

Practical Bioinformatics

Mark Voorhies

6/25/2010

Better answers to Mike's questions

- Yes, in practice a great deal of bioinformatics programming is re-using, stitching together, and tweaking pre-existing programs. The main utility of implementing an algorithm from the ground up (as with our dynamic programming examples) is to develop your own understanding of how the algorithm works (and to confirm that a given tool works the way that you think it does).

Better answers to Mike's questions

- Yes, in practice a great deal of bioinformatics programming is re-using, stitching together, and tweaking pre-existing programs. The main utility of implementing an algorithm from the ground up (as with our dynamic programming examples) is to develop your own understanding of how the algorithm works (and to confirm that a given tool works the way that you think it does).
- We discussed that one response to finding a bug in a program is to fix it and send a patch to the original author. The other thing you should do is re-run any important calculations that were generated from the buggy code and check that your inferences are still correct. This is why it is important to keep a good record of your calculations (e.g., in the form of time-stamped python scripts). A revision-control tool like git simplifies correlating buggy code versions with the calculations that they were applied to.

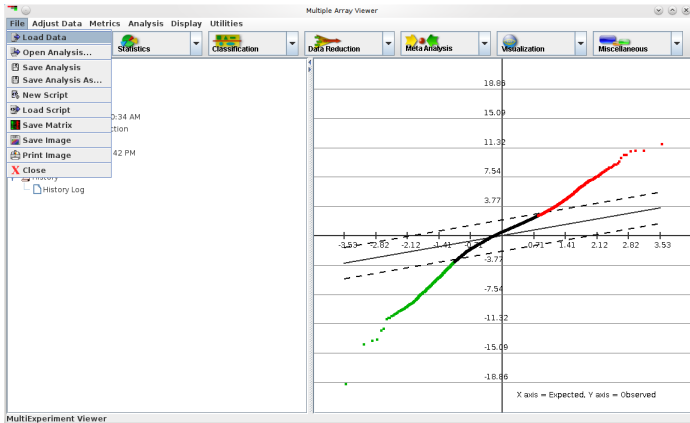
MeV SAM: Extract, normalize, and filter data

```
from ExpTransform import getExpTransform
from Database import DbLite
db = DbLite()
genome = db.GetGenomeByName("G217B")
oligos = db.Oligos()
from CdtFile import CdtFile
wgtadir = "/home/mvoorhie/papers/WGTA/output.4_1_2010/"
validCdt = CdtFile.fromCdt(open(
    wgtadir+"wgtavalid_exprvalid_orthovalid_mgtavalid.cdt"))
spots = []
for i in validCdt:
    gene = genome.GetGene(i.Uniqid())
    for j in oligos.CognateSearch(gene):
        spots.append((j, gene))
def formatRatio(x):
    if(x is None):
        return ""
    else:
        assert(type(x) == float)
        return str(x)
(header, rows) = getExpTransform("Merge of Y/pool, M/pool data",
    [i[0] for i in spots])
out = open("valid.ymp.txt", "w")
out.write("\t".join(["UID", "NAME"]+list(header))+"\n")
for ((uid, gene), row) in zip(spots, rows):
    out.write("\t".join([uid.Uid(), gene.Name()+
        [formatRatio(i) for i in row]]+"\n"))
out.close()
```

MeV SAM: Extract, normalize, and filter data

```
for (expVectorID, expVectorName) in cursor.fetchall():
    names.append(expVectorName)
    for (oligo, retval) in zip(oligos, logratios):
        if (cursor.execute("""
            SELECT weight, R, G, log2norm, flag
            FROM ExpVectorTerm, ExpSpot, ExpArray, MetaSpot
            WHERE ExpVectorID = %s
            AND ExpArray.sid = ExpVectorTerm.sid
            AND ExpArray.sid = ExpSpot.sid
            AND ExpArray.aid = MetaSpot.aid
            AND ExpSpot.mid = MetaSpot.mid
            AND MetaSpot.uid = %s
            """, (expVectorID, oligo)) > 0):
            logratio = 0.0
            for (weight, R, G, log2norm, flag) in cursor.fetchall():
                if ((flag < 0) or (None in (R, G))):
                    logratio = None
                    break
                try:
                    logratio += weight*(R-G+log2norm)
                except:
                    logratio = None
                    break
            if (2**R + 2**G < threshold):
                logratio = None
                break
        retval.append(logratio)
```

MeV SAM: Load data



MeV SAM: normalized, log transformed, TDT data

Expression File Loader

Select File Loader Help

File (Tab Delimited Multiple Sample (*.*))

Select expression data file

Selected files

Spotted DNA/cDNA Array OR Other Array type Affymetrix Array

Annotation

Retrieve Annotation from Resourcerer

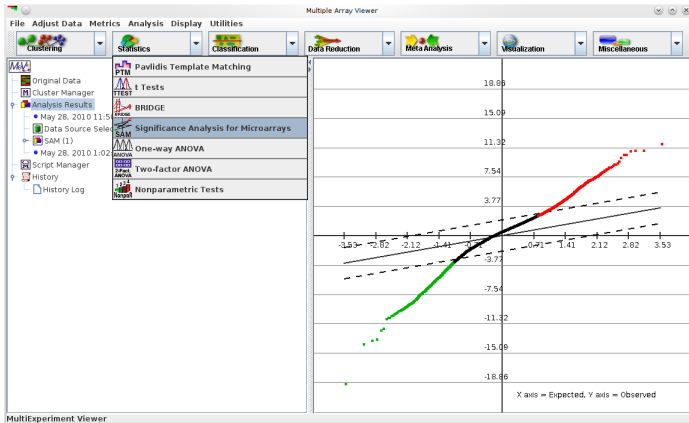
Upload annotation

Expression Table

LID	NAME	Van 37(p...	Van 25(p...	Van 37(p...	Van 25(p...	Sinem 37...	Sinem 25...	Sinem 37...	Sinem 25...
G217Borf...	HISTO_ZL...	0.1614	-0.148596	0.1753469	0.362108	0.577747	0.1519115	0.341125	0.0316291
G217Borf...	HISTO_GL...	-2.43742	0.930904	-1.90897311	1.445108	-2.817643	-0.5293995	-2.261435	-0.6800709
G217Borf...	HISTO_ZY...	-0.63367	0.191354	-0.3073431	0.900458	-0.362973	-0.3298385	-0.050225	-0.1610209
G217Borf...	HISTO_KF...	0.6876	0.184604	0.5546469	-0.384092	-0.2919985	0.553025	-0.3281709	
G217Borf...	HISTO_DA...	0.4285	0.123004	0.1704469	-0.171692	0.043097	0.4064015	0.395825	0.6291291
G217Borf...	HISTO_GV...	0.6084	0.901804	0.1737469	0.389208	0.306017	0.2735915	0.657025	0.8292291
G217Borf...	HISTO_FE...	0.5154	0.863004	0.2078469	0.445808	0.082357	0.3465315	0.578025	0.6331791
G217Borf...	HISTO_DA...	1.0539	0.191004	0.4331569	-0.576932		0.2435515	0.999145	0.8293591
G217Borf...	HISTO_DK...	-0.2252	0.299004	0.2539469	0.486508	-0.420503	-0.3414995	-0.727975	-0.4560709
G217Borf...	HISTO_ZL...	0.4153	0.436904	0.1309469	0.655508	0.113397	-0.0556985	0.238425	0.2546291
G217Borf...	HISTO_ZT...	0.4768	-0.393796	0.4358469	-0.683112	0.167047	0.5233515	0.153525	0.4971291
G217Borf...	HISTO_ZL...	0.74336	-0.143896	0.1528369	-1.411312	0.050337	-2.3118785	0.498325	-0.7542809
G217Borf...	HISTO_ZE...	-1.59	0.594804	-0.2378531	0.818408	-2.790703	-1.5506985	-2.336975	-1.4818709
G217Borf...	HISTO_RX...	0.2342	0.435304	0.3724669	-0.573562	0.186802	-0.2337685	0.280325	-0.3431709

Click the upper-leftmost expression value. Click the Load button to finish.

MeV SAM: Choose SAM



MeV SAM: Describe experiment, choose parameters

SAM Initialization

Two-class unpaired | Two-class paired | Multi-class | Censored survival | One-Class

Group Assignments

Van 37/pooled 1	<input checked="" type="radio"/> Group A	<input type="radio"/> Group B	<input type="radio"/> Neither group
Van 25/pooled 1	<input type="radio"/> Group A	<input checked="" type="radio"/> Group B	<input type="radio"/> Neither group
Van 37/pooled 2	<input checked="" type="radio"/> Group A	<input type="radio"/> Group B	<input type="radio"/> Neither group
Van 25/pooled 2	<input type="radio"/> Group A	<input checked="" type="radio"/> Group B	<input type="radio"/> Neither group
Sinem 37/pool 1	<input checked="" type="radio"/> Group A	<input type="radio"/> Group B	<input type="radio"/> Neither group
Sinem 25/pool 1	<input type="radio"/> Group A	<input checked="" type="radio"/> Group B	<input type="radio"/> Neither group
Sinem 37/pool 2	<input checked="" type="radio"/> Group A	<input type="radio"/> Group B	<input type="radio"/> Neither group
Sinem 25/pool 2	<input type="radio"/> Group A	<input checked="" type="radio"/> Group B	<input type="radio"/> Neither group

Note: Group A and Group B MUST each contain more than one sample.

Save grouping Load grouping Reset

Number of permutations

Enter number of permutations:

S0 and Q Value parameters

Select S0 using OR Enter s0 percentile (0-100)

Calculate q-values? No (quick) Yes (slow!)

Imputation Engine

K-nearest neighbors imputer Number of neighbors:

Row average imputer

Save Imputed Matrix

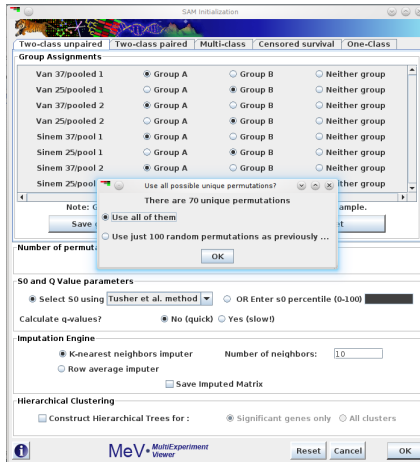
Hierarchical Clustering

Construct Hierarchical Trees for : Significant genes only All clusters

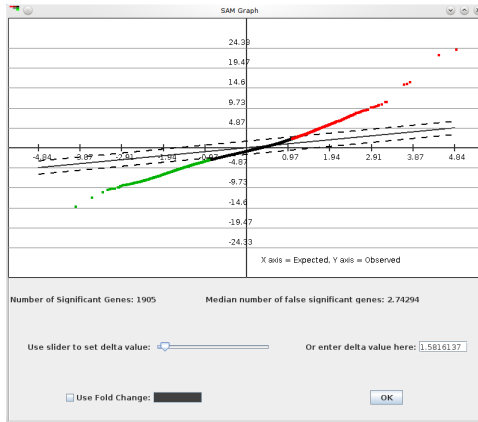
MeV MultiExperiment Viewer

Reset Cancel OK

MeV SAM: Choose permutations for FDR



MeV SAM: Choose delta



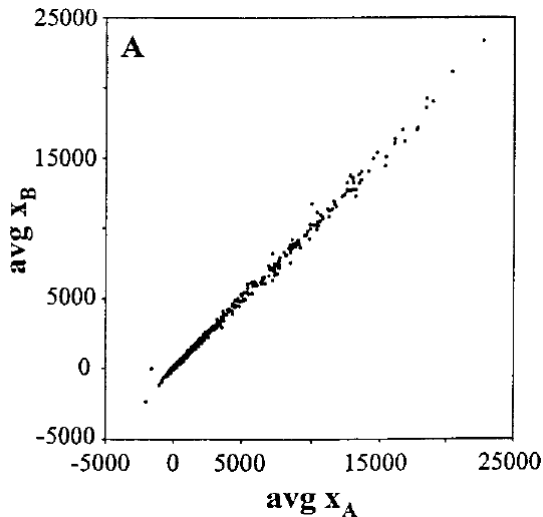
Significant analysis of microarrays applied to the ionizing radiation response

Virginia Goss Tusher, Robert Tibshirani, and Gilbert Chu

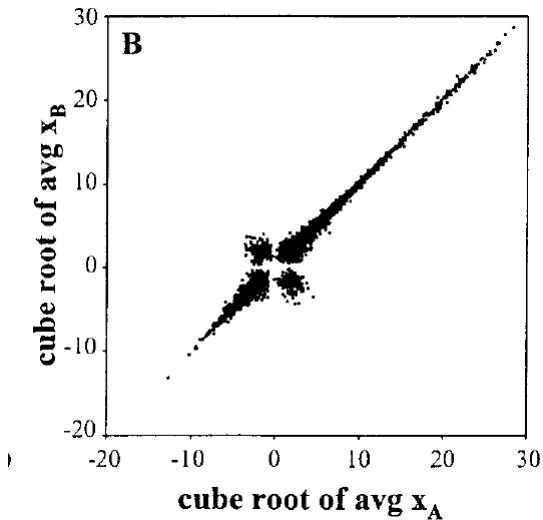
Normalizing Affy Arrays with Technical Replicates

This is very similar to mean normalization for two color arrays.

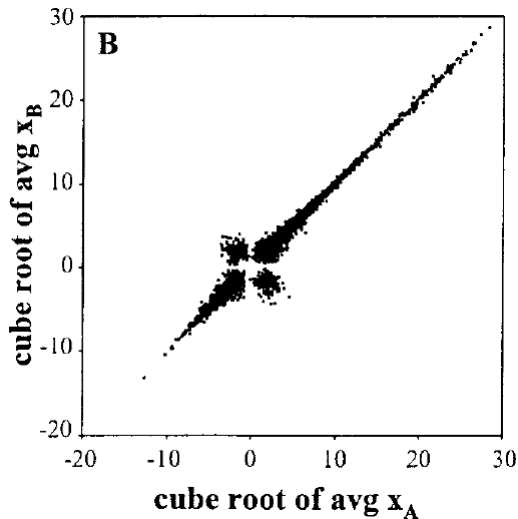
1a: Comparing normalized data



1b: Cube Root Transform

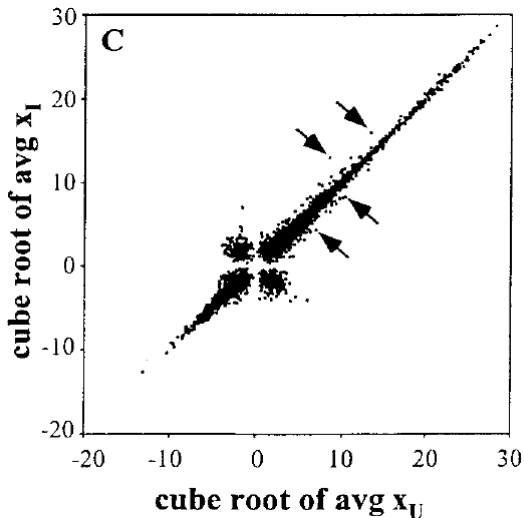


1b: Cube Root Transform

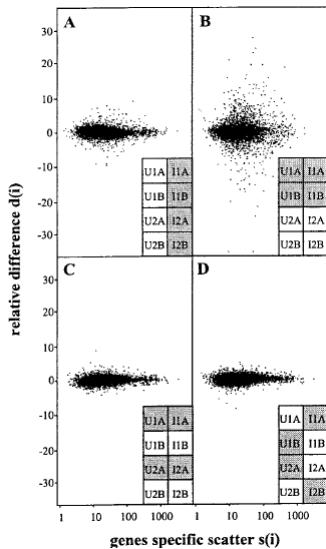


- Motivation is to get good resolution of all of the data
- Problems: weird behavior near zero, compression of error for negative values, not biologically motivated
- Better: add constant offset (or filter low intensity data) and log transform

1c: Outliers in treatment comparison

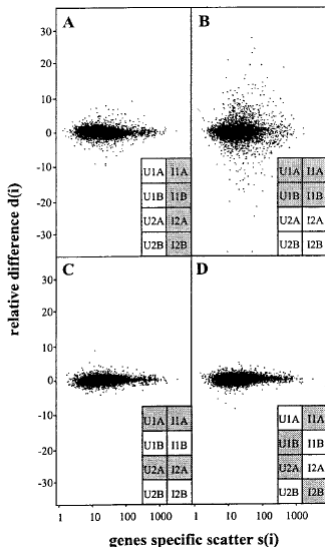


2: $d(i)$ statistic



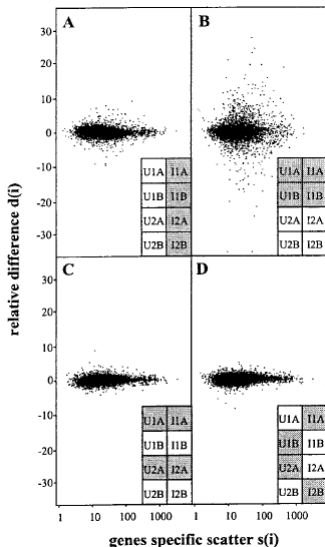
$$d(i) = \frac{\bar{x}_I(i) - \bar{x}_U(i)}{s(i) + s_0}$$

2: d(i) statistic



$$d(i) = \frac{\bar{x}_I(i) - \bar{x}_U(i)}{s(i) + s_0}$$
$$s(i) = \sqrt{a \sum_m [x_m(i) - \bar{x}_I(i)]^2 + \sum_n [x_n(i) - \bar{x}_U(i)]^2}$$
$$a = (1/n_I + 1/n_U) / (n_I + n_U - 2)$$

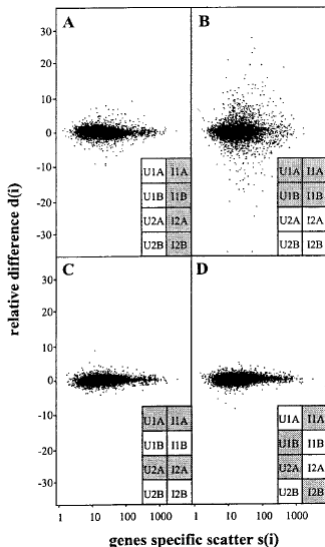
2: $d(i)$ statistic



$$d(i) = \frac{\bar{x}_I(i) - \bar{x}_U(i)}{s(i) + s_0}$$
$$s(i) = \sqrt{a \sum_m [x_m(i) - \bar{x}_I(i)]^2 + \sum_n [x_n(i) - \bar{x}_U(i)]^2}$$
$$a = (1/n_I + 1/n_U) / (n_I + n_U - 2)$$

- 1 vs. 2 = biological replicate
- A vs. B = technical replicate
- I vs. U = treatment

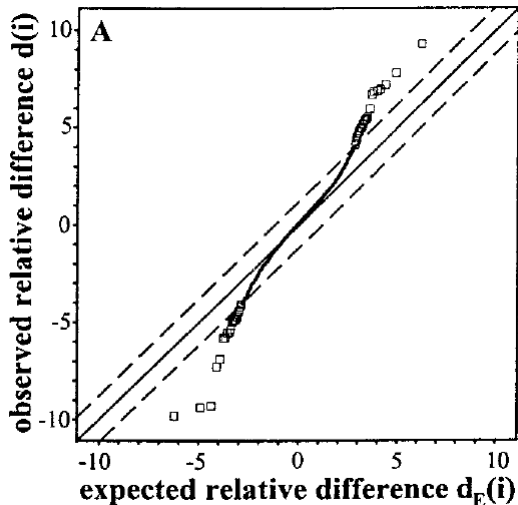
2: d(i) statistic



$$d(i) = \frac{\bar{x}_I(i) - \bar{x}_U(i)}{s(i) + s_0}$$
$$s(i) = \sqrt{a \sum_m [x_m(i) - \bar{x}_I(i)]^2 + \sum_n [x_n(i) - \bar{x}_U(i)]^2}$$
$$a = (1/n_I + 1/n_U) / (n_I + n_U - 2)$$

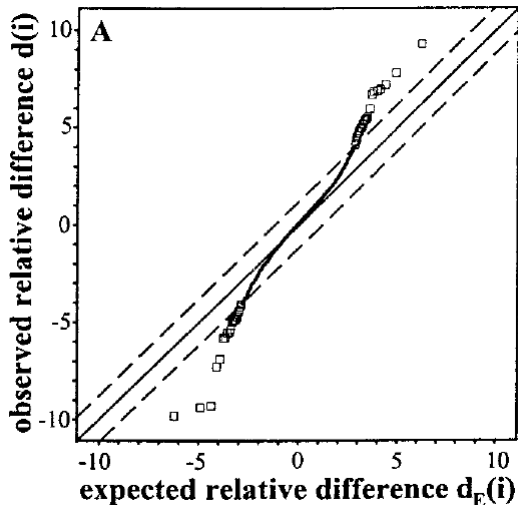
- 1 vs. 2 = biological replicate
- A vs. B = technical replicate
- I vs. U = treatment
- s_0 forces a minimum variance for the low intensity data

3a: The SAM plot



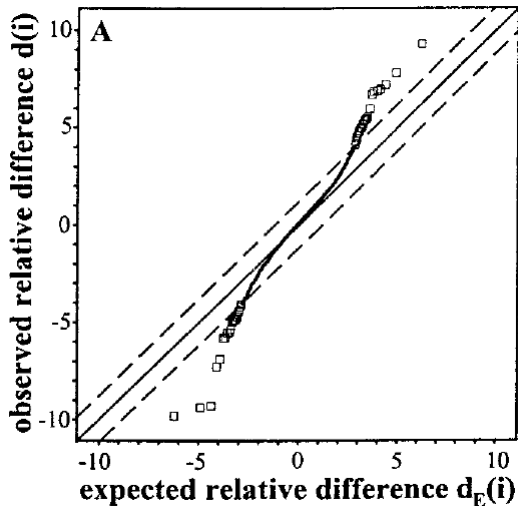
- “Expected” is the average $d(i)$ for all “balanced” permutations of the data.

3a: The SAM plot



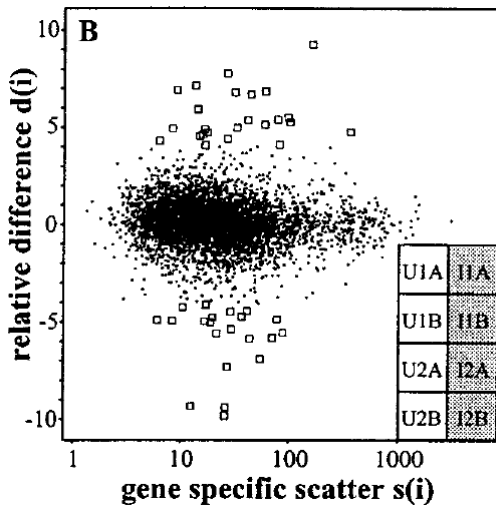
- “Expected” is the average $d(i)$ for all “balanced” permutations of the data.
- “delta” is an offset from the line of best fit, giving two diagonal thresholds.
- The plot is monotonic, so diagonal thresholds are also horizontal and vertical thresholds.

3a: The SAM plot

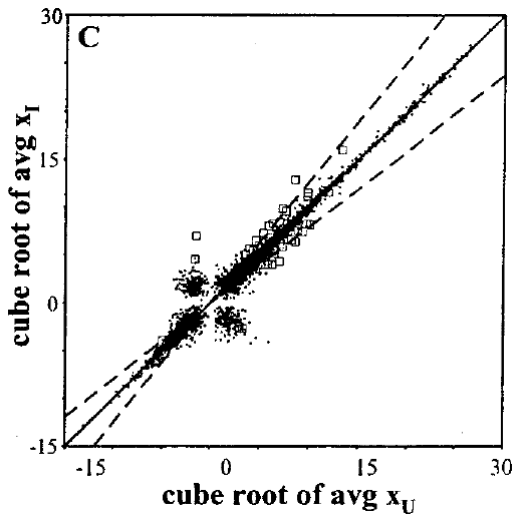


- FDR is calculated by replacing “observed” with each of the balanced permutations in turn (or a random sample for large data sets).

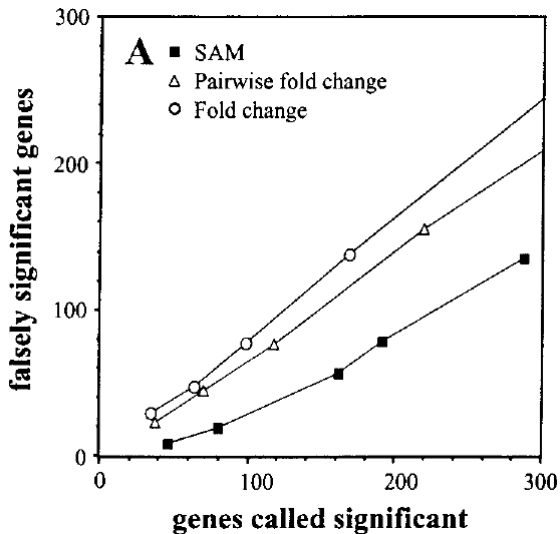
3b: Variance of significant genes



3c: Expression levels of significant genes

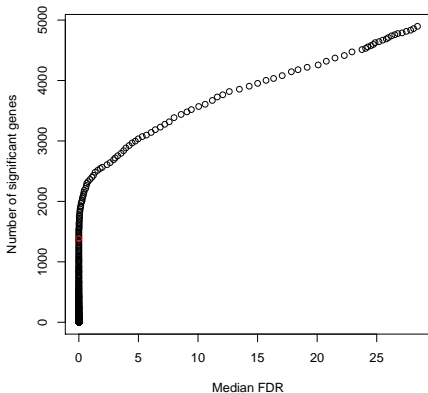


4a: Sensitivity vs. Specificity

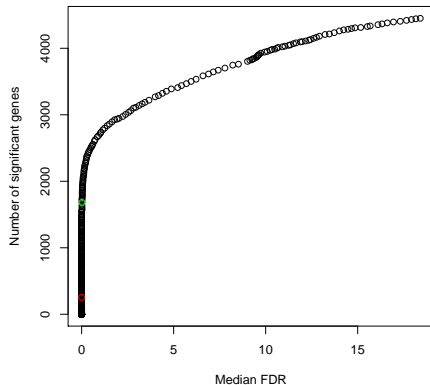


Sensitivity vs. Specificity: Pseudo-ROC plots

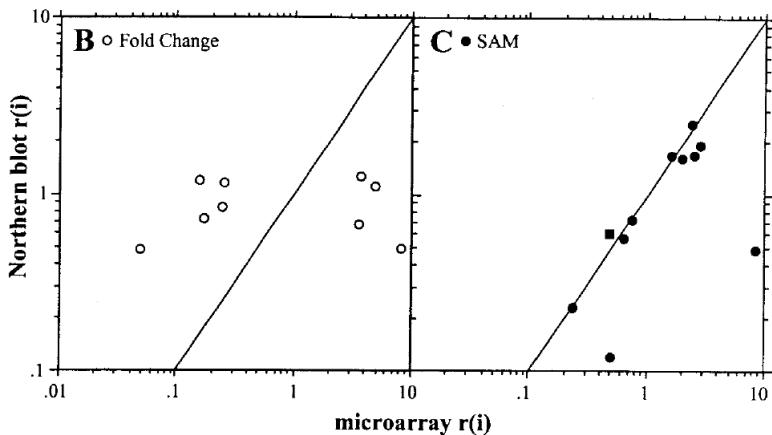
Two-class SAM



One-class SAM



4b and c: Experimental Validation



- Python modules
 - numpy (numeric)
 - matplotlib
 - scipy
 - rpy
 - Pycluster (Biopython)
 - MySQLdb
- Distributions
 - Enthought Python Distribution
 - Ubuntu Linux