

Statistical learning and virtual screening in drug discovery

Hugh Chipman, Acadia University

Joint work with Wanhua Su, Marcia Wang, Sunny Wang, Will Welch, Stan Young, Yan Yuan, Mu Zhu.

Outline:

1. Introduction to the drug discovery problem
2. “Off-the shelf” supervised learning methods
3. Modifications and fancy tricks
4. Interlude: Is a better learner what we really want?
5. Sequential Design.

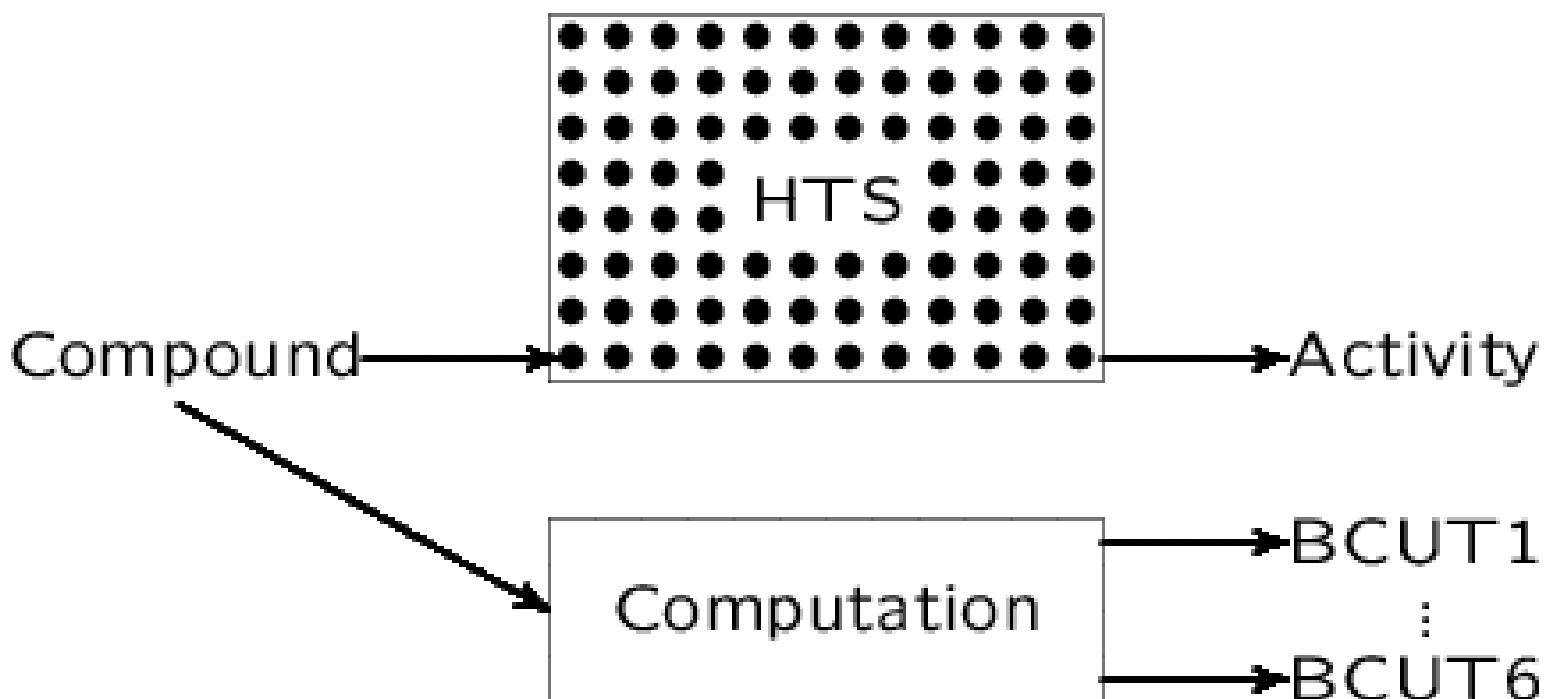
1. Introduction to the drug discovery problem:

High throughput screening (HTS)

Response variable: Activity

(% inhibition, IC50 concentration, inactive/active)

Explanatory variables: Molecular descriptors (e.g., BCUT's)



Statistical challenges

1. Classification/Regression modelling:

- Given a training set with activity and descriptors, construct a model to predict activity using the descriptors.
- This allows “virtual screening” of compounds - identify most likely actives without testing them all.

2. Experimental design

- How do you select a training set (1000's of compounds) from a library of millions of compounds?
- Sequential design.

A “standard” classification/regression problem

- Use the molecular descriptors to predict activity.
- But there are a number of challenges:
 - Extreme imbalance of active/inactive compounds.
 - Quantifying a compound’s structure is difficult.
 - Some descriptor sets are highly correlated.
 - Screening leads to strong clustering in descriptor space.
 - Highly nonlinear and local effects.

Example datasets - from National Cancer Institute (NCI)

1. AIDS Antiviral Data (Inactive/Active)

- Response: 0/1 inactive/active (active = highly active or mildly active)

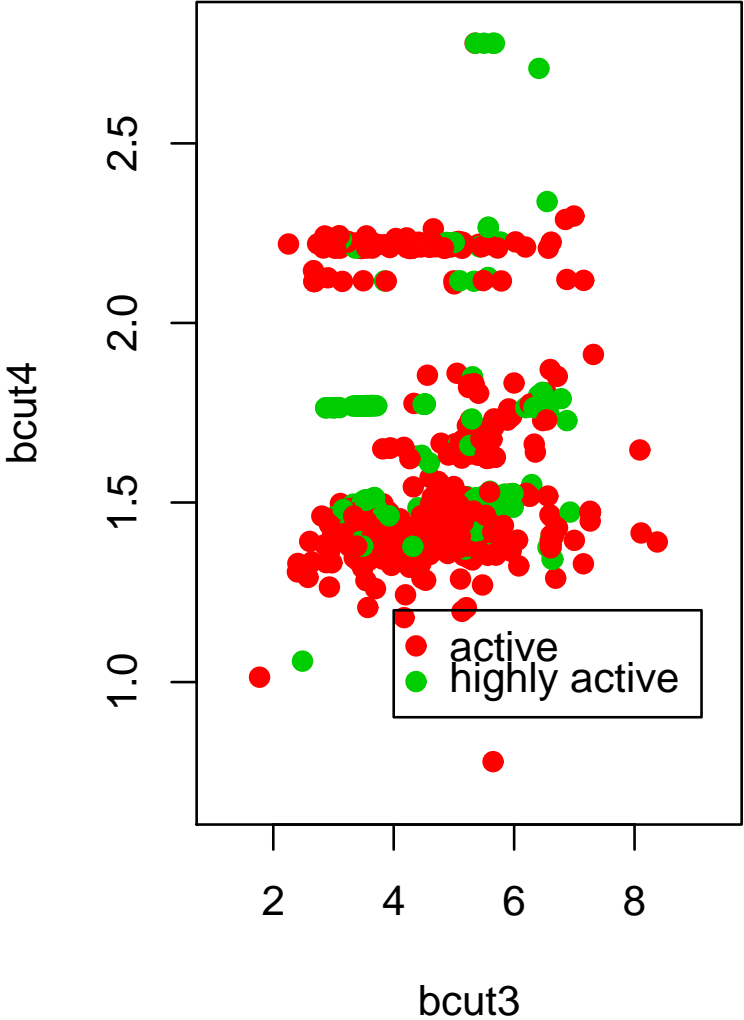
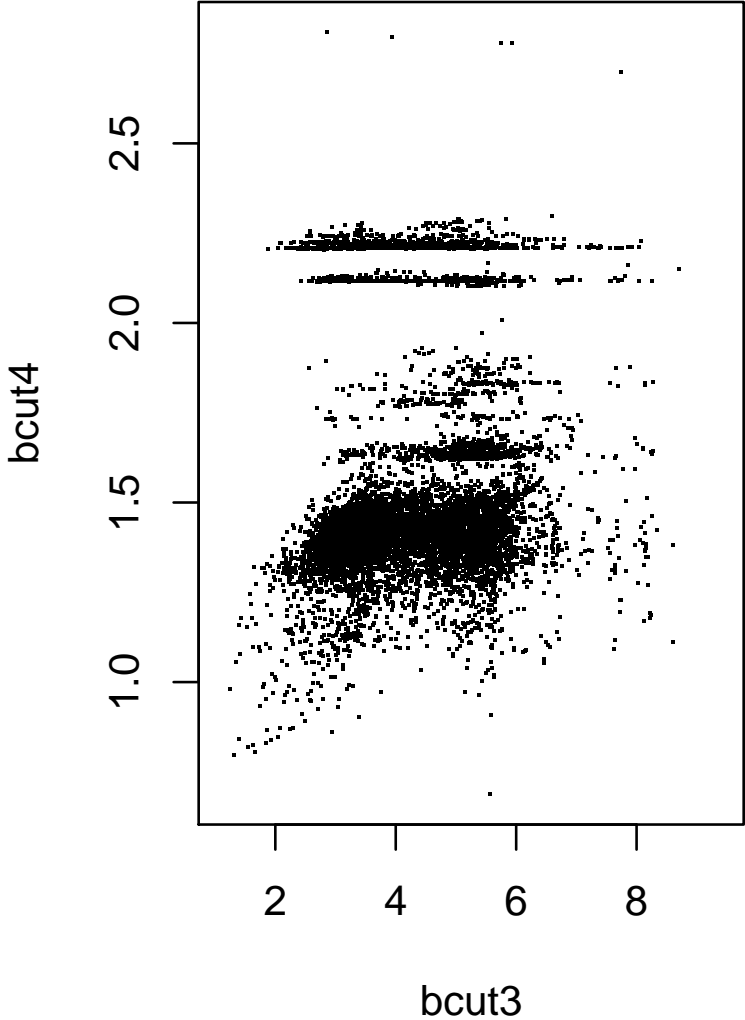
	Training data	Test data	Total
Active	304	304	608
Inactive	14,602	14,602	29,204
Total	14,906	14,906	29,812

- 6 BCUT descriptors

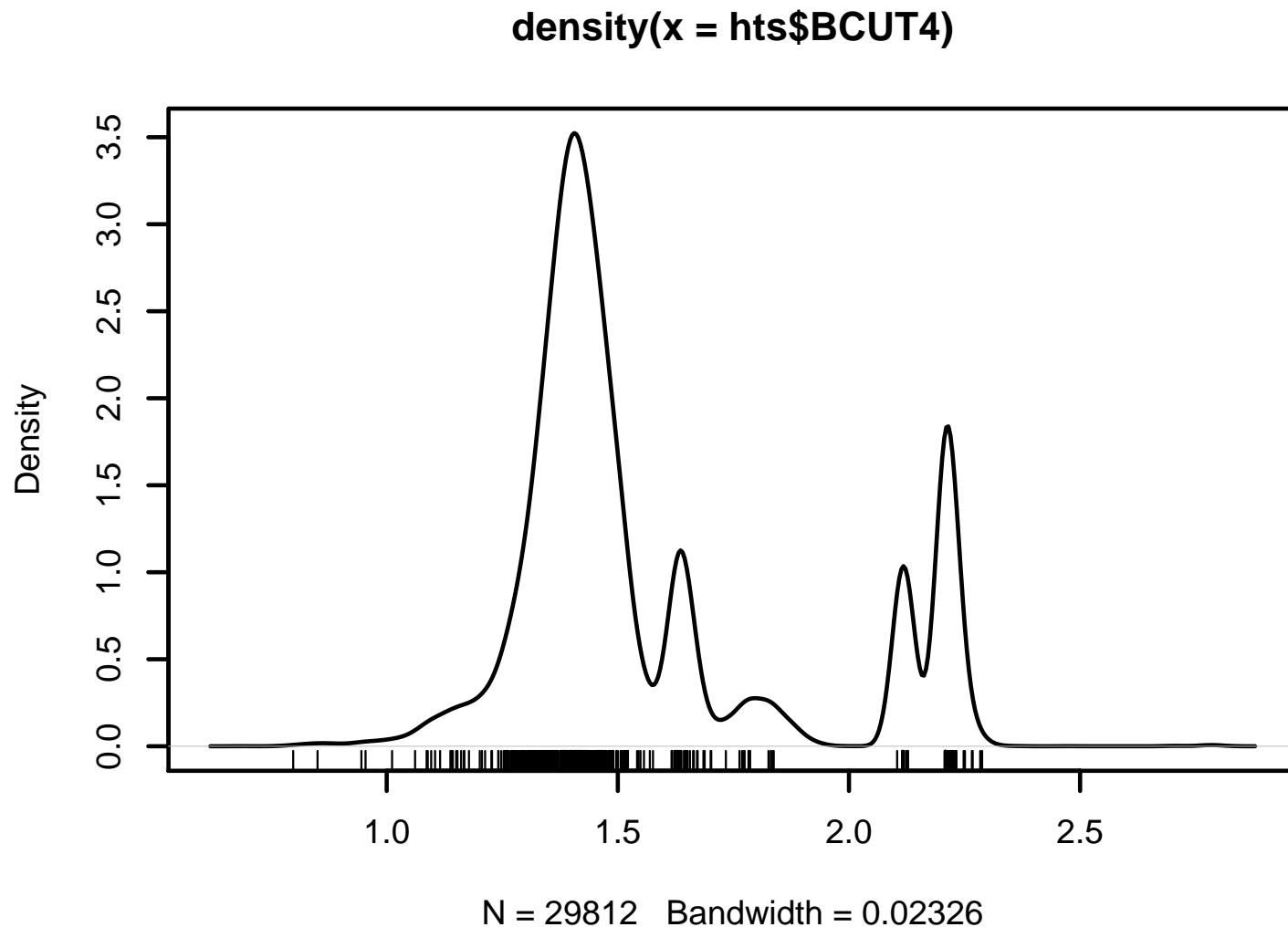
2. A similar dataset with a continuous response exists: $-\log(\text{EC50})$.

EC50 = compound concentration that protects infected cells by 50% ($-\log(\text{EC50})$ is a larger-the-better response).

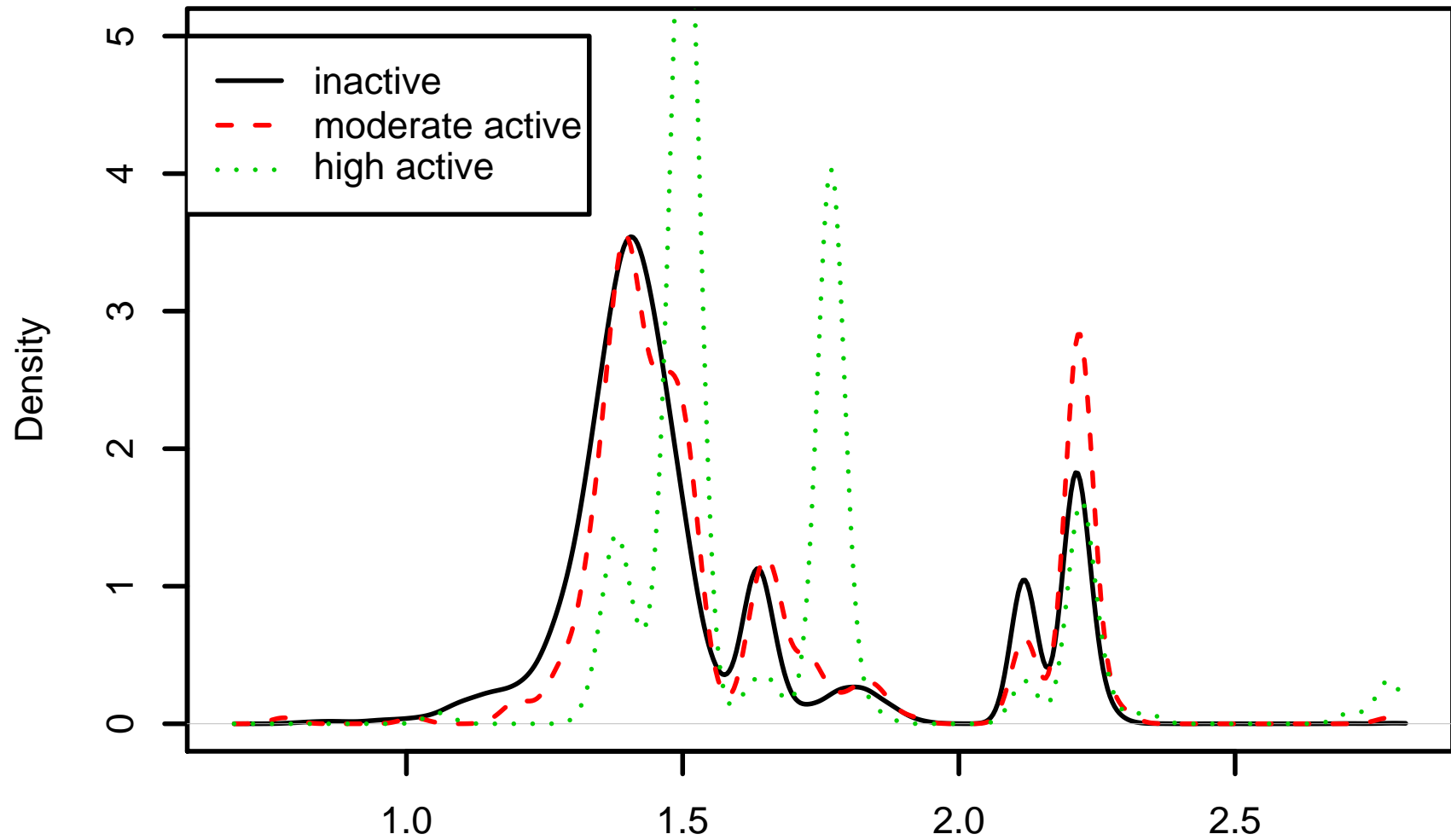
Some views of the data



Another view: one dimensional probability density.



density of BCUT4 by activity



N = 29204 Bandwidth = 0.0233

2. “Off the shelf” statistical learning methods

Comparative study of various models, including

- Logistic regression.
- Generalized additive models (nonlinear logistic).
- Neural networks (nonlinear logistic with interactions).
- Multivariate adaptive regression splines (MARS).
- Trees (CART, C4.5).
- K-nearest neighbours.

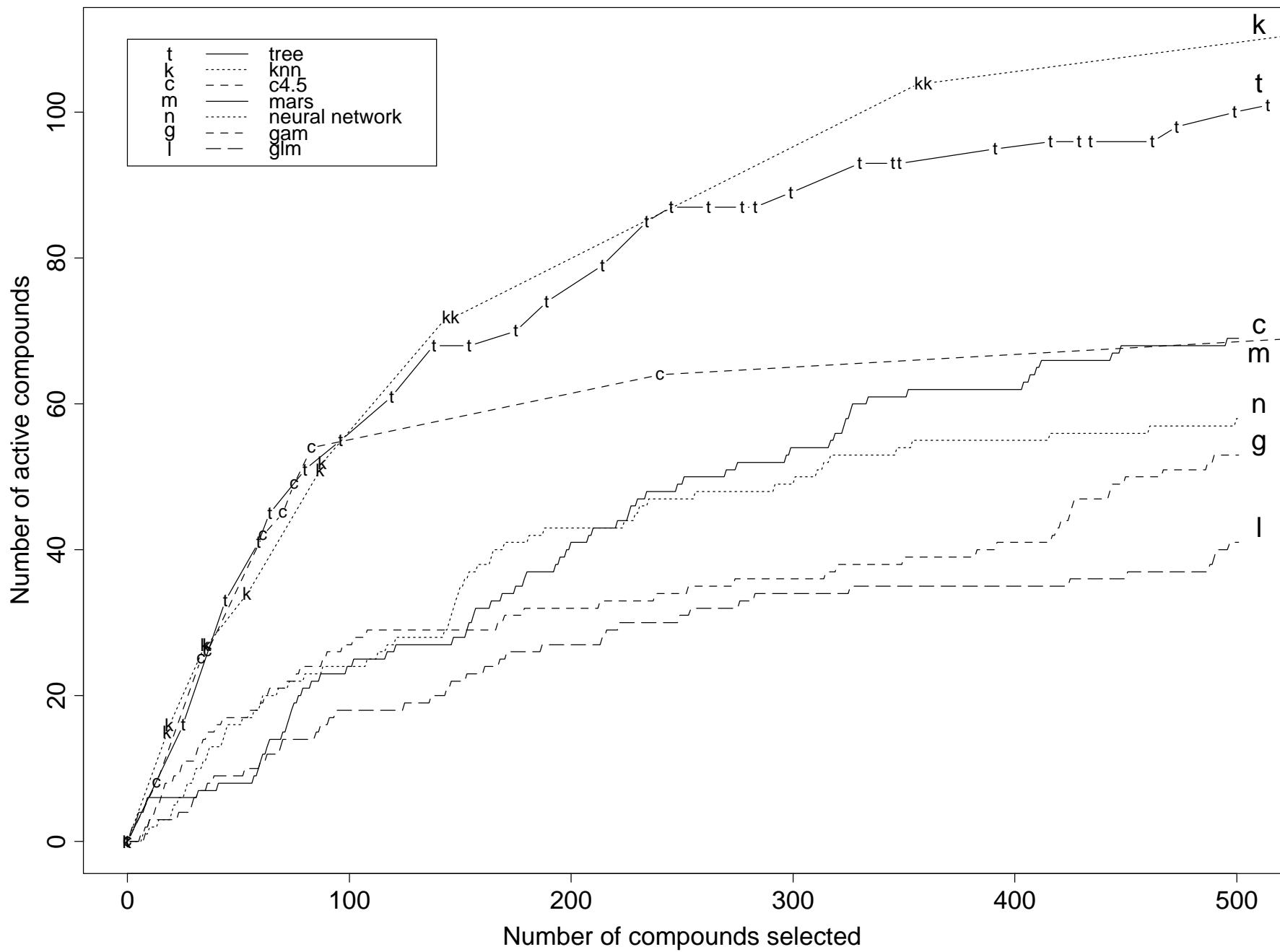
Comparing methods via a “Hit curve”

Main idea: measure performance in a way relevant to the drug discovery problem.

For each model:

1. Rank compounds in test set.
2. Select compounds most likely to be active (one at a time).
3. As you select compounds, count the number of active compounds found.

Sound bite summary: Think “Google search”!

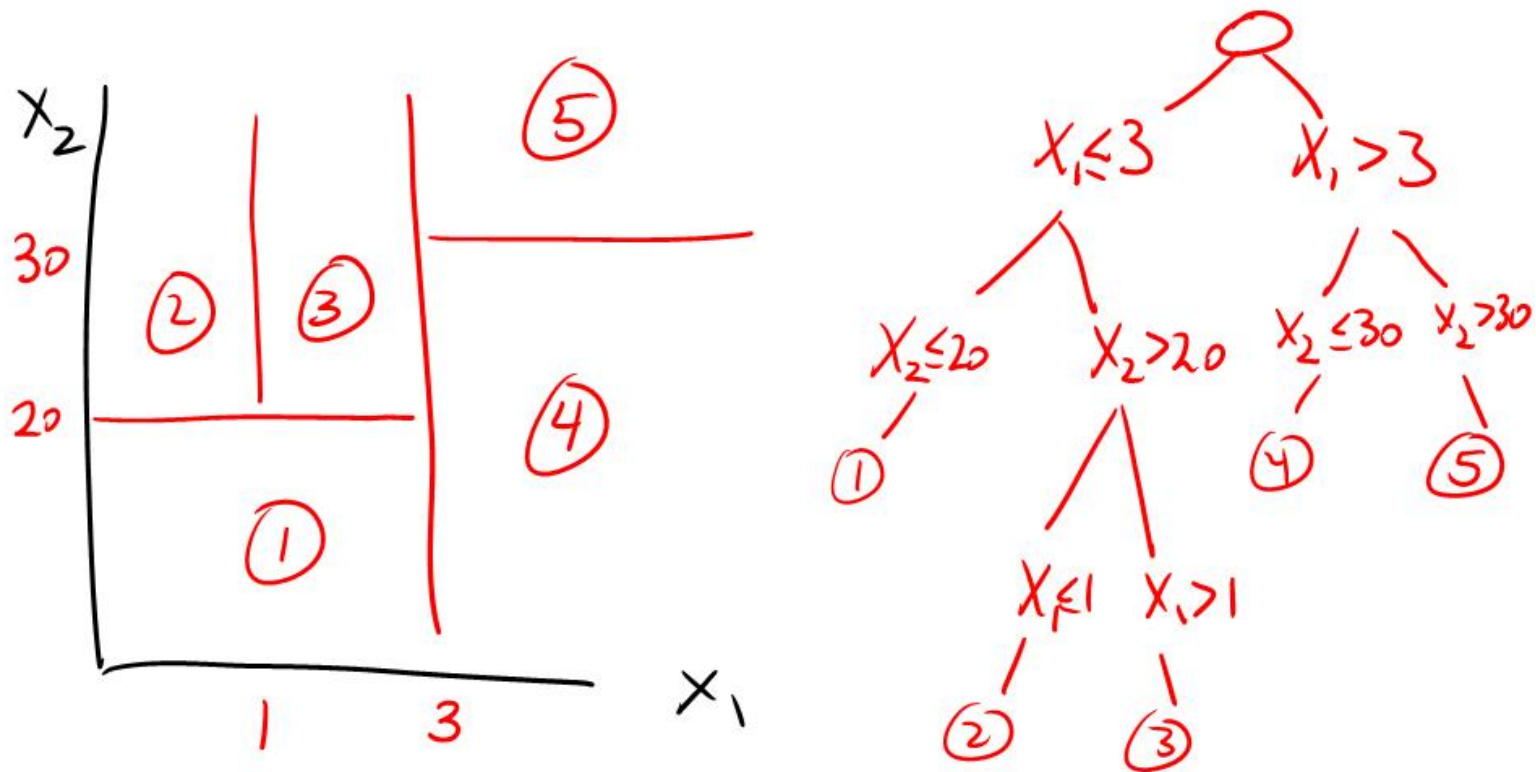


Findings from the comparisons:

- Local methods (trees, knn) seem to give best performance.
- Assuming linearity or even continuity seems to hurt performance.
- Although k-nearest neighbours performs well, it is less stable. If a different train/test division is used, it can do poorly or quite well.

3. Modifications and fancy tricks

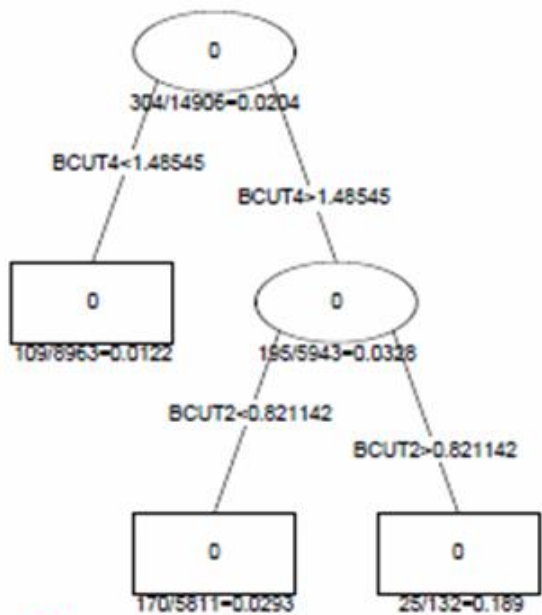
Tree Models



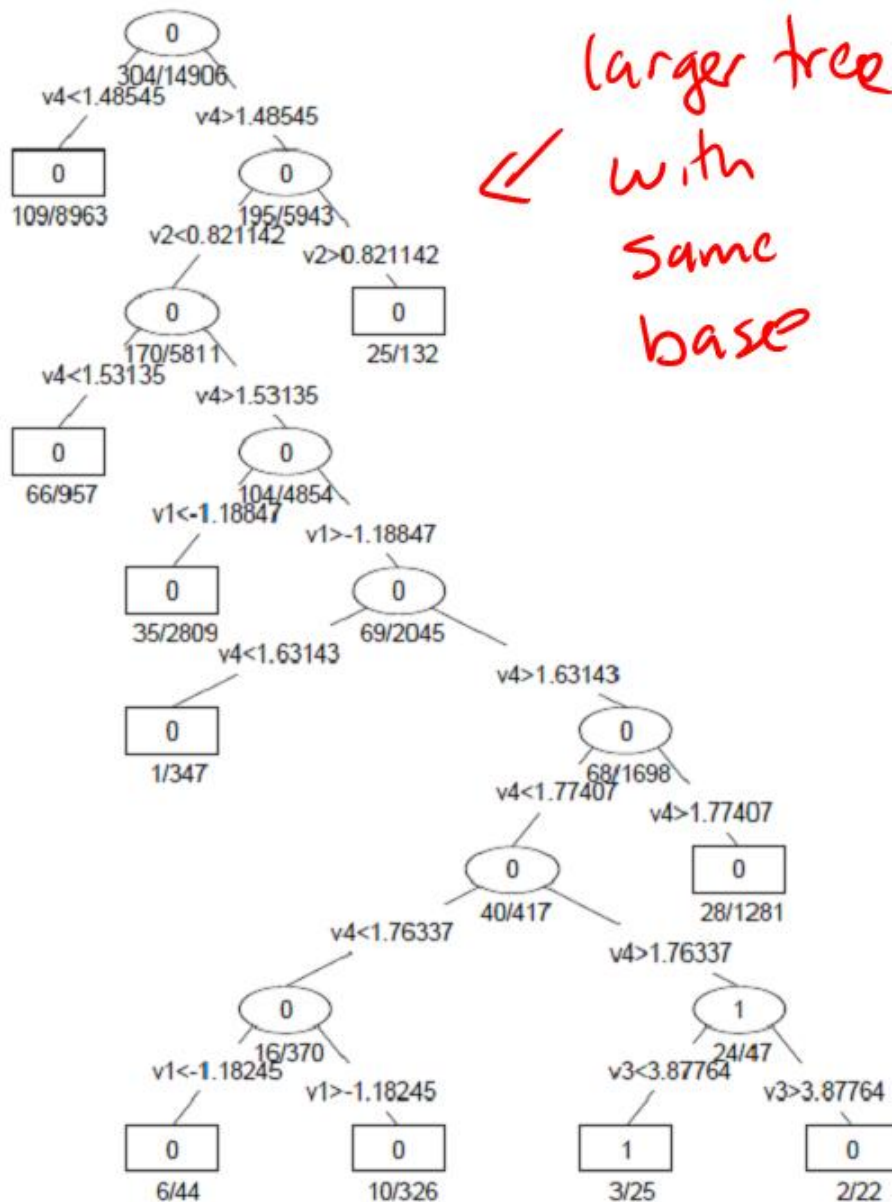
Tree structure is estimated from data (“learned”) so that response variable (eg activity) is as homogeneous as possible within each terminal node.

Improvements on trees

- Goal is to rank observations from best to worst.
- A problem with trees is that every observation in a node gets the same prediction.
- \Rightarrow When we go to make predictions for a test set, there will be many ties.
- Next page: Big trees are better than small ones, because they have fewer ties.



↑ small tree



← larger tree with same base

That is,

- More nodes \Rightarrow fewer identical predictions.
- When we test observations one at a time, we can selective include observations in “the good nodes”

Another problem with trees:

So bigger trees are good, but they lead to a problem: deciding what are the “best” nodes.

Example:

- Node 1 has 10 compounds, 3 active
- Node 2 has 100 compounds, 25 active

Which node is best?

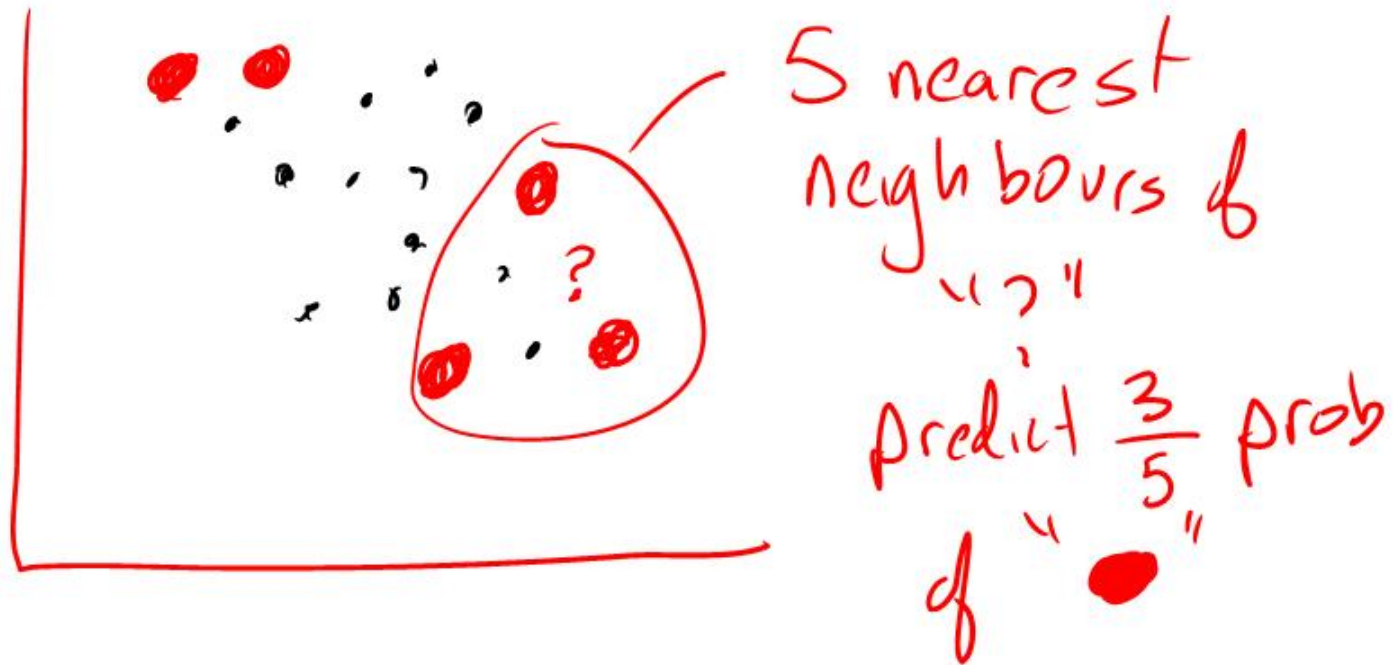
I'd choose node 2, because I'm more confident that it represents a “high yield” group.

Formalize this - calculate a 95% confidence interval for $p = P(\text{active})$, choose the nodes that have the highest lower bound.

Result: better ranking of groups of observations when group size varies, also better “tie breaking” (eg select a 10/10 group before a 1/1 group).

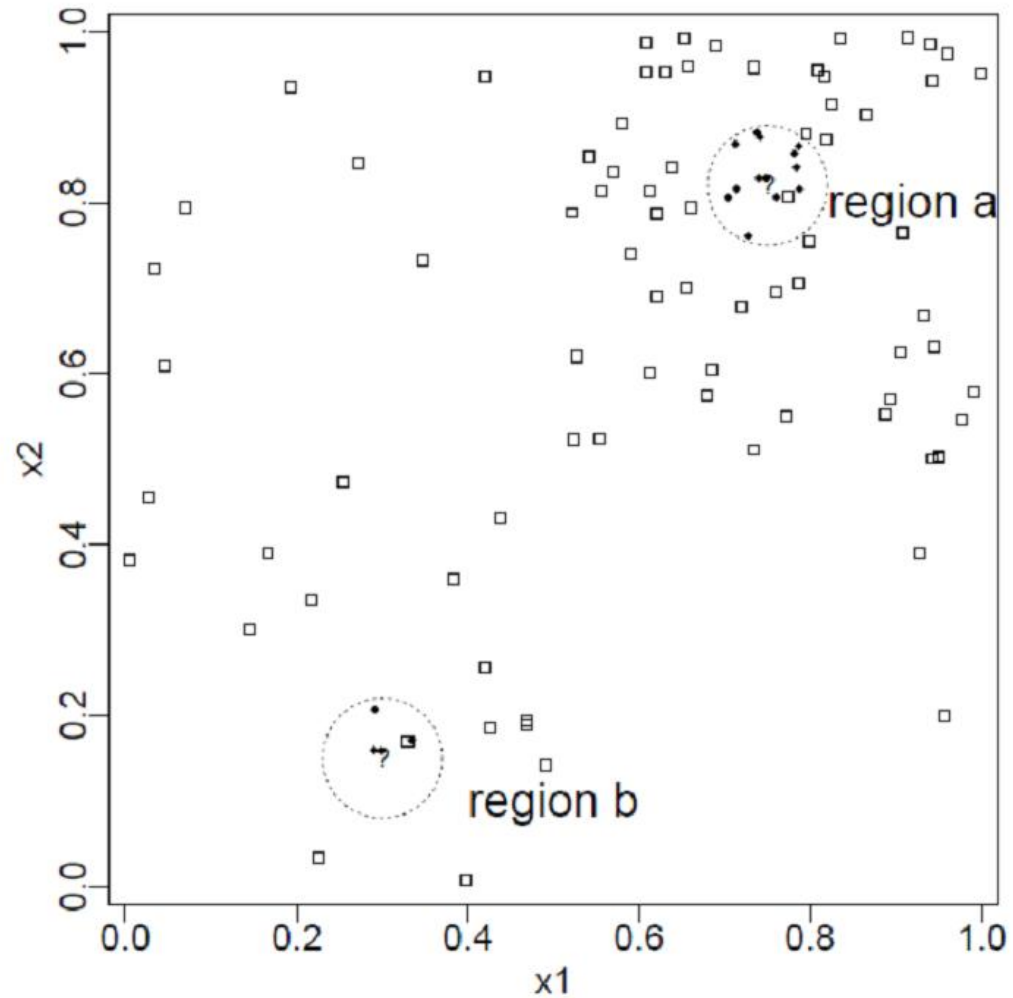
K-nearest neighbours and modifications

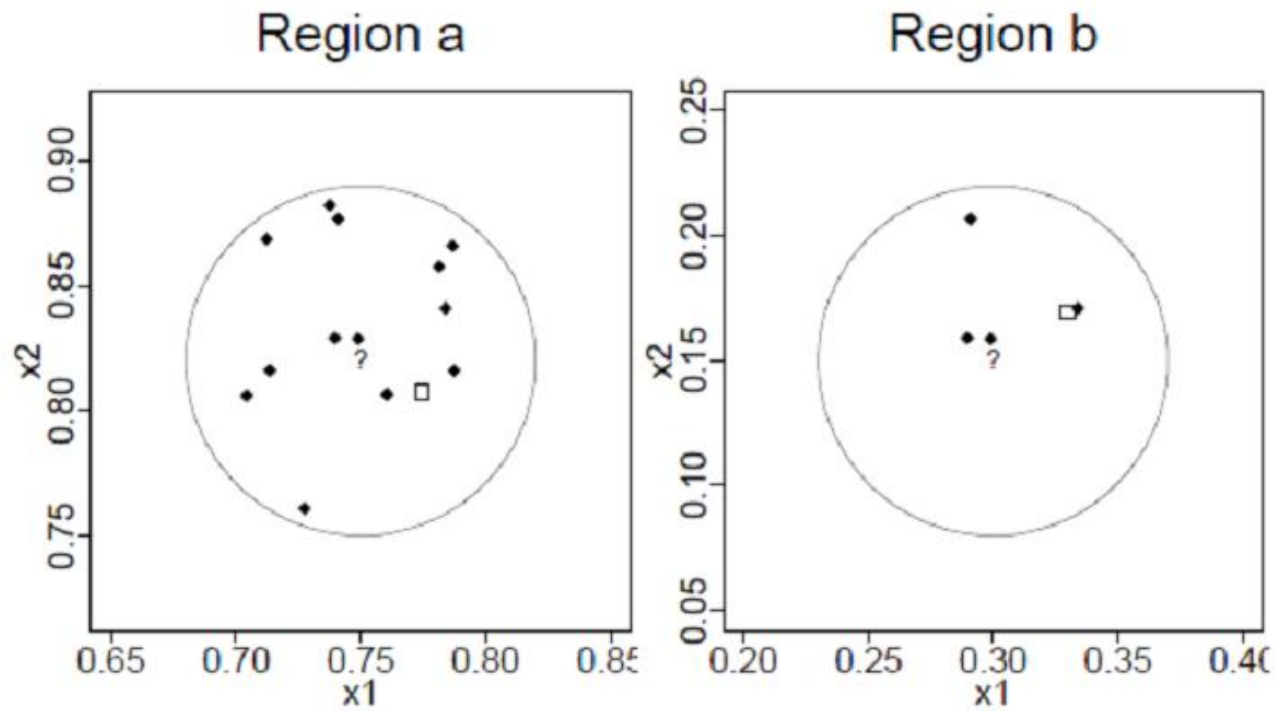
Basic idea: To predict at a point (labelled "?" below) find the k nearest points, and use them to estimate the probability of activity



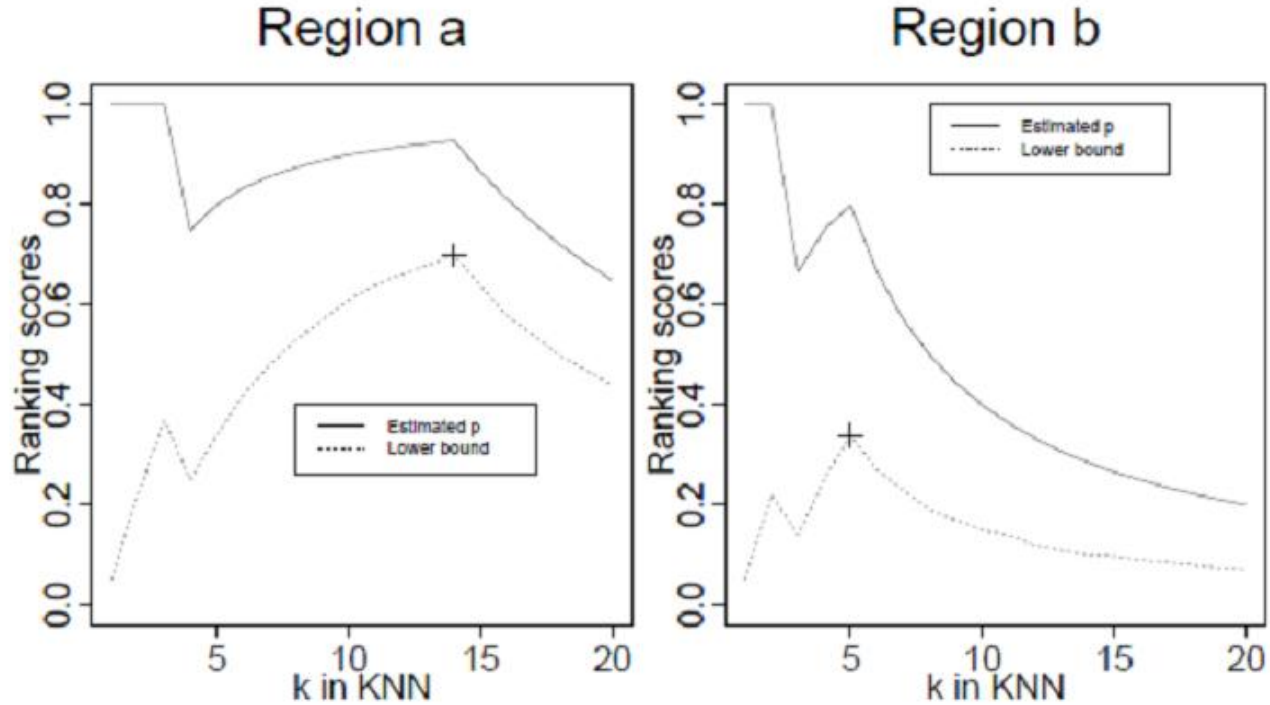
Changing “k”?

To the right,
we see two
regions where
different k
might be best





Choose different value of "k" depending on how many actives and inactives nearby. In region "a" use a larger k than in "b"



- How do we choose k? Basically calculate the lower confidence limit for “p” at each k, and choose the k that gives the largest lower bound.
- This trades off “large sample for accuracy” against “local sample avoiding bias”

Other methods:

- Locally Adjusted GO estimator (LAGO) - Zhu, Su, Chipman (2006)
 - Sort of like kernel density estimation, but with adaptive kernels *only* on the active compounds
- Subspace averaging - Wang, Welch, Chipman (in progress)
 - Run KNN in many subspaces, average the predictions.
 - “Ensemble” idea, seems effective for ranking.
- Cluster Structure Activity Relation Analysis (CSARA - Wang, Salloum, Young, Welch, Chipman, 2007)
 - Cluster data into many groups, select “best” groups after assaying one compound from each group.
 - Experimental design and nonparametric modelling combined.
- Tree Harvesting (Yuan, Welch, Chipman, in progress)
 - Extract and simplify the best sequences of rules from a tree.

Mixture discriminant analysis:

So far, all the models considered have focused on directly predicting response Y given inputs (descriptors) X .

An alternative for categorical Y is a “generative model”:

- Model X conditional on each class (i.e., estimate $p(x|y)$).
- Use Bayes’ rule to predict Y given X

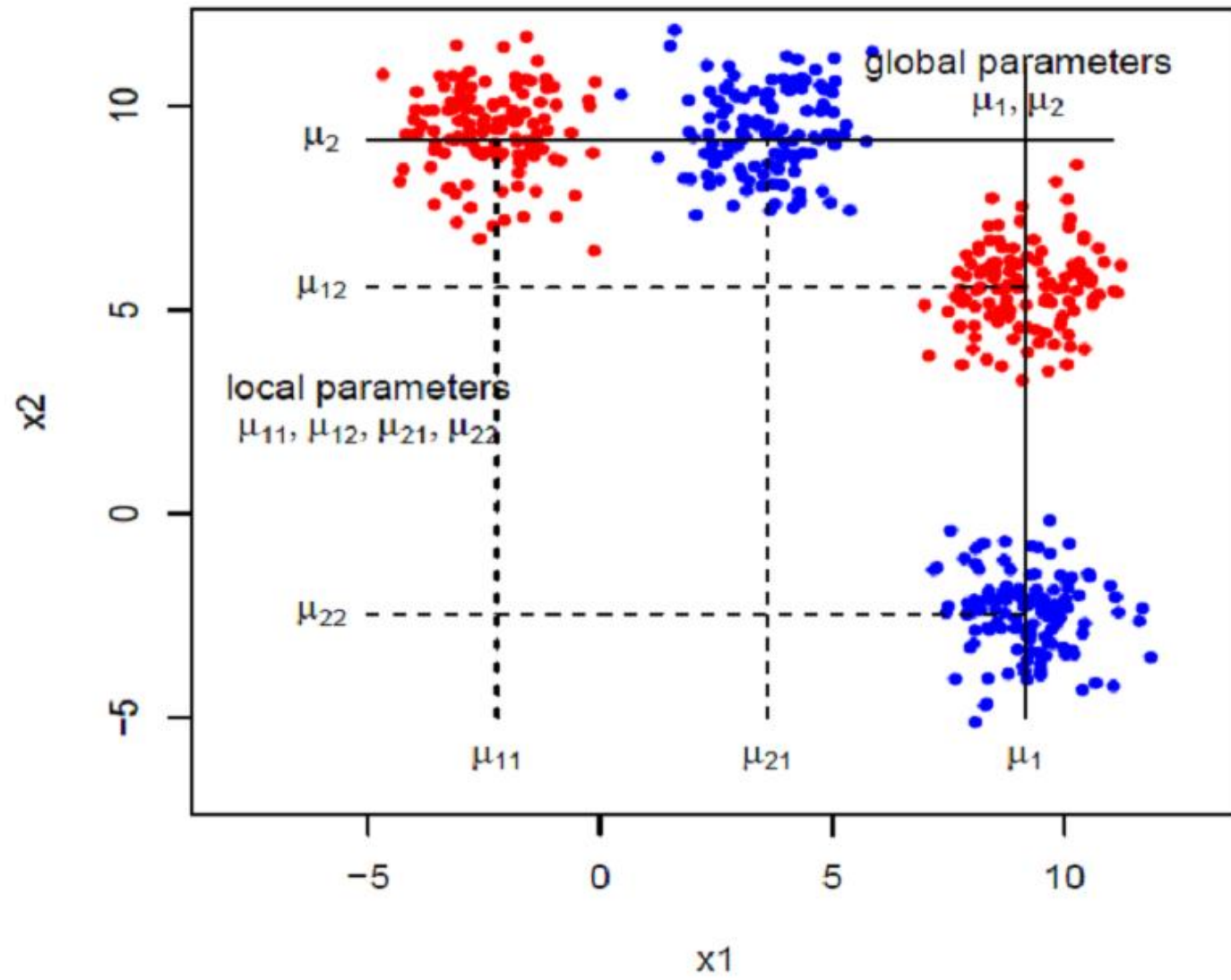
$$P(Y = 1|x) = \frac{P(x|Y = 1)p(Y = 1)}{p(x)}$$

- Linear and quadratic discriminant analysis are examples of this: model $X|Y$ as multivariate normal.

Mixture discriminant analysis:

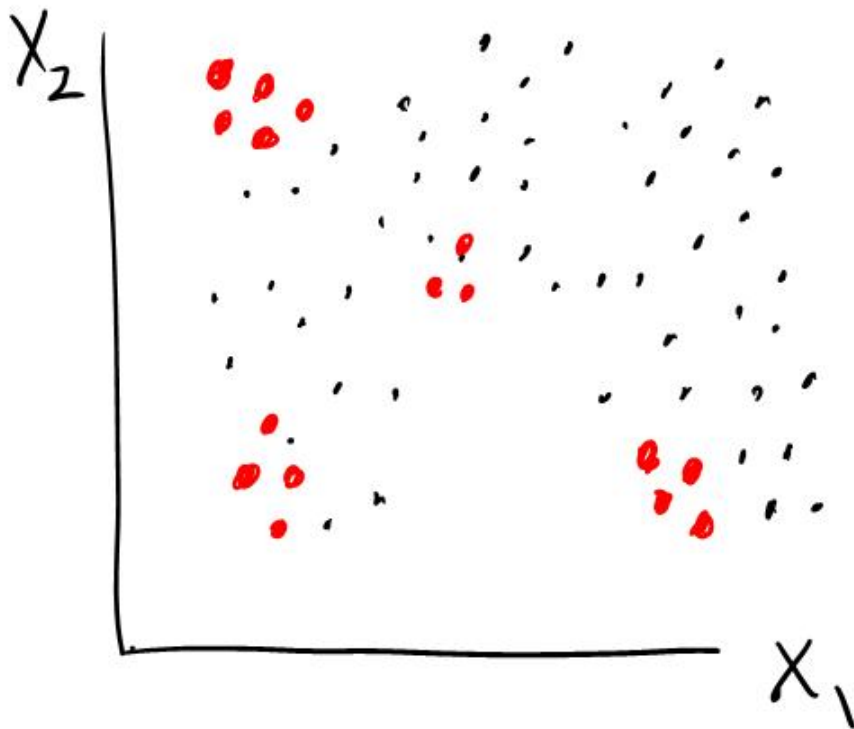
Generalization here:

- Replace multivariate normal within each class by a mixture of multivariate normals (subpopulations within each class).
- Since data may be high-dimensional (many descriptors), constrain the mixture components to lie on different low-dimensional subspaces.
- Since the active class is very rare, we want to avoid a large number of parameters in the model
 - So choose a model in which most parameters are common across different classes, and only a few parameters differ.



4. Interlude: Is a better learner what we really want?

- inactive
- active

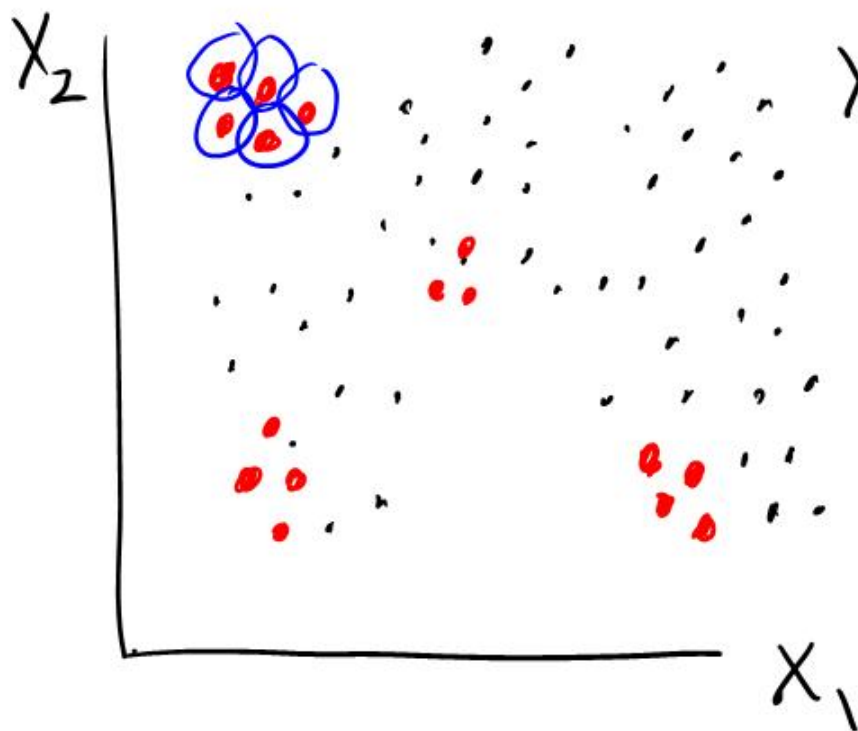


Finding the active compounds:

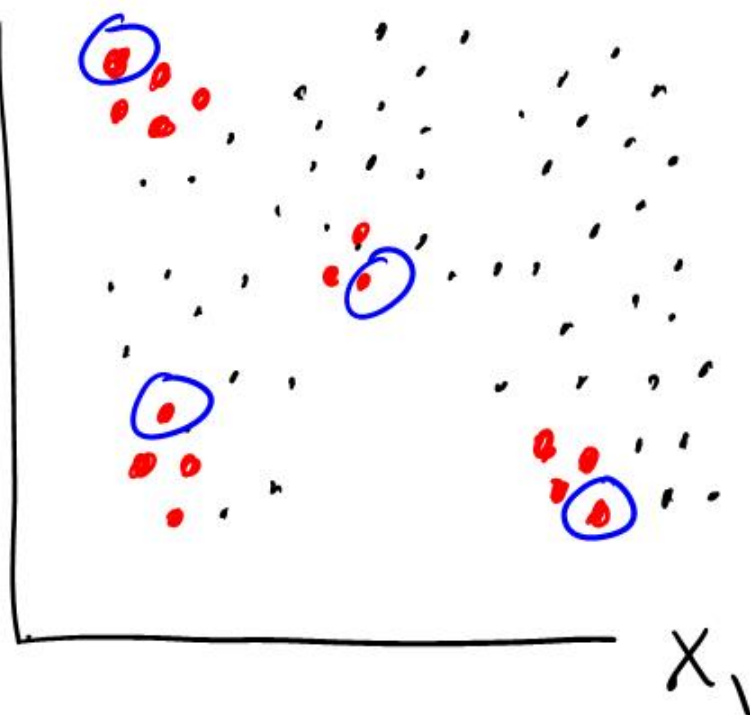
Is more always better?

Finding a diverse set of compounds can be more important than finding many compounds (Right hand side better than left)

found $\frac{5}{16}$ actives



found $\frac{4}{16}$ actives



Some important issues this raises:

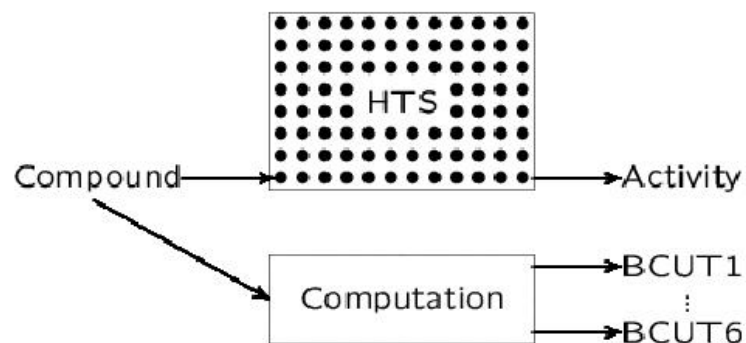
- Design is at least as important as analysis (you can't hope to predict where the actives are unless you have data that gives you clues).
- For any kind of local modelling, we want to “fill the descriptor space”
- Sequential approaches are likely to be better.
- How do we quantify the tradeoff between “diversity” and “number of hits”?

5. Sequential Design

Sequential Design

The game we play:

- Standard “regression” modelling: predict compound activity (Y) using descriptors (X) of its molecular structure
- Descriptors X can be calculated without lab work - using only a description of the molecule’s structure.
- Activity Y can only be measured with an assay (lab work).



- We have a “library” of between 1,000 and 1,000,000 compounds with known descriptors (known X 's).
- Known descriptors \Rightarrow candidate design points are fixed points – experimental design consists of “choosing the best points”

Sequential Design

Sketch of the sequential design algorithm:

1. Select an initial design (i.e., initial set of compounds) X_0 with n_0 points via some criterion.
2. Assay those compounds (obtaining corresponding Y_0)
3. Build a model using data $D_0 = (X_0, Y_0)$.
4. Repeat $j = 1, \dots, n$:
 - (a) For each potential design point $x_i \in$ candidate set C , calculate the design criterion (**details... next page**)
 - (b) Select point x_{i^*} with best design criterion yielding design $X_j = (X_{j-1}, x_{i^*})^T$.
 - (c) Assay this point, yielding $Y_j = (Y_{j-1}, y_{i^*})^T$
 - (d) Build model M_j using $D_j = (X_j, Y_j)$.

Note:

- At the end of this algorithm, we will have $n_0 + n$ observations.

Sequential Design

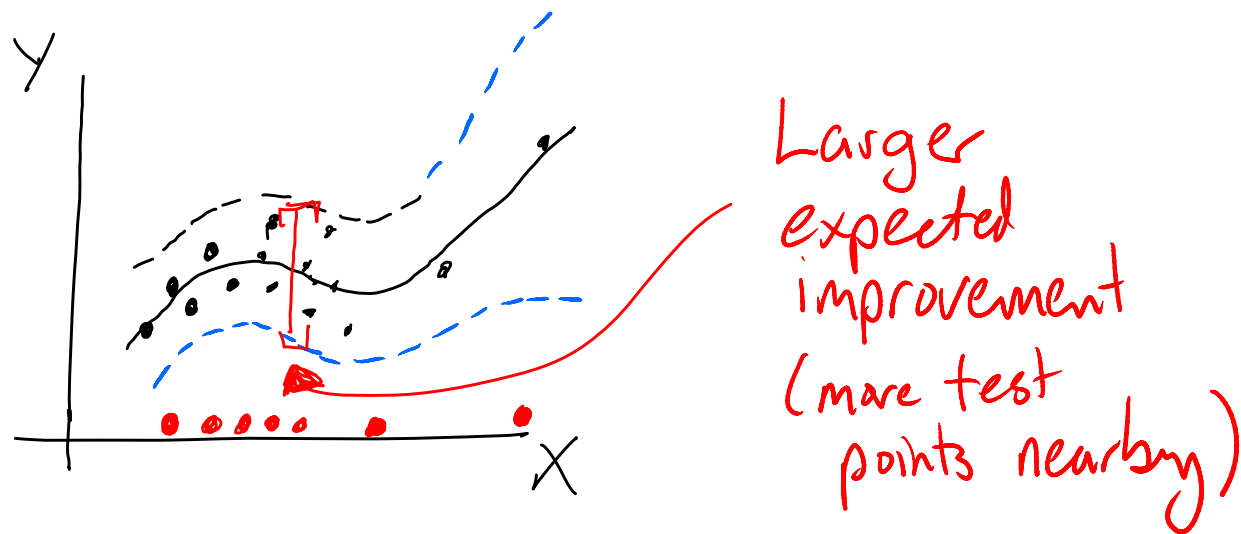
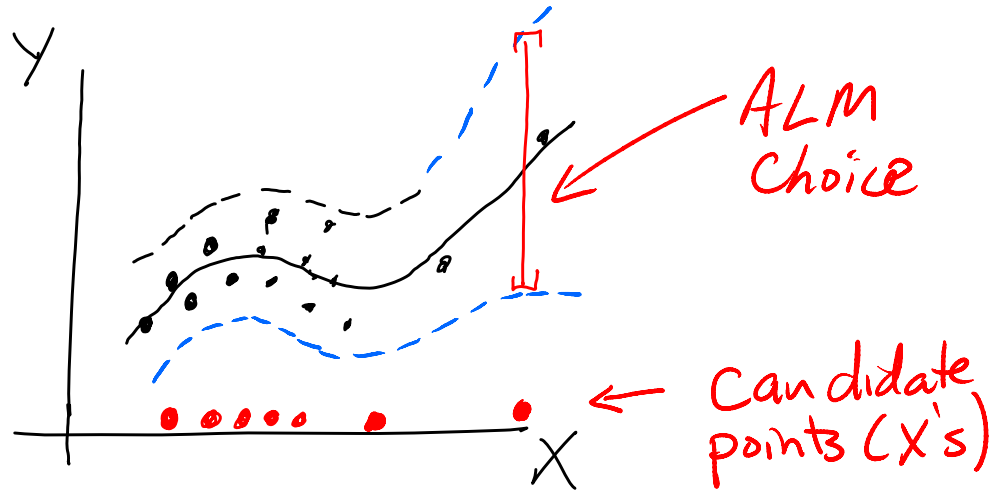
Three possible design criteria:

1. Maximize variance of predicted response
(*where do I know the least about Y ? - MacKay (1992)*).
2. Maximize expected reduction in variance of predicted response, averaged over a candidate set C
(*what data point will improve my model's predictions most? - Cohn (1996)*)
3. Hybrid - easier to calculate than #2, captures idea of expected reduction.

Modelling procedure:

- BART: Bayesian Additive Regression Trees
- Key features are the ability to
 - flexibly model y given x
 - quantify uncertainty in predictions

Maximum uncertainty vs. expected improvement



Maximum uncertainty vs. expected improvement

ALM criterion (maximum variance)

- Use BART to generate a posterior variance for predicted response $\hat{\mu}(x_i)$ for all x_i points in the candidate set.
- Add the point with the largest variance.

$$i^* = \operatorname{argmax}_i \operatorname{Var}(\hat{\mu}(x_i)) \quad i \in \text{Candidate set}$$

Hybrid criterion:

- “weight” each candidate point according to how many other candidate points it is “close” to.
- “closeness” is difficult to define for arbitrary predictor variables, so...
- Instead use correlation as a measure of closeness.

$$i^* = \operatorname{argmax}_i w_i \operatorname{Var}(\hat{\mu}(x_i))$$

where $w_i = \sum_{j=1}^n \operatorname{cor}(\hat{\mu}(x_i), \hat{\mu}(x_j))$

Example: AIDS Antiviral data

- 29,812 compounds
- Binary response is **Activity** (1=active, 0=inactive)
- Only 608/29,812 compounds are active.
- Goal: find the active compounds.
- Predictors are six “Burden numbers” (BCUTS) meant to characterize molecular structure.

Example: AIDS data

- For illustrative purposes, we'll divide up data as follows:
 - 1000 observations for training.
 - 3000 observations as candidate points to be sequentially selected.
 - * 500 of these 3000 will be chosen by each method.
 - 2000 observations as a “test set” .
- Consider three sequential sampling schemes:
 1. BART + Simple Random Sampling
 2. BART + ALM (max variance)
 3. BART + correlation-weighted ALM

Performance:

How do we measure performance (on the test set)?

- Binary deviance

$$-2 \sum_i y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)$$

- Average Hit Rate (AHR)
 - Idea: measure ability of a model to rank compounds from “most active” to “least active”.

$$AHR = \frac{\sum_{i=1}^n y_{(i)} \left(\sum_{j=1}^i y_{(j)} / i \right)}{\sum_{i=1}^n y_{(i)}}$$

where $y_{(i)}$ is the observed response of the i -th ranked observation.

Performance:

- Average Hit Rate example calculation:

i	prediction $\hat{p}(i)$	response $y(i)$	# hits so far $\sum_{j=1}^i y(j)$	$y(i) \times$ hit rate $y(i) \sum_{j=1}^i y(j) / i$
1	.9	1	1	$1/1 = 1$
2	.8	0	1	$0/2 = 0$
3	.75	1	2	$2/3 = .66$
4	.3	0	2	$0/4 = 0$
5	.2	0	2	$0/5 = 0$
6	.1	0	2	$0/6 = 0$

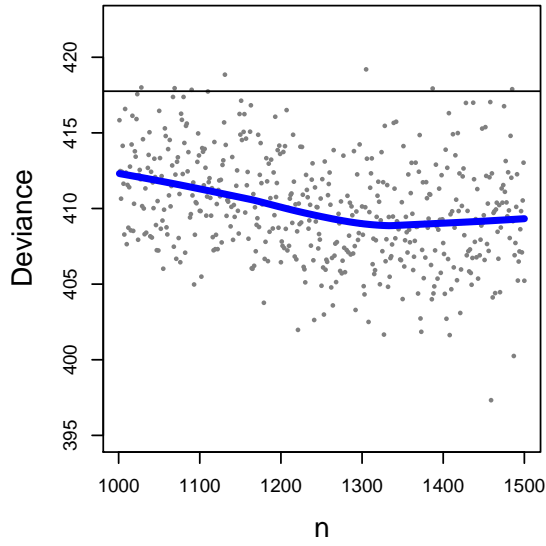
$$\text{AHR} = (1 + 0 + .66 + 0 + 0 + 0) / 2 = 0.83$$

- AHR properties:
 - $\text{AHR}=0 \Leftrightarrow$ all active compounds are ranked last.
 - $\text{AHR}=1 \Leftrightarrow$ all active compounds are ranked first.
 - AHR is a “larger the better” performance measure.

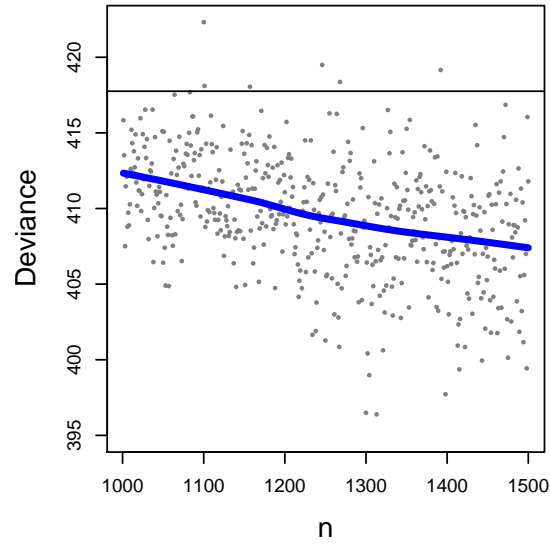
AIDS Example:

Results on next page show test criteria as points are added one-at-a-time.

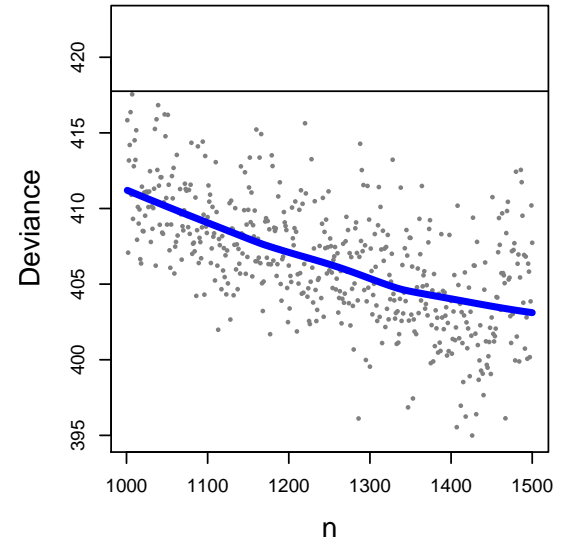
Simple random sample



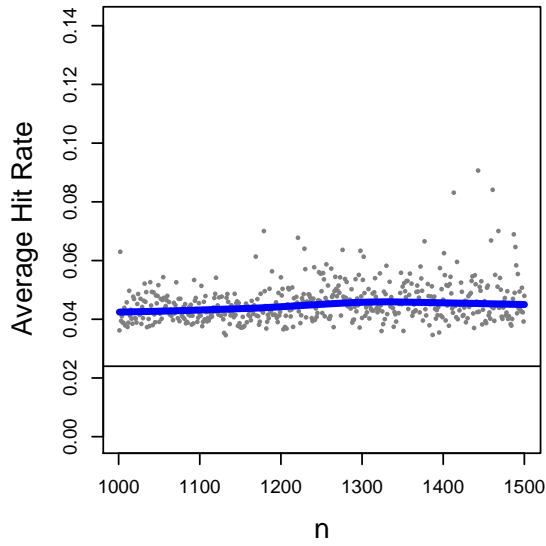
ALM



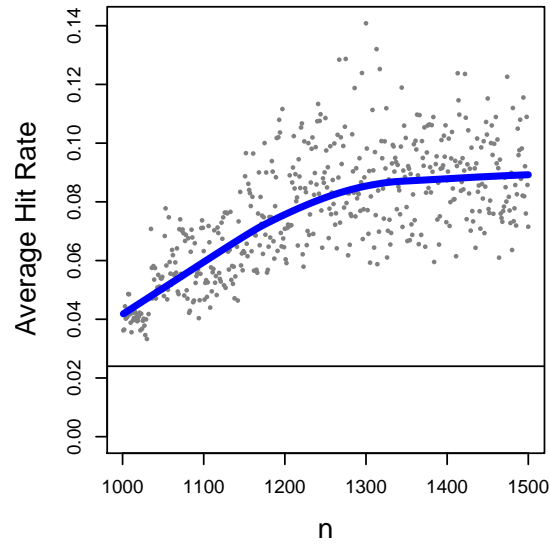
ALM * Correlation



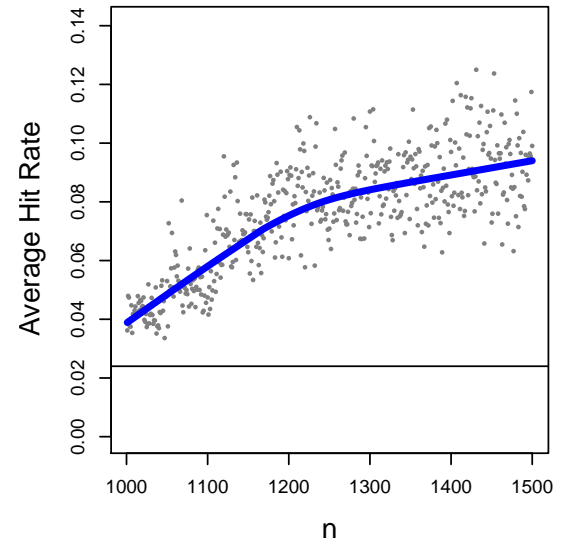
Simple random sample



ALM



ALM * Correlation



AIDS Example:

- Sequential sampling yields faster improvements in deviance and AHR than random sampling.
- Hybrid method may perform slightly better than ALM.

Conclusions:

- Drug discovery presents many interesting problems:
 - Design
 - Modelling
 - How to build descriptors
 - Important practical questions about what we really want to achieve