# Recall, Precision and Average Precision

Mu Zhu[*]

Working Paper 2004-09
Department of Statistics & Actuarial Science
University of Waterloo

August 26, 2004

## Abstract

Detection problems are beginning to attract the attention of statisticians (e.g., Bolton and Hand 2002). When evaluating and comparing different detection algorithms, the concepts of recall, precision and average precision are often used, but many statisticians, especially graduate students and research assistants doing the "dirty work," are not familiar with them. In this article, we take a systematic and slightly more formal approach to review these concepts for the larger statistics community, but the technical level of this presentation remains elementary.

**Key Words**: Cross validation; Drug discovery; Fraud detection; Mean-value theorem; ROC curve; Supervised learning.

[*] Mu Zhu is Assistant Professor, Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, ON, N2L 3G1, Canada. Email: m3zhu@uwaterloo.ca.

# 1 Introduction

Suppose we have a large collection of items, $\mathcal{C}$, of which only a fraction $\pi$ ($\pi \ll 1$) is relevant to us. We are interested in computational tools to help us identify and single out these items. The reason why an item is considered relevant depends on the context of a specific problem. For example, for fraud detection the relevant items are the fraudulent transactions; for drug discovery the relevant items are chemical compounds that show activity against a target (such as a specific virus); and so on. Typically, supervised learning methods (e.g., classification trees, neural networks) are used to build a predictive model using some training data. The model is then used to screen a large number of new cases; it often produces a relevance score or an estimated probability for each of these cases. The top-ranked cases can then be passed onto further stages of investigation.

If the top 50 cases are investigated, we shall say that these 50 cases are the ones "detected" by the algorithm, although, strictly speaking, the algorithm really does not detect these cases *per se*; it merely ranks them as being more likely than others to be what we want.

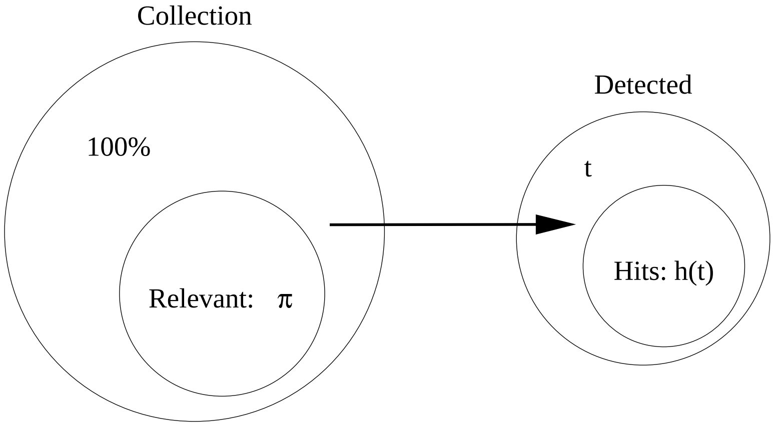

Figure 1: Illustration of a typical detection operation. A small fraction $\pi$ of the entire collection $\mathcal{C}$ is relevant. An algorithm selects a fraction $t$ from $\mathcal{C}$, of which $h(t)$ is relevant.

Suppose an algorithm detects a fraction $t \in [0, 1]$ from $\mathcal{C}$, of which $h(t) \in [0, t]$ is later confirmed to be relevant; these are often called "hits" (as opposed to "misses"). Incidentally the function $h(t)$ is sometimes called a "hit curve." Figure 1 provides a schematic illustration. Figure 2 shows some typical hit curves for a hypothetical case where 5% of all cases are relevant. The dotted curve on the top, $h_P(t)$, is an ideal curve produced by a perfect algorithm; every item detected is an actual hit until all potential hits (5% in total) are exhausted. The dotted curve at the bottom, $h_R(t)$, is that of random selection. The solid (blue) and dashed (red) curves, $h_A(t)$ and $h_B(t)$, are that of typical detection algorithms. Note that it is possible for two hit curves to cross each other.

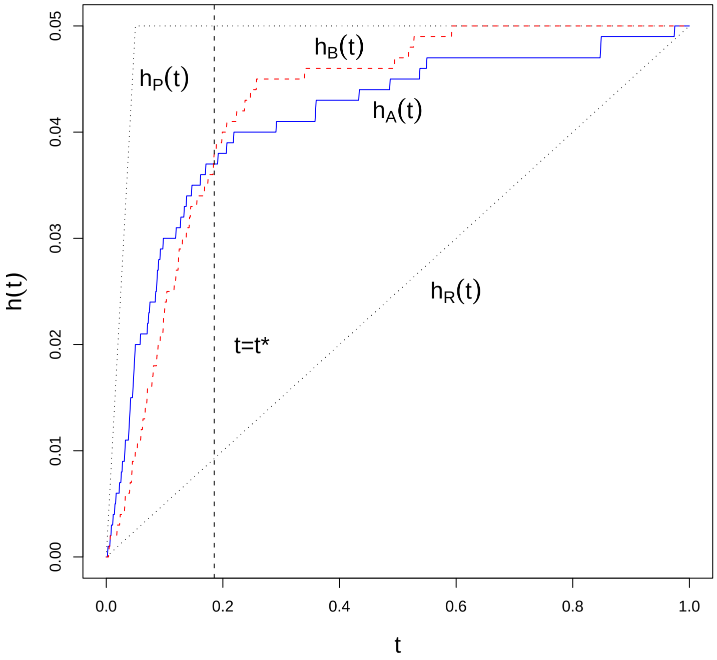

Figure 2: Illustration of some hit curves. The curve $h_P(t)$ is an ideal curve produced by a perfect algorithm; $h_R(t)$ corresponds to the case of random detection. The curves $h_A(t)$ and $h_B(t)$ are typical hit curves produced by realistic detection algorithms; note that they cross each other at $t^*$.

The hit curve $h(t)$ tells us a lot about the performance of a model or an algorithm. For example, if $h_A(t) > h_B(t)$ for any fixed $t$, then algorithm A is unambiguously superior to algorithm B; and if an algorithm consistently produces hit curves that rise up very quickly as $t$ increases, then it can often be regarded as a strong algorithm. In particular, a perfect algorithm would have a hit curve that rises with a maximal slope of one until $t = \pi$, i.e., everything detected is a hit until all possible hits are exhausted; afterwards the curve necessarily stays flat (see the curve $h_P(t)$ in Figure 2).

However, we need a performance measure not only for the purpose of evaluating or comparing algorithms, but also for the purpose of fine-tuning the algorithms at the developing stage. Often an algorithm will have a number of tuning or regularization parameters that must be chosen empirically with a semi-automated procedure such as cross validation. For example, for the $K$ nearest-neighbor algorithm we must select the appropriate number of neighbors, $K$. This is done by calculating a cross-validated version of the performance criterion for a number of different $K$s and selecting the one that gives the best cross-validated performance. For this purpose, it is clear that we prefer a performance measure that is a single numeric number, not an entire curve!

Two numeric performance measures often considered are the *recall*, defined as the probability of detecting an item given that it is relevant, and the *precision*, defined as the probability that an item is relevant given that it is detected by the algorithm. More formally, let $\omega$ be a randomly chosen item from a given collection $\mathcal{C}$; let

$$y = \begin{cases} 1 & \text{if } \omega \text{ is relevant} \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \hat{y} = \begin{cases} 1 & \text{if the algorithm detects } \omega \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\text{Recall} = \Pr(\hat{y} = 1 | y = 1) \quad \text{and} \quad \text{Precision} = \Pr(y = 1 | \hat{y} = 1).$$

At the particular detection level of $t$, the recall and precision are simply

$$r(t) = \frac{h(t)}{\pi} \quad \text{and} \quad p(t) = \frac{h(t)}{t}. \tag{1}$$

However, as will be made more specific below (Section 2), there is an inherent trade-off between recall and precision. In particular, $r(t)$ generally increases with $t$ while $p(t)$ generally decreases with $t$. Therefore, both recall and precision must be taken into account simultaneously when we evaluate or compare different detection algorithms, which is still inconvenient. Therefore, instead of recall or precision, the *average precision* is the most commonly used performance measure; we will say more about this in Section 3.

## 2 Recall-precision Trade-off

The trade-off between recall and precision is well-known (e.g., Gordon and Kochen 1989; Buckland and Gey 1994). In this section, we give a formal derivation of this trade-off. Gordon and Kochen (1989) also derived this trade-off mathematically, but we believe our derivation is more rigorous and concise than Gordon and Kochen (1989).

**Lemma 1** *Let $\pi$, $t$, $r(t)$ and $p(t)$ be defined as above. Then*

$$p(t) = \frac{\pi r(t)}{t}.$$

*Proof.* This follows directly from equation (1). Alternatively, one can establish this result directly from the definitions. By definition and using Bayes Theorem,

$$p(t) = \Pr(y = 1|\hat{y} = 1) = \frac{\Pr(\hat{y} = 1|y = 1)\Pr(y = 1)}{\Pr(\hat{y} = 1)}.$$

By definition, $\Pr(y = 1|\hat{y} = 1) = r(t)$. Moreover, $\Pr(y = 1) = \pi$ since a fraction $\pi$ of the collection is relevant, whereas $\Pr(\hat{y} = 1) = t$ since the detection level is at $t$. The result follows. ■

Note that we have expressed recall and precision both as functions of $t$. This is because as the detection level changes, the capacity of the algorithm often changes as well. Results in this section will shed some light on the basic nature of these functions. It is convenient for analytical reasons to think of $h(t)$, $r(t)$ and $p(t)$ as continuous and smooth functions of $t$ that are differentiable almost everywhere. This is reasonable especially since the collection $\mathcal{C}$ is usually quite large.

**Lemma 2** *$r(0) = 0$ and $r(1) = 1$. Furthermore, $r'(t) \geq 0$ almost everywhere on the interval $[0, 1]$.*

*Proof.* At $t = 0$, nothing has yet been detected, so $r(0) = 0$. Likewise at $t = 1$, everything has been detected, including 100% of the relative items, so $r(1) = 1$. For any increment from $t$ to $t + \Delta t$, either some relevant items are identified, in which case we have $r(t+\Delta t) > r(t)$, or no relevant item is identified, in which case recall would remain the same , i.e., $r(t + \Delta t) = r(t)$. Therefore

$$r'(t) \equiv \lim_{\Delta t \to 0} \frac{r(t + \Delta t) - r(t)}{\Delta t} \geq 0$$

for almost every $t$ where the above limit exists. ■

Lemma 2 says $r(t)$ is a non-decreasing function of $t$. For a typical detection algorithm, in fact, it can often be assumed that $r''(t) \leq 0$. In other words, the detection rate tends to slow down, reflecting the rather intuitive fact that detecting a hit becomes more and more difficult as $t$ increases. Theoretically, nothing would prevent an algorithm from behaving otherwise, but such an algorithm would not be of interest in practice. Imagine an algorithm that detects very few relevant items in the beginning and gradually starts to identify more and more relevant items as $t$ approaches 1 — one would be better off doing random selection!

**Proposition 1** *Suppose $r(t)$ is twice differentiable. We shall assume that $r''(t) \leq 0$ for almost every $t \in [0, 1]$.*

**Lemma 3** *Suppose $p(t)$ is differentiable. If Proposition 1 holds, then $p'(t) \leq 0$ for almost every $t \in [0, 1]$.*

*Proof.* By Lemma 1,

$$p(t) = \frac{\pi r(t)}{t} \implies p'(t) = \pi \frac{r'(t)t - r(t)}{t^2}.$$

It suffices to show that $r'(t)t \leq r(t)$. By the mean value theorem and the fact that $r(0) = 0$ (Lemma 2), there exist $0 \leq s \leq t$ such that

$$r(t) = r'(s)t$$

The fact that $r''(t) \leq 0$ (Proposition 1) implies $r'(s) \geq r'(t)$ since $s \leq t$. It now follows that $r(t) = r'(s)t \geq r'(t)t$. ■

Lemma 2 and Lemma 3 suggest there is an inherent trade-off between recall and precision as long as $r(t)$ is a decelerating function.

# 3 Average Precision

As mentioned earlier, the trade-off between recall and precision means both of them must be considered simultaneously when we evaluate and compare different detection algorithms. A popular measure that takes into account both recall and precision is the *average precision.* Lemma 1 makes it clear that precision and recall are related. One can express precision as a function of recall, which we denote by $p(r)$.

**Definition 1** *The average value of $p(r)$ over the entire interval from $r = 0$ to $r = 1$,*

$$\frac{1}{1-0}\int_0^1 p(r)dr = \int_0^1 p(r)dr,$$

*is called the* average precision*.*

**Lemma 4** *Assume $r(t)$ is differentiable almost everywhere, then*

$$\int_0^1 p(r)dr = \int_0^1 p(t)dr(t) = \int_0^1 p(t)r'(t)dt = \int_0^1 \frac{\pi r(t)r'(t)}{t}dt. \tag{2}$$

Although Lemma 4 is straight-forward, it does suggest, in a way, that the concept of the average precision is perhaps not the most intuitive. In particular, note that for any interval where $r'(t) = 0$, $p(t)$ is necessarily decreasing in $t$. However, since $dr = 0$, the average precision is not affected by what happens in this interval. It is instructive to examine a few concrete examples.

**Example 1 (Random Selection).** Suppose $h(t) = \pi t$, i.e., the proportion of relevant items among those detected so far stays constant at $\pi$. This is the case of random selection. In this case, $r(t) = h(t)/\pi = t$. By Lemma 4, the average precision is equal to

$$\int_0^1 \frac{\pi r(t)r'(t)}{t}dt = \int_0^1 \frac{\pi t}{t}dt = \pi. \quad \blacksquare$$

**Example 2 (Perfect Detection).** Suppose

$$h(t) = \begin{cases} t, & t \in [0, \pi]; \\ \pi, & t \in (\pi, 1]. \end{cases}$$

That is, everything detected is relevant until all relevant items are exhausted at $t = \pi$. This is the case of *ideal* detection; one can't possibly do any better than this. This implies

$$r(t) = \begin{cases} \frac{t}{\pi}, & t \in [0, \pi]; \\ 1, & t \in (\pi, 1]. \end{cases}$$

By Lemma 4, the average precision in this case is equal to

$$\int_0^1 \frac{\pi r(t) r'(t)}{t} dt = \int_0^\pi \left( \frac{\pi \times \frac{t}{\pi} \times \frac{1}{\pi}}{t} \right) dt + \int_\pi^1 \left( \frac{\pi \times 1 \times 0}{t} \right) dt = 1. \quad \blacksquare$$

**Example 3 (Practical Calculation).** In practice, the integral (2) is replaced with a finite sum

$$\int_0^1 p(t) dr(t) = \sum_{i=1}^{n} p(i) \Delta r(i)$$

where $\Delta r(i)$ is the change in the recall from $i-1$ to $i$. We illustrate this with a concrete example. Table 1 below summarizes the performance of two hypothetical algorithms, A and B.

Table 1: The Performance of Two Algorithms.

| | Algorithm A | | | Algorithm B | | |
|---|---|---|---|---|---|---|
| Item ($i$) | Hit | $p(i)$ | $\Delta r(i)$ | Hit | $p(i)$ | $\Delta r(i)$ |
| 1 | 1 | $\frac{1}{1}$ | $\frac{1}{3}$ | 1 | $\frac{1}{1}$ | $\frac{1}{3}$ |
| 2 | 1 | $\frac{2}{2}$ | $\frac{1}{3}$ | 0 | $\frac{1}{2}$ | 0 |
| 3 | 0 | $\frac{2}{3}$ | 0 | 0 | $\frac{1}{3}$ | 0 |
| 4 | 1 | $\frac{3}{4}$ | $\frac{1}{3}$ | 1 | $\frac{2}{4}$ | $\frac{1}{3}$ |
| 5 | 0 | $\frac{3}{5}$ | 0 | 0 | $\frac{2}{5}$ | 0 |
| 6 | 0 | $\frac{3}{6}$ | 0 | 0 | $\frac{2}{6}$ | 0 |
| 7 | 0 | $\frac{3}{7}$ | 0 | 0 | $\frac{2}{7}$ | 0 |
| 8 | 0 | $\frac{3}{8}$ | 0 | 1 | $\frac{3}{8}$ | $\frac{1}{3}$ |
| 9 | 0 | $\frac{3}{9}$ | 0 | 0 | $\frac{3}{9}$ | 0 |
| 10 | 0 | $\frac{3}{10}$ | 0 | 0 | $\frac{3}{10}$ | 0 |

In this case, we have

$$\text{AP(A)} = \sum_{i=1}^{10} p(i) \Delta r(i) = \left( \frac{1}{1} + \frac{2}{2} + \frac{3}{4} \right) \times \frac{1}{3} \approx 0.92$$

and

$$\mathrm{AP(B)} = \sum_{i=1}^{10} p(i)\Delta r(i) = \left(\frac{1}{1} + \frac{2}{4} + \frac{3}{8}\right) \times \frac{1}{3} \approx 0.63.$$

Intuitively, algorithm A performs better here because it detects the three hits earlier on. ■

The three examples above illustrate clearly that the notion of the average precision, though not the most intuitive from how it is defined, nevertheless behaves in a manner that is consistent with our intuition.

# 4 More Examples

## 4.1 Artificial Examples

The four hit curves plotted in Figure 2 are artificially generated using $\pi = 0.05$. Here, we calculate the average precision for these curves. The results are listed in Table 2. The results for $h_P(t)$ and $h_R(t)$ agree with what we have seen from Example 1 and 2 in Section 3. The results for $h_A(t)$ and $h_B(t)$ are not too different. This is to be expected since the two hit curves track each other fairly well, but the results tell us again that the average precision tends to favors algorithms that detect more hits earlier on.

It is often quite natural for most statisticians to think that the hit curve is somewhat similar in spirit to the well-known receiver operating characteristic (ROC) curve. Some connections definitely exist, but this example points out some important distinctions between the two. A widely popular criterion for comparing two ROC curves is the area underneath the ROC curve (e.g., Cantor and Kattan 2000). Earlier when we said that a good algorithm would tend to produce hit curves that rise up very quickly (Section 1), it may have suggested that we could also use the area underneath the hit curve as a performance measure. In this example, we can see clearly from Figure 2 that the area under $h_A(t)$ is actually less than the area under $h_B(t)$, but, because for detection problems we generally care much more about the accuracy of the top-ranked items, we actually prefer to give more weight to the earlier part of the hit curve. There, the advantage of $h_A(t)$ over $h_B(t)$ is clear. The average precision here seems to be doing a good job in this regard.

Table 2: Artificial Examples.

| Hit Curve | Average Precision |
|---|---|
| $h_P(t)$ | 1.00 |
| $h_A(t)$ | 0.26 |
| $h_B(t)$ | 0.20 |
| $h_R(t)$ | 0.05 |

## 4.2 Drug Discovery

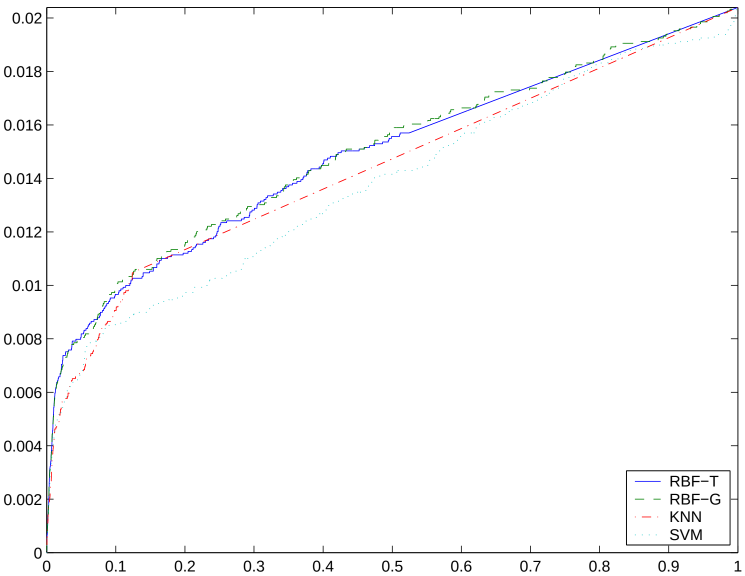

Figure 3: Drug discovery data. The hit curves produced by four competing algorithms on an independent test data set.

We now calculate the average precision for four competing algorithms applied to a real drug discovery data set. The original data are from the National Cancer Institute (NCI) with class labels added by GlaxoSmithKlein, Inc. There are 29, 812 chemical compounds, of which only 608 are active against the HIV virus; this gives $\pi \approx 0.02$. Each compound is described by $d = 6$ chemometric descriptors known as BCUT numbers. For details of what these BCUT numbers are, refer to Lam *et al.* (2002).

Using the idea of stratified sampling, the data set is randomly split into a training set and a test set, each with 14, 906 compounds, of which 304 are active compounds so that the ratio $\pi \approx 0.02$ is preserved on both the training and the test set. Each algorithm builds a predictive model using the training set. The model then ranks every compound in the test set and produces a hit curve. Based on the hit curve, the average precision is calculated.

In reality, in order to evaluate the performance of different algorithms, it is often necessary to account for the variability caused by the initial random split of the data into a training and a test set. For example, one could repeat the entire procedure a number of times, each time using a different random split of the data. For this illustration, however, we only split the data once.

The four competing algorithms are: a support vector machine (SVM) using the radial basis function kernel, K-nearest neighbors (KNN), a special adaptive radial basis function network using the Gaussian kernel (RBF-G) and the triangular kernel (RBF-T). The two RBF algorithms are first developed by Zhu *et al.* (2003). For SVM, the signed distances from the separating hyperplane are used to rank the observations in the test set. All tuning parameters for the four algorithms are selected using the same 5-fold cross-validation on the training set. The background and details of these algorithms are not directly relevant to this article; those interested can refer to Hastie *et al.* (2001) for an easy-to-read overview.

Figure 3 shows side-by-side the hit curves produced by these algorithms on the same test set. The plot suggests that, using this particular split of the data, the two RBF algorithms appear to be superior to KNN and SVM. The conclusion drawn using the average precision as a criterion (Table 3) is the same.

Table 3: Drug Discovery Example.

| Algorithm | Average Precision |
|---|---|
| RBF-T | 0.2560 |
| RBF-G | 0.2543 |
| KNN | 0.1800 |
| SVM | 0.1766 |

## 5 Upper Bound on $r'(t)$

Another important property of $r(t)$ is that its first derivative is bounded above. This is not directly relevant to the discussion in this article, but we have found in our work that it is sometimes important to be aware of this property. We state this property here for completeness.

**Lemma 5** $r'(t)$ *is bounded above:*

$$r'(t) \leq \frac{1}{\pi} \quad \forall\ t.$$

*Proof.* By increasing the detection level from $t$ to $t + \Delta t$, the fraction of items correctly identified can increase by at the most $\Delta t$. This means

$$h(t+\Delta t) - h(t) \leq \Delta t \implies \frac{h(t+\Delta t) - h(t)}{\Delta t} \leq 1.$$

Therefore,

$$h'(t) \equiv \lim_{\Delta t \to 0} \frac{h(t+\Delta t) - h(t)}{\Delta t} \leq 1.$$

Hence,

$$r(t) = \frac{h(t)}{\pi} \implies r'(t) = \frac{h'(t)}{\pi} \leq \frac{1}{\pi}. \quad \blacksquare$$

# 6 Concluding Remarks

This is a review article written primarily to serve as an easy reference. The main ideas and concepts reviewed in this article are not original, but we have organized and presented these ideas in a coherent framework that is much more concise and systematic than anything else we have encountered. We have also provided a number of novel examples. It is our hope that this article will give statisticians pursuing related work an easy-to-read and comprehensive reference to these concepts that are otherwise not familiar within the larger statistics community.

# Acknowledgment

This research is partially supported by the Natural Science and Engineering Research Council of Canada. The author would like to thank Fuping Huang and Wanhua Su for their research assistance.

# References

Bolton, R. J. and Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, **17**(3), 235–255.

Buckland, M. and Gey, F. (1994). The relationship between recall and precision. *Journal of the American Society for Information Science*, **45**(1), 12–19.

Cantor, S. B. and Kattan, M. W. (2000). Determining the area under the ROC curve for a binary diagnostic test. *Medical Decision Making*, **20**(4), 468–470.

Gordon, M. and Kochen, M. (1989). Recall-precision trade-off: A derivation. *Journal of the American Society for Information Science*, **40**(3), 145–151.

Hastie, T. J., Tibshirani, R. J., and Friedman, J. H. (2001). *The Elements of Statistical Learning: Data-Mining, Inference and Prediction*. Springer-Verlag.

Lam, R. L. H., Welch, W. J., and Young, S. S. (2002). Uniform coverage designs for molecule selection. *Technometrics*, **44**(2), 99–109.

Zhu, M., Chipman, H. A., and Su, W. (2003). An adaptive method for statistical detection with applications to drug discovery. In *2003 Proceedings of the American Statistical Association, Biopharmaceutical Section [CD-ROM]*, pages 4784–4789.