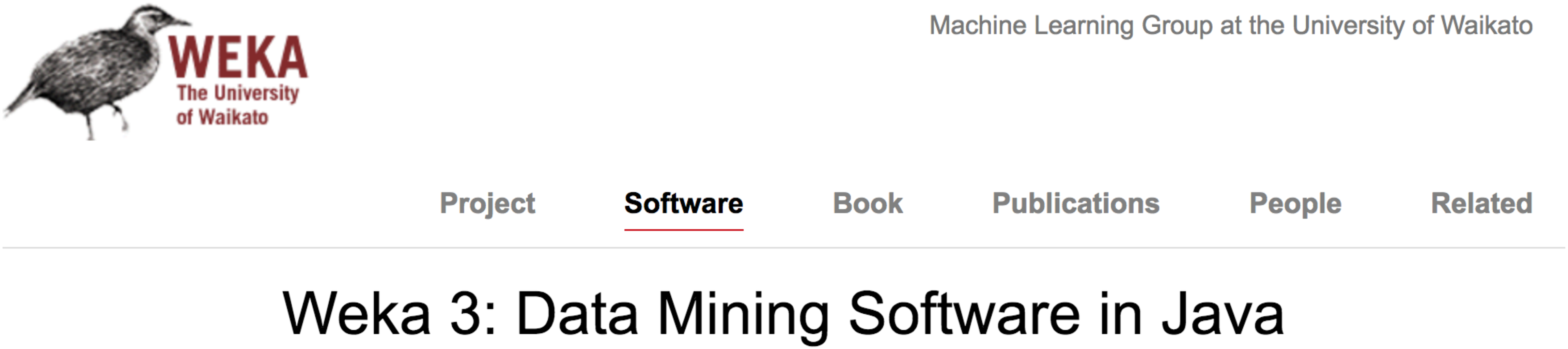
(e) Patient with a large right pleural effusion (fluid in the pleural space). The model correctly labels the effusion and focuses on the right lower chest.

(f) Patient with congestive heart failure and cardiomegaly (enlarged heart). The model correctly identifies the enlarged cardiac silhouette.

Figure 3. ChexNet localizes pathologies it identifies using Class Activation Maps, which highlight the areas of the X-ray that are most important for making a particular pathology classification.

GUI machine learning

- WEKA



Machine Learning Group at the University of Waikato

Weka 3: Data Mining Software in Java

Good technical book online

- The elements of statistical learning. Hastie, Tibshirani, and Friedman
  - https://web.stanford.edu/~hastie/Papers/ESLII.pdf
- Pattern recognition and machine learning. Christopher Bishop
  - http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20-%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%202006.pdf

The end