8-27-2004

MergeMaid: R Tools for Merging and Cross-Study Validation of Gene Expression Data

Leslie Cope
Departments of Oncology and Biostatistics, Johns Hopkins University, cope@jhu.edu

Xiaogang Zhong
Department of Applied Mathematics, Johns Hopkins University, zhong@ams.jhu.edu

Elizabeth S. Garrett-Mayer
Departments of Oncology and Biostatistics, Johns Hopkins University, esg@jhu.edu

Giovanni Parmigiani
The Sydney Kimmel Comprehensive Cancer Center, Johns Hopkins University & Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, gp@jimmy.harvard.edu

MergeMaid:
R tools for merging and cross-study validation of gene expression data

Leslie Cope
Departments of Oncology and Biostatistics
Johns Hopkins University, Baltimore MD 21205
cope@jhu.edu

Xiaogang Zhong
Department of Applied Mathematics and Statistics
Johns Hopkins University, Baltimore MD 21218
zhong@ams.jhu.edu

Elizabeth S. Garrett-Mayer
Departments of Oncology and Biostatistics
Johns Hopkins University, Baltimore MD 21205
esg@jhu.edu

Giovanni Parmigiani
Departments of Oncology, Biostatistics and Pathology
Johns Hopkins University, Baltimore MD 21205
gp@jhu.edu

August 27, 2004

Abstract

Summary: Cross-study validation of gene expression investigations is critical in genomic analysis. We developed an R package and associated object definitions to merge and visualize multiple gene expression datasets. Our merging functions use arbitrary character IDs and generate objects that can efficiently support a variety of joint analyses. Visualization tools support exploration and cross-study validation of the data, without requiring normalization across platforms. Tools include "integrative correlation" plots that is, scatterplots of all pairwise correlations in one study against the corresponding pairwise correlations of another, both for individual genes and all genes combined. Gene-specific plots can be used to identify genes whose changes are reliably measured across studies. Visualizations also include scatterplots of gene-specific statistics quantifying relationships between expression and phenotypes of interest, using linear, logistic and Cox regression.

Availability: Free open source from url
`http://www.bioconductor.org`

Contact: Xiaogang Zhong `zhong@ams.jhu.edu`

Supplementary information: Documentation available with the package.

1 Introduction

Genomic data analysis investigates the transcriptional activity of many genes simultaneously. Because of cost and limitations in the accessibility of biological samples, most genomic investigations use a limited number of biological samples and focus on specific sample types. While this provides highly valuable insight on gene regulation, important biological and medical questions require comparison and integration of gene expression information across studies and technologies. Our ability to efficiently integrate and accumulate information from related genomic experiments will be critical in the success of the massive investment made on genomic studies. Yet, multi-study analysis to date is limited compared to the formidable potential.

To facilitate multi-study analysis, we developed an R package (MergeMaid) and the associated object definitions to merge and jointly visualize multiple gene expression data sets. Our merging function generates objects that can efficiently support a variety of joint analyses. Our visualization tools allow for exploration of the data without requiring normalization across platforms.

MergeMaid is not the first program developed to merge data from multiple sources. Many gene expression databases include annotation and data mining tools to facilitate the comparison of data from different studies. There are some freestanding programs available as well. For example, the Institute for Genetic Research (TIGR) has introduced the RESOURCERER (10) annotation database, containing annotation data for several gene expression platforms and functions to match genes across platforms using either Unigene ids (2) or the actual base sequence. GeneHopper (9) is a web-based program that uses Unigene ids to match genes across expression platforms.

MergeMaid occupies a unique place among such programs. It is our belief that issues of data storage and retrieval should be kept separate from those of analysis. Accordingly, MergeMaid is not associated with any particular database. The package does not include functions for matching probe ids across platforms. Merging is carried out on the basis of user-provided IDs, and the user is free to use RESOURCER, GeneHopper or any other method to obtain a uniform set of gene IDs.

With MergeMaid, we place the emphasis on the visualization and statistical analysis of merged gene expression data sets. This is why we have implemented the package in R (7). R is a free, open-source, object-oriented statistical programming environment, featuring extensive support for microarray analysis as part of the Bioconductor Project (5), which includes a "dynamic and extendible annotation system that collects mappings between manufacturer-specified probe set identifiers and public use nomenclature, ontology, and bibliographic systems" (6). MergeMaid is available through the Bioconductor project and is fully compatible with the other packages included there. The only dependencies are the *survival* package, which is included with the basic installation of R, and the *Biobase* package from Biocon-

ductor. The current Bioconductor release requires R 1.9 but MergeMaid is compatible with earlier versions of R, as long as a matching release of Biobase is used.

2 MergeMaid

Version 2.0 of MergeMaid includes the following primary functions, with corresponding data classes

mergeExprs	Merge Datasets into an object of class **mergeExprSet**.
intCor	Compute integrative correlation coefficients, returns an object of class **mergeCor**.
modelOutcome	Fit various models to the data. Models currently available include linear and logistic regression, and Cox hazards, returns an object of class **mergeCoeff**.

In addition, there are a number of functions for the manipulation and visualization of data. These functions depend on the data class for which they are defined and will be discussed in more detail below.

The mergeExprs function and the mergeExprSet class The primary data class in the MergeMaid package, required for all analytic functions, is the `mergeExprSet`, based on the `exprSet` class defined in bioconductor. The `mergeExprs` function is used to build `mergeExprSet` objects.

data	a list of `ExprSet` objects, one per study
geneStudy	incidence matrix indicating which genes are measured in each study.
notes	

The best way to build a `mergeExprSet` object is with the function `mergeExprs`. The function `mergeExprs` accepts expression data in a variety of formats, including `exprSet` objects, simple matrices of expression values and other `mergeExprSet`s. Any combination of these is acceptable. Merging is based on user-supplied gene ids (e.g. Genbank, Unigene, or LocusLink ID's). These IDs should make up the rownames for each expression data matrix.

Frequently an expression array will include multiple probesets for some genes, and these may be assigned the same geneid. This presents a special problem for the merging of data across platforms, becoming important when carrying out an analysis on the merged data, (e.g. regression or survival analysis) for which genes need to be unambiguously matched. In general, appropriate measures are left up to the user at ID assignment. To prevent potential problems, replicates within a dataset which still share the same ID are averaged during the merging process.

There are a number of functions to access and manipulate the data in a `mergeExprSet`.

exprs	returns the contents of the `data` slot
geneStudy	returns the contents of the `geneStudy` slot
notes	returns the contents of the `notes` slot
names	returns study names
geneNames	returns the entire list of gene IDs
phenoData	returns a list containing the phenodata (if any) included for each study
[	returns a `mergeExprSet` object containing only the indicated studies
intersection	returns a single `exprSet` containing all studies and all common genes
data<-	replaces the contents of the `data` slot
notes<-	replaces the contents of the `notes` slot
names<-	replaces the study names
geneNames<-	replaces gene IDs.

The two main analytic functions in the package are defined for `mergeExprSet` objects as well, but are discussed in separate sections, as each has an associated class.

The intCor function and the mergeCor class When working with data from different sources is to important to identify those genes which are measured in similar ways in the various datasets, and which thus can be used in joint analyses.

MergeMaid includes a gene reproducibility index called the **integrative correlation coefficient** (3; 8) which is calculated using the function `intCor`. Within each study, and for each pair of genes, we calculate the correlation coefficient of expression values across subjects. By examining whether, for a specific gene, these correlations agree across studies we can quantify the reproducibility of results without relying on direct comparison of expression across platforms.

For a pair of studies, let C^1 and C^2 be the two $M \times M$ correlation matrices for genes. For gene m, the gene-specific correlation of correlations is the correlation of the m-th row in C^1 with the m-th row in C^2. Gene-specific correlation of correlations can be used to filter genes for reproducibility across studies. The threshold should depend on the sample sizes and the levels of both biological and technological variation.

The output from the `intCor` function is an object of class `mergeCor`, containing integrative correlation coefficients for a set of studies. Such an object contains the following slots

cors	a list of correlation matrices, one per study, indexed by gene
pairwise.cors	matrix containing the integrative correlation for each pair of studies.

If n is the number of studies then for $i < j \leq n$, the pairwise correlation of correlations for studies i and j is stored in column $n(i-1) - i(i-1)/2 + j - i$ of the pairwise.cors slot.

The *total integrative correlation* for each gene is obtained by averaging the $n(n-1)/2$ pairwise integrative correlations.

The methods available for this class are:

cors	Accessor function for the cors slot.
pairwise.cors	Accessor function for the pairwise.cors slot
integrative.cors	Accessor function, returns total integrative correlation for each gene.
plot	Draw scatterplots to compare correlations of genes.

In addition, there is a function called `intcorDens`, which plots a smooth density curve for the true distribution of integrative correlation coefficients as well as two null distribution density curves, each obtained by permuting expression values randomly within gene. These plots can be used to identify a useful threshold of reproducibility. Since the permutation required the original expression data, this function is defined for mergeExprSet objects rather than for mergeCor objects, but in spirit belongs here.

The modelOutcome function and the mergeCoeff class The function `modelOutcome` calculates gene/study specific coefficients for a variety of models. The output from the `modelOutcome` function is an object of class `mergeCoeff`. Such an object contains the following slots

coeffs	a matrix of coefficients, rows=genes, columns=studies
coeff.std	matrix of standardized coefficients
zscore	matrix of zscores for the coefficients

Only 3 models are implemented in the first version of MergeMaid: linear regression, logistic regression and cox proportional hazard rate regression.

Methods for this class include:

coeff	Accessor function for the coeff slot.
coeffstd	Accessor function for the coeff.std slot.
zscore	Accessor function for the zscore slot.
plot	Draw scatterplots to compare coefficients from different studies.
coeff<-	Replacement function for the coeff slot.
coeffstd<-	Replacement function for the coeff.std slot.
zscore<-	Replacement function for the zscore slot.

The plot function is actually defined for the matrix class, rather than for the mergeCoeff class. The appropriate syntax is `plot(coeff(mergecoeff))` or `plot(zscore(mergecoeff))` so that the coefficient of interest is specified.

3 Case Study

We illustrate the use of the MergeMaid package using data from two studies of gene expression in lung cancer, labeled here Stanford (4) and Harvard (1). The discussion of data preparation and analysis here is necessarily brief. For full details see (8). The Stanford study uses two-channel cDNA arrays and includes a total of 67 lung tumor samples. The Harvard study includes 186 lung tumor samples and 17 normal samples on Affymetrix HU95a arrays.

Both studies contain samples with various histologic patterns including adenocarcinoma. Both studies find that the different histologic patterns can be identified by gene expression pattern, and also use cluster analysis to identify subclasses of adenocarcinoma.

However, although the conclusions are similar on their face, there are some important differences. The two studies disagree on the number of subclasses of adenocarcinoma (the Stanford group identified 3, while the Harvard group identified 4).

Also survival analyses by subgroup gave different results in the two studies. The Stanford group found very impressive differences in survival among the subclasses of cancer, while the Harvard group found much more modest differences. Consider, for example, the gene ornithine decarboxylase which in the Stanford study was found to be highly expressed in the class of lung cancers with the best survival outcomes, while the Harvard study found the same gene to be highly expressed in a class of cancers with poor survival outcomes.

By combining the data from the two studies and making a comparative reanalysis of both we hope to determine the amount of agreement between the two and also identify specific predictive genetic markers. We cannot expect reanalysis to provide information about every result. Results from cluster analysis may be driven by real differences in the two cohorts. No matter how carefully we filter the gene set and normalize the data, regardless of statistical methodology, it may be impossible to cross-validate findings in this situation.

Here we consider the relatively simple question of whether adenocarcinomas can consistently be distingished from other cancers by gene expression profile. Our answer is yes.

Data Preparation

All computation was conducted in R, using the MergeMaid package, and other R functions. Data for each study are available as supplements to the respective publications.

First of all, to make the studies more comparable, a logarithmic transformation was applied to the Stanford data while a cube root transformation was applied to the Harvard data. Both datasets were then represented as Bioconductor exprSet objects, and the two

were merged using Unigene Cluster IDs . There are 3171 genes common to both datasets. Of these, 307 genes met all quality criteria estalished by investigators in each study. We restrict attention to these 307 prefiltered genes. The phenodata for each study included indicators of cancer type.

Analysis and Results

As a first step, we use the function *intCor* to determine which genes are most reproducible across the two studies. Using just the Stanford data, we choose one gene g, and calculate the correlation between g and each one of the other 306 genes. We do the same with the Harvard data. This gives 306 pairs of matched correlation coefficients for each gene. The correlation coefficient of these values is our measure of how well expression values for a gene reproduce across studies.

Built in plotting functions allow the us to visulize integrative correlations. Figure 1 shows two of the genes. The gene in the top panel shows little reproducibility across the two studies, with an integrative correlation coefficient close to zero. The gene shown on the bottom has the highest integrative correlation among all genes. The distribution of the observed correlation coefficients can be seen in Figure 2, along with the expected distribution when the true correlation is zero.

We might pre-filter here, selecting the most promising genes at this stage, and using only those for further analysis. We choose instead to proceed a little farther with the whole set of genes.

We take a logistic regression approach to the problem of classifying cancers. The function *modelOutcome* calculates logistic regression coefficients for all common genes, within each study, and returns standardized coefficients and zscores as well. To facilitate cross-study comparisons, the following analyses use standardized coefficients. In Figure 3 standardized logistic regression coefficients for the two studies are plotted against one another. All genes are included in the top panel, the 50% of genes with the highest integrative correlations are shown in the middle panel, the 50% with the lowest integrative correlations are on the bottom. The correlation between logistic regression coefficients is is 0.65 when all genes are included. This improves to 0.78 for the most reproducible 50% of the genes. Even the nonreproducible genes have an overall correlation of 0.55.

We can use these plots to identify good candidates in our search for genes that distinguish between adenocarcinoma and squamous cell carcinoma. Those genes falling closest to the diagonal on this plot, having large and very similar absolute regression coefficients in both studies should be both informative about cancer type, and also good choices for platform independent markers. Formally, we averaged the study specific regression coefficients using sample size weighting, and selected genes with the highest absolute average coefficients.

Once a specific panel of genes is selected, we define the panel profile score. Within each

study, expression values are centered and standardized, gene by gene. Expressions for those genes with a negative logistic regression coefficient are multiplied by -1 so that all genes vote in the same direction. Standardized expressions are averaged across genes to give a single panel profile score for each study.

We partition the data into training and test sets to evaluate our results. The training set consists of 100 samples, 50 from each study. Each group of 50 is half adenocarcinoma and and half other cancers. The test set contains the remaining 153 subjects from the Stanford study including 89 adenocarcinomas and 18 from the Stanford study including 12 adenocarcinomas. Logistic regression coefficients were calculated and ranked using only the training data, and the sign of the coefficient is determined then and carried over to the test panel as well.

We consider 3 panels of genes. The Stanford panel consists of the 20 genes with the largest logistic regression coefficients in the Stanford study. The Harvard panel likewise consists of the 20 most promising genes in the Harvard study. To obtain the third composite panel, we first use the integrative correlation coefficient to identify the most reproducible 50% of the genes, and then select the 20 reproducible genes with the largest average regression coefficient. The Stanford and Harvard panels have only 5 genes in common. Each shares 12 genes with the common panel, including 4 of the 5 common genes. We use area under a ROC curve to measure performance.

Receiver Operator Characteristic (ROC) curves provide a compact way to represent specificity and sensitivity for a classification tool. The area under the ROC curve, which always takes values between zero and one, is commonly used to summarize results. We constructed 6 ROC curves, evaluating each of the 3 gene panels on both the 18 chip Stanford test set and the 89 chip Harvard test set. Results are summarized in Table 1. As was to be expected, each study-specific panel does much better on its own test set than it does on the other, although overall results are not too bad. We are gratified to see, though, that the composite panel outperforms both study specific panels on either data set.

4 Discussion

Our goal in creating the MergeMaid package is to facilitate the joint statistical analysis of multiple studies from possibly very different sources. By this time there is a great deal of expression data available, created using different technologies and stored in various databases the world over. Our knowledge of genes, their identifying characteristics and their functions are changing constantly and very rapidly. We believe that it is extremely important at this time to avoid dependence on any particular database and any specific method of gene identification. For the same reasons, we prefer that analytic procedures be open, transparent and easy to modify.

The heart of the package is the mergeExprSet data structure, an extension of the Bio-

conductor exprSet structure. Analytic and plotting functions are chosen to give comparable results though data comes from different sources. The functions included in the inital release are ones that we have found useful in our own work. Package capabilities are easily extended, and we expect to add additional features and functions in future releases. The open source philosophy of R makes it easy for users to customize routines to meet their own needs.

References

[1] Arindam Bhattacharjee, William G. Richards, Jane Staunton, Cheng Li, Stefano Monti, Priya Vasa, Christine Ladd, Javad Beheshti, Raphael Bueno, Michael Gillette, Massimo Loda, Griffin Weber, Eugene J. Mark, Eric S. Lander, Wing Wong, Bruce E Johnson, Todd R. Golub, David J. Sugarbaker, and Matthew Meyerson. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences USA*, 98:13790–13795, 2001.

[2] M S Boguski and G D Schuler. Establishing a human transcript map. *Nat Genet*, 10:369–371, 1995.

[3] Leslie Cope, Xiaogang Zhong, Elizabeth Garrett-Mayer, Edward Gabrielson, and Giovanni Parmigiani. Cross-study validation of the molecular profile of brca1-linked breast cancers. *Clinical Cancer Research*, under review, 2004.

[4] Mitchell E. Garber, Olga G. Troyanskaya, Karsten Schluens, Simone Petersen, Zsuzsanna Thaesler, Manuela Pacyna-Gengelbach, Matt van de Rijn, Glenn D. Rosen, Charles M. Perou, Richard I. Whyte, Russ B. Altman, Patrick O. Brown, David Botstein, and Iver Petersen. Diversity of gene expression in adenocarcinoma of the lung. *Proceedings of the National Academy of Sciences USA*, 98:13784–13789, 2001.

[5] Robert Gentleman. BioConductor: open source software for bioinformatics. `http://www.bioconductor.org`, 2003.

[6] Robert Gentleman and Vincent Carey. Visualization and annotation of genomic experiments. In G. Parmigiani, E.S. Garrett, R.A. Irizarry, and S.L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, New York, 2003. Springer Verlag.

[7] Ross Ihaka and Robert Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.

[8] Giovanni Parmigiani, Elizabeth S. Garrett-Mayer, Ramaswami Anbazhagan, and Edward Gabrielson. Cross-study comparison of gene expression data sets for the molecular classification of lung cancer. *Clinical Cancer Research*, 10(9):in press, 2004.

[9] BAT Svensson, AJ Kreeft, GJ van Ommen, JT den Dunnen, and JM Boer. Genehopper: a web-based search engine to link gene-expression platforms through genbank accession numbers. *Genome Biology*, 4:R35.1–35.5., 2003.

[10] Jennifer Tsai, Razvan Sultana, Yudan Lee, Geo Pertea, Svetlana Karamycheva, Valentin Antonescu, Jennifer Cho, Babak Parvizi, Foo Cheung, and John Quackenbush. Resourcerer: a database for annotating and linking microarray resources within and

across species. *Genome Biology*, 2:software0002.1–0002.4, 2001.

	Combined	Stanford	Harvard
Stanford	0.917	0.917	0.819
Harvard	0.913	0.872	0.906

Table 1: Area under the Reciever Operator Characteristic curve. Each of 3 panels, Combined, Stanford and Harvard, was evaluated on two test sets, one from each study.

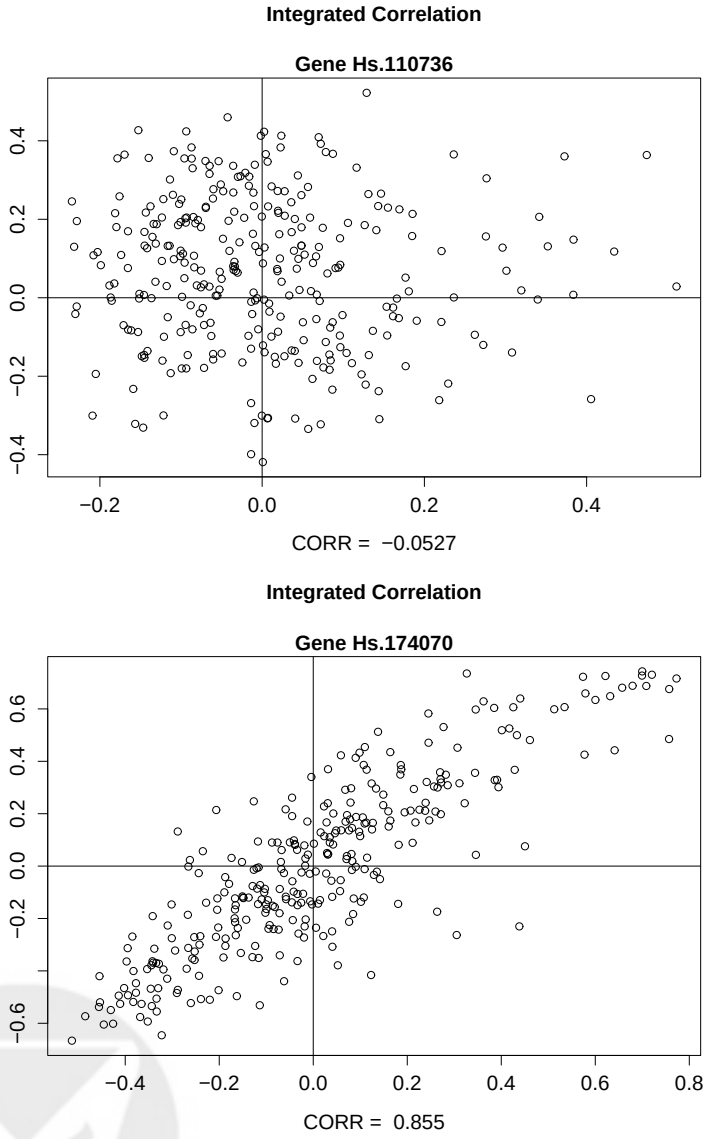

Figure 1: Visualization provided by the *mergeCor* plot function. The top panel illustrates gene-specific correlations across studies for a gene with low reproducibility, while the bottom panel shows a gene with high reproducibility.

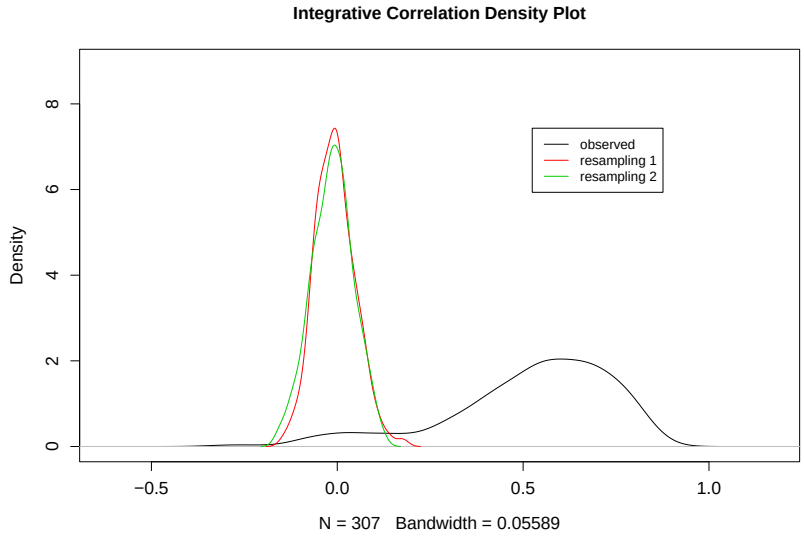

Figure 2: Visualization provided by *intcorDens*. This plot shows the observed distribution of integrative correlation coefficients as well as a null distribution obtained by permuting sample IDs.

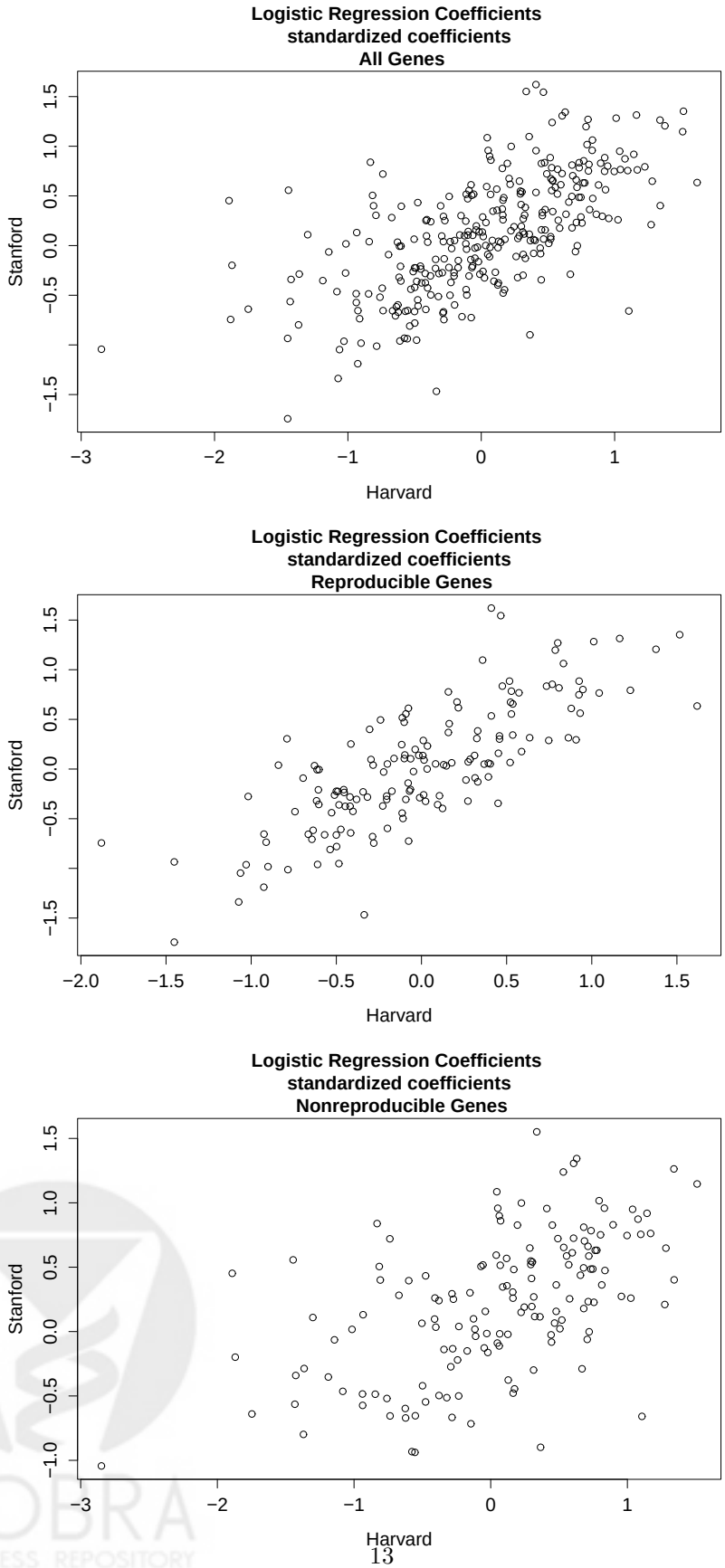

Figure 3: Visualization provided by the *mergeCoeff* plot function. Each plot shows logistic regression coefficients for each gene. The top panel includes all 307 genes. The middle panel includes the 154 genes with the highest integrative correlation coefficients. The bottom panel shows the 153 genes with the lowest integrative correlations.