

Correlate

Sparse canonical correlation analysis for the integrative analysis of genomic data

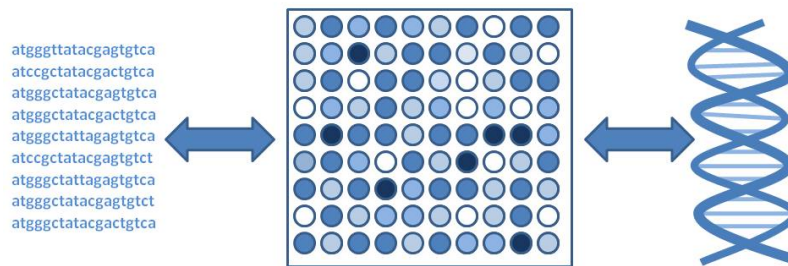
User guide and technical document

Sam Gross *

Balasubramanian Narasimhan †

Robert Tibshirani ‡

Daniela Witten §



*Department of Statistics, Harvard University, Cambridge MA 02138. Email: smgross@fas.harvard.edu.

†Department of Statistics and Department of Health Research & Policy, Stanford University, Stanford CA 94305. Email: naras@stat.stanford.edu.

‡Department of Health Research & Policy and Department of Statistics, Stanford University, Stanford CA 94305. Email: tibs@stat.stanford.edu.

§Department of Statistics, Stanford University, Stanford CA 94305. Email: dwitten@stanford.edu.

Acknowledgments: We would like to thank the R core team for permission to use the R statistical system and Thomas Baier and Erich Neuwirth for permission to use the R DCOM server.

Contents

1	Introduction	3
2	Obtaining Correlate	4
3	System Requirements	4
4	Installation	5
5	Uninstalling Correlate	5
6	Documentation	5
7	Examples	6
8	Data Formats	6
9	Handling Missing Data	7
10	Running Correlate	9
10.1	The Load Form	9
10.2	The Run Form	11
11	Interpretation of Correlate output	13
11.1	Output worksheet	13
11.2	Output figures	16
11.2.1	Permutation plots	17
11.2.2	Factor plots	17
11.3	Next steps	21
12	A typical analysis for Correlate	21
13	Some more ideas for using Correlate	21
14	Technical details of the Correlate procedure	22
14.1	Imputation of missing data	22
14.2	Sparse canonical correlation analysis	22

14.2.1	Standard data sets	23
14.2.2	Ordered data sets	23
14.2.3	Unpenalized data sets	24
14.3	Selection of Tuning Parameters and Computation of P-values	24
14.4	Obtaining multiple sets of feature weights	26
15	Frequently Asked Questions	27
15.1	General Questions	27
15.2	Correlate Usage Questions	27

List of Figures

1	Sample worksheet: breastdna	8
2	Invoking Correlate	9
3	The Load Form	10
4	The Run Form	12
5	The Correlate Controller.	13
6	New data1 vs data 2 worksheet.	14
7	Close-up of new data1 vs data 2 worksheet.	15
8	Close-up of new data1 vs data 2 worksheet.	15
9	Close-up of new data1 vs data 2 worksheet.	16
10	Close-up of new data1 vs data 2 worksheet.	16
11	The data1 vs data 2 plots worksheet: permutation plots when automatic tuning parameter selection is performed on data set 2, which is of type Standard.	18
12	The data1 vs data 2 plots worksheet: permutation plots when automatic tuning parameter selection is performed on two Standard data sets.	19
13	The data1 vs data 2 plots worksheet: A sample factor plot.	20

List of Tables

1 Introduction

Correlate is a flexible method for performing an integrative analysis of two genomic data sets.

For instance, suppose you have a single set of patient samples on which gene expression and DNA copy number measurements are available. Say you wish to identify a region of DNA copy number change that is correlated with the expression of a set of genes. Correlate performs *sparse canonical correlation analysis* (sparse CCA), described in [6], which finds a weighted average (or *linear combination*) of the DNA copy number measurements that is correlated with a

weighted average of gene expression measurements. Each DNA copy number probe and gene expression probe is assigned a weight indicating its role in the weighted average. Many weights will be exactly equal to zero, and so the results are interpretable. Similar analyses can be done using SNP data and other data types.

`Correlate` is a very general tool for correlating any two datasets. For example you can use it to correlate a set of clinical variables with a set of genomic measurements.

2 Obtaining `Correlate`

`Correlate` can be freely downloaded from the url <http://www-stat.stanford.edu/~tibs/Correlate>. Please note that the Excel front-end is an *addition* to the CCA functions in the PMA package for R.

3 System Requirements

`Correlate` requires:

- Windows 2000 or higher. `Correlate` will not work with Windows 95, 98, NT or ME.
- The latest updates for your operating system available from <http://windowsupdate.microsoft.com>. To prevent any problems, access this and other Microsoft sites using **Internet Explorer** rather than Netscape. Clicking on the **Product Updates** link pops up a box that will automate the installation of the latest patches. Beware that several (time-consuming) reboots are usually needed and you might need administrative privileges to install the patches. It is generally a good idea to update your system for security reasons any way.
- The latest version of **R**. This is freely available from the web-site <http://www.r-project.org>. Use any of the mirrors and download a Windows executable version. The installation is very simple; one has to merely run the setup program.

Please note that people have reported some problems with `Correlate` when multiple versions of R as installed on the same computer. If that is the case with your computer, you might want to uninstall all but the latest version.

- Microsoft Excel 2000 or higher. We recommend that users install appropriate Microsoft Office service packs that are available from <http://office.microsoft.com>. *Correlate will not work with earlier versions of Excel such as Excel 97.*

Obviously, performance gets better with faster processors and more RAM.

4 Installation

`Correlate` is installed by running a setup program. *You must have administrative privileges to install `Correlate`.*

Here are the details of the setup process.

1. The setup process first checks if `R` is installed. If not, you are prompted to install `R` from a specified URL.
2. If Excel or `R` is running, you will be asked to quit those programs prior to installation.
3. The setup process will install the `R DCOM` server if it is not already installed.
4. The setup process will install the `Correlate R` package.
5. The setup process will install the `CorrelateVB` Visual Basic Addin.
6. The setup process will install a Stanford Tools package that will allow you to manage the number of buttons on the precious screen real estate.
7. The Setup process might ask you to reboot if the DCOM server needed to be installed.

`Correlate` usually installs itself in `C:\Program Files\CorrelateVB`. Although users can change this directory at the time of installation although we recommend that only the drive letter be changed and not the name of the directory.

5 Uninstalling `Correlate`

Use the `Control Panel` to uninstall the software. Use the `Add or Remove Programs` menu. If you are asked if shared components should be kept and not discarded, elect to keep them as a conservative measure, unless you are really hard-pressed for space.

Note that uninstalling `Correlate` does not uninstall *all* components that were originally installed. In particular, the `R DCOM` server is left installed. You can uninstall it by using the `Control Panel` if you wish, although we recommend that you keep it.

6 Documentation

This manual for `Correlate` is also available from the `Correlate` web-site. After `Correlate` has been installed, the manual is also available as a PDF file in the subdirectory `doc` of the `Correlate` installation directory.

If you don't already have a PDF reader installed, you can do so from the web-site `www.adobe.com`.

7 Examples

Some examples of the use of `Correlate` are in the directory `C:\Program Files\CorrelateVB\Examples` in the default installation. These examples are meant to familiarize the users with the format in which `Correlate` expects the data.

8 Data Formats

`Correlate` allows for the integrative analysis of two data sets with measurements taken on a single set of samples. The two data sets should be separate worksheets in a single Excel workbook. It is fine for the Excel workbook to contain more than two worksheets; in this case, when you load in the data, `Correlate` will ask which worksheets you would like to use. See examples of acceptable workbooks in the example directory described in Section 7. Details of the worksheets are as follows:

- The first row will usually contain the sample labels. Note that both worksheets should have the same number of samples and hence the same number of sample labels. The sample labels must match up exactly between the two worksheets: any discrepancies in spelling or punctuation will result in an error. Also, the sample labels within each worksheet must be unique, since `Correlate` will use these as unambiguous identifiers of the samples. However, the samples can be in different orders in the two worksheets.
- The data usually starts in row 2 or in a later row.
- The rows of the worksheets correspond to features (for instance, genes, SNPs, or CGH spots). There can (and usually will) be different numbers of rows in the two worksheets.
- The first two columns contain the feature (row) labels. Note that unlike the sample labels, the feature labels don't need to agree with each other, and there can (and usually will) be different numbers of rows/features in the two worksheets. Often Column A will contain the gene name and Column B will contain some other information about the gene (accession number, nucleotide position, etc).
- The third column can optionally contain a `Blocking variable`. For instance, if the rows/features in a given worksheet are DNA copy number measurements on various chromosomes then column C can specify the chromosome for each row/feature. Then the user will be given the option to run `Correlate` using all of the blocks or using just a subset of the blocks (for instance, chromosomes 2 and 3 only).
- When the data is loaded in, `Correlate` will ask the user what row the sample labels are in (usually this will be row 1), what row the data begins in (usually this will be row 2), and whether the 3rd column is a `Blocking variable`.

- **IMPORTANT:** If your data has some sort of spatial ordering that you want `Correlate` to know about - for instance, if one of the worksheets corresponds to DNA copy number measurements - then the rows of the worksheet should respect that ordering. For instance, suppose you have DNA copy number measurements for chromosomes 1, 2, ..., 23. Then the chromosome number can be a blocking variable in column C. Within each value of the blocking variable - say, within chromosome/block 1 - the rows should be ordered according to their chromosomal position.

To make this a little more concrete, an example worksheet is shown in Figure 1. This worksheet contains a blocking column in Column C.

There are two example data sets that come with `Correlate`: a breast cancer data set consisting of 19,672 gene expression and 2,149 DNA copy the row-wise average for all experiments.

9 Handling Missing Data

`Correlate` imputes missing values via a K-Nearest Neighbor algorithm in the R package `impute`. Full details may be found in [2]. $k = 10$ nearest neighbors are used. Here is how it works:

1. For each feature i having at least one missing value:
 - (a) Let S_i be the samples for which feature i has no missing values.
 - (b) Find the k nearest neighbors to feature i , using only samples S_i to compute the Euclidean distance. When computing the Euclidean distances, other features may have missing values for some of the samples S_i ; the distance is averaged over the non-missing entries in each comparison.
 - (c) Impute the missing sample values in feature i , using the averages of the non-missing entries for the corresponding sample from the k nearest neighbors.
2. If a feature still has missing values after the above steps, impute the missing values using the average (non-missing) expression for that feature.

If the number of features is large, the near-neighbor computations above can take too long. To overcome this, we combine the K-Nearest Neighbor imputation algorithm with a **Recursive Two-Means Clustering** procedure:

1. If number of features p is greater than p_{max} (default 1500):
 - (a) Run a two-means clustering algorithm in feature space, to divide the features into two more homogeneous groups. The distance calculations use averages over non-missing entries, as do the mean calculations.

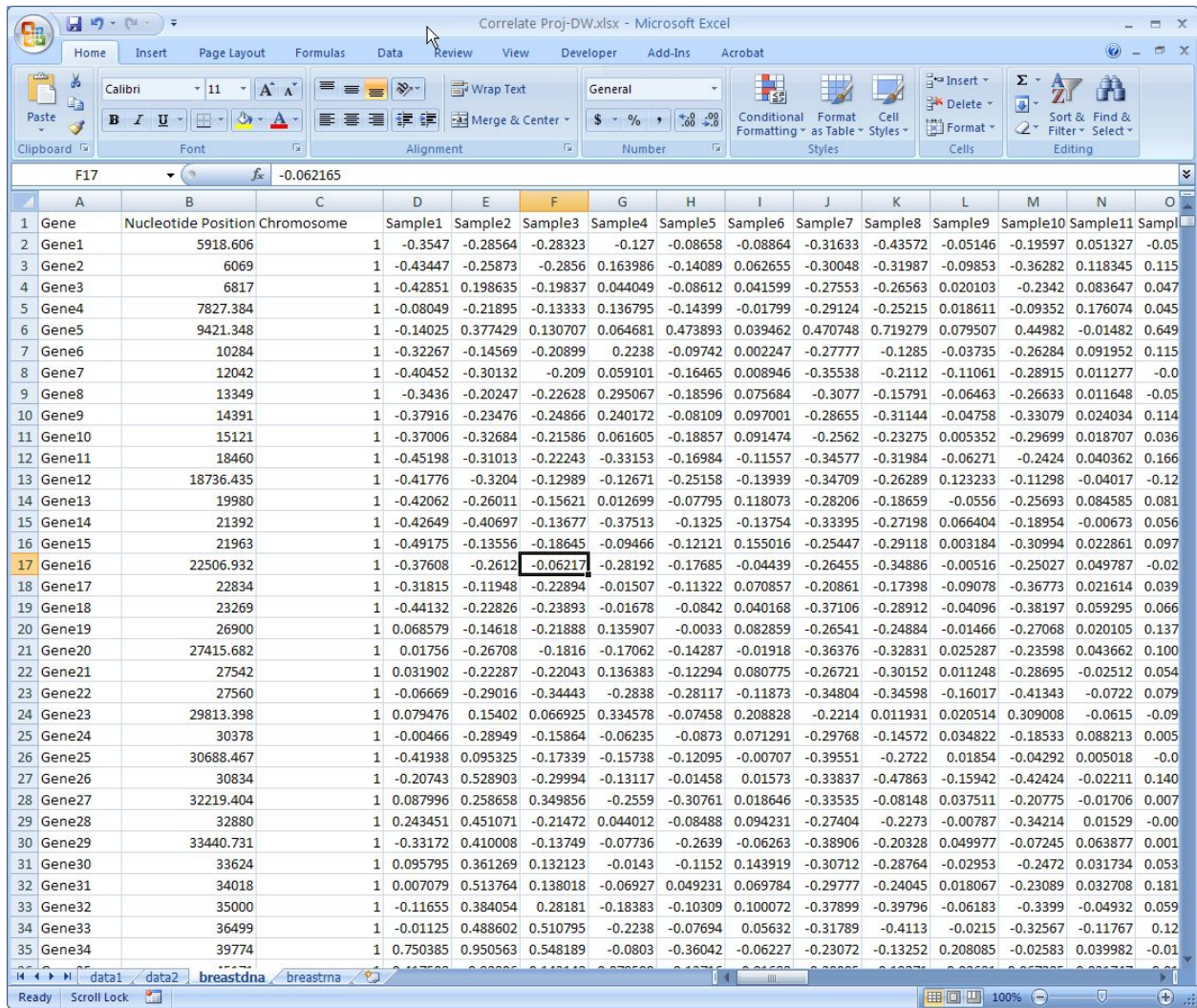


Figure 1: Sample worksheet: breastdna

- (b) Form two smaller expression arrays, using the two subsets of features found in (a). For each of these, recursively repeat step 1.
2. If p is less than p_{max} , impute the missing features using K-Nearest-Neighbor averaging.

10 Running Correlate

10.1 The Load Form

To begin, click on the `Correlate` button in the toolbar. See illustration in figure 2.

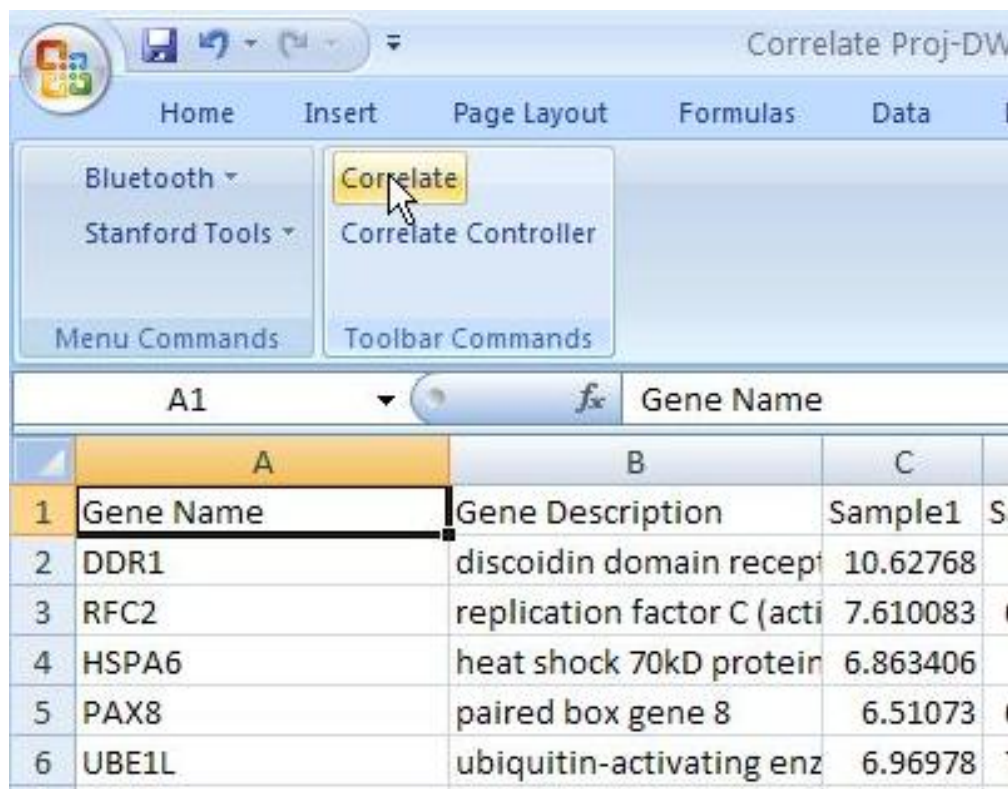


Figure 2: Invoking `Correlate`

A dialog form shown in figure 3 now pops up to help you load in the data. You have to select the worksheets that will serve as data sets 1 and 2 for the analysis. (If only 2 worksheets are present then this will be an easy choice; if more than 2 worksheets are present, then click on the two corresponding to the data that you wish to use.)

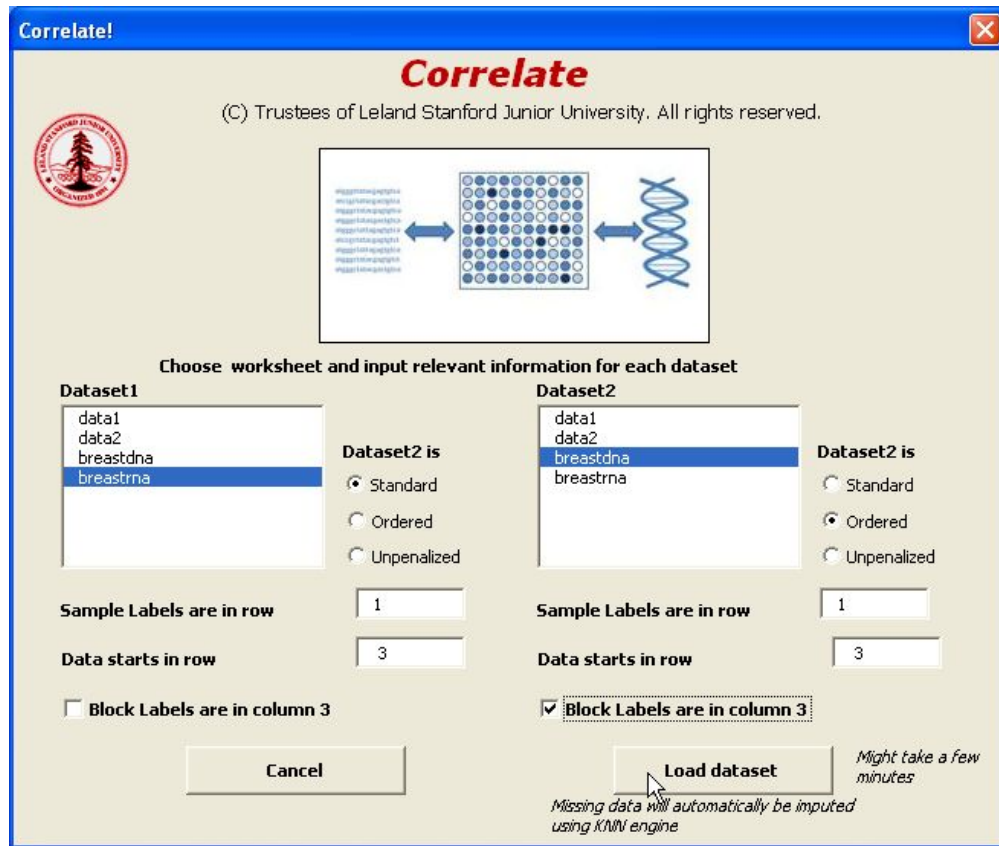


Figure 3: The Load Form

For each data set, you must specify what type of penalization you would like applied to the data. The three options are:

1. **Standard**. This will result in weights that are *sparse*: that is, some of the weights will be exactly equal to zero. This is the default setting for *gene expression*, *SNP*, and most other data.
2. **Ordered**. This will result in canonical vectors that are *sparse* and *smooth*: that is, many of the feature weights will be exactly equal to zero, and adjacent features will tend to have similar weights. This is appropriate for a worksheet that corresponds to *DNA copy number* data.
3. **Unpenalized**. This will result in all of the features/rows having non-zero weights. This can be used if the rows correspond to quantitative clinical measurements (blood pressure,

age, etc.). It can only be used if there are fewer rows than columns. Therefore, *do not* use this option if the rows correspond to genes, SNPs, or other genomic data with many measurements.

Next, for each data set, specify which row contains the sample labels (usually this will be row 1) and in which row the data begins (usually row 2 or 3).

Finally, specify for each data set whether column C contains blocking labels, as described in Section 8. If so, then it will be assumed that the data begins in column D. If not, then it will be assumed that the data begins in column C.

Click the `Load Dataset` button to do the analysis. It may take a few minutes to load in large data sets.

If you had any missing data in either of the data sets, a new worksheet named `xxx - imputed` containing the imputed dataset is added to the workbook, where `xxx` contains the name of the worksheet that contained the missing data. This data can be used in subsequent analyses to save time. If there is no missing data, this worksheet is not added.

10.2 The Run Form

After the data has been loaded, a new form will appear containing options for you to run the analysis. Figure 4 displays a screen shot of the Run Form. For each data set, specify which blocks should be used in the analysis (if applicable) by using the `Select All` and `Unselect All` boxes and by clicking on individual blocks. Next, specify how tuning parameters should be selected. Normally you will want to use `Auto` selection, in which case `Correlate` will use a permutation approach to determine the tuning parameter values to be used. If you know which tuning parameters you would like to use, select `Manual` and enter the tuning parameter in the box. Details about the meaning of the tuning parameters and the selection of the tuning parameters are given in Section 14.

Next, specify whether you would like the resulting feature weights to be `Any sign`, `Positive`, or `Negative`. This is an option only for `Standard` and `Unpenalized` data (specified in the `Load Form`, Section 10.1). Details are `Forcing the weights to be Positive`, or `Negative` can make the resulting linear combination more interpretable. More details are given in Section 14.

Now, specify how many sets of feature weights (canonical vectors) you wish to compute. Also, specify how many permutations are desired. Permutations are needed in order to determine the tuning parameter for a `Standard` data set in an automatic way, and in order to report a p-value. In fact, the only time that permutations are not needed is if all of the data sets are of type `Ordered` or `Unpenalized` and if a p-value is not wanted. We recommend running at least 50 or so permutations for any final analyses. But since permutations are the slowest part of `Correlate`, the default number of permutations is 5 in order to make it run more quickly.

If all data sets are of type `Ordered` or `Unpenalized`, *or* of type `Standard` with the tuning

Figure 4: The Run Form

parameter specified manually, then the user has the option to *not* do any permutations in order to save time. In this case, no p-value will be output. To do this, un-check the Output p-value? box. If one of the data sets is Standard and its tuning parameter was not chosen manually, then you do not have the option to uncheck this box.

Note that running permutations can be quite slow. If permutations are performed, a progress bar will pop up to let you know what fraction of the permutations have been completed.

The software adds one or two more worksheets to the workbook. The sheet named data1 vs data2 is used for writing any output. The sheet named data1 vs data2 plots will be created in most cases, and contains some plots. (data1 and data2 will be replaced with the names of the worksheets in use.)

After you use the Run Form once, if you wish to re-run Correlate on the same data then you can invoke the Correlate Controller which will allow new analyses to be run without having to re-load the same data. See Figure 5.

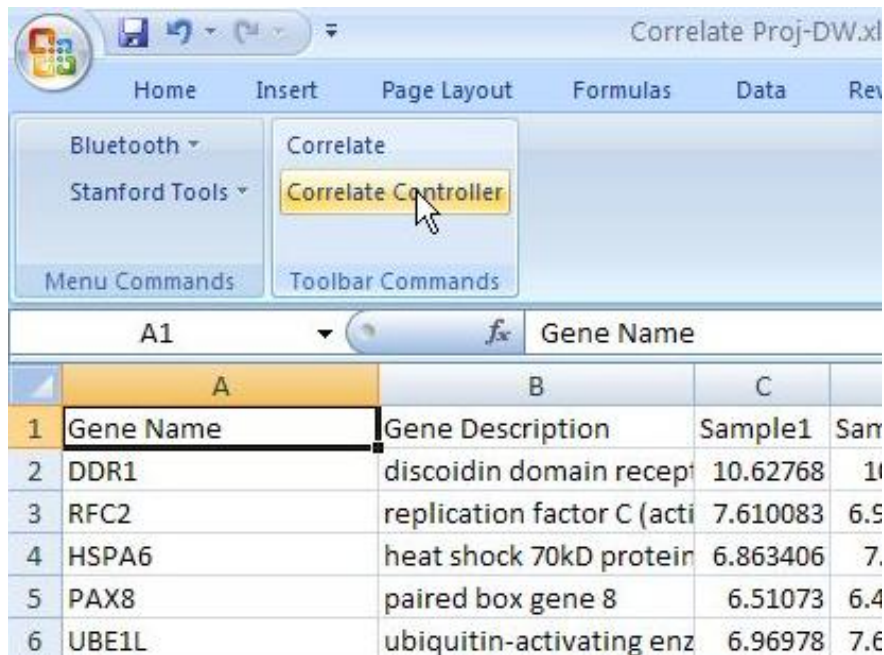


Figure 5: The Correlate Controller.

11 Interpretation of Correlate output

11.1 Output worksheet

Correlate outputs a new worksheet titled data1 vs data2. (data1 and data2 will be replaced with the names of the data sets on which Correlate was run.) A screenshot of the new worksheet can be seen in Figure 6.

There are a few different pieces of information in this new worksheet, which we will look at individually.

First, look at the top left corner of the worksheet (Figure 7). For each of the two data sets, this region of the worksheet specifies the following:

1. The *names* of the data sets.

		dataset1	dataset2			summary statistics	
name	breastrna	breastdna	# of samples	89			component 1
type	standard	ordered	# of Components	1	dataset1 nonzero weights		1111
penalty	0.166667	0.03178	random seed	65535	dataset2 nonzero weights		71
output constraint	any sign	any sign	# of permutations	5	cors		0.80574
# of rows	19671	135	p-value				
first dataset output				second dataset output			
labels	weights		labels	weights			
SF3B4	splicing fa	0.101921	Gene82	156282	1		0.137347251
HSPC003	HSPC003	0.097829	Gene83	157693	1		0.137347251
RAB3-GAP150	rab3 GTPa	0.091918	Gene84	157812	1		0.137347251
MYO14	hypotheti	0.09133	Gene85	159634	1		0.137347251
GNPAT	glycerone	0.090716	Gene86	159925.95	1		0.137347251
B4GALT3	UDP-Gal:t	0.090573	Gene87	160212.83	1		0.137347251
NDUFS2	NADH deH	0.09023	Gene88	163241	1		0.137347251
FLJ12671	hypotheti	0.087871	Gene89	164403	1		0.137347251
MRPL24	mitochoni	0.087298	Gene90	164792	1		0.137347251
LOC51107	CGI-78 prc	0.085159	Gene91	164980	1		0.137347251
HSPC155	hypotheti	0.08502	Gene92	165280	1		0.137347251
PEX11B	peroxison	0.084839	Gene93	165442	1		0.137347251
TPR	translocat	0.083614	Gene94	167345	1		0.137347251
GGPS1	geranylge	0.083482	Gene111	198434	1		0.137232269
C1orf27	chromoso	0.083434	Gene112	199181.29	1		0.137232269
FL110876	hypotheti	0.081533	Gene113	99369	1		0.137232269

Figure 6: New data1 vs data 2 worksheet.

2. The *type* of data set (Standard, Ordered, or Unpenalized)
3. The *tuning parameter penalty* used (this was either specified by the user in the Run Form, or was automatically selected using a permutation approach).
4. The *output constraint* (if the data set was of type Standard or Unpenalized, then the user was able to select a sign constrain in the Run Form).
5. The *number of rows of the data set* - a sanity check to confirm that Correlate read in the data properly.

Now, look at the top center of the worksheet (Figure 8). The following is specified:

1. The *number of samples* in the two data sets run.
2. The *number of components* requested by the user in the Run Form.
3. The *random seed* used by the user in the Run Form (default value is 65535).

	A	B	C
1	Correlate! Output		
2			
3		dataset1	dataset2
4	name	breastrna	breastdna
5	type	standard	ordered
6	penalty	0.166667	0.03178
7	output constraint	any sign	any sign
8	# of rows	19671	135
9			
10			

Figure 7: Close-up of new data1 vs data 2 worksheet.

4. The *number of permutations* performed (specified by the user in the Run Form).
5. The *p-value* obtained, if requested by the user in the Run Form.

	# of samples	89
	# of Components	1
	random seed	65535
	# of permutations	5
	p-value	

Figure 8: Close-up of new data1 vs data 2 worksheet.

We now move to the top right of the worksheet (Figure 9):

1. *dataset 1 nonzero weights* is the number of non-zero weights obtained on data set 1 when Correlate was run.
2. *dataset 2 nonzero weights* is the number of non-zero weights obtained on data set 2 when Correlate was run.
3. *cors* is the correlation between the weighted sum of features in data set 1 and the weighted sum of features in data set 2. The correlation will be a number between -1 and 1, and will generally be close to 1 indicating that weighted sums were found that were highly correlated with each other.

Note that if the Run Form was run with more than one component requested, then Figure 9 will contain additional columns containing the same information for the additional components.

summary statistics		
		component 1
dataset1 nonzero weights		1111
dataset2 nonzero weights		71
cors		0.80574

Figure 9: Close-up of new data1 vs data 2 worksheet.

Finally, we move to the middle and bottom of the worksheet. This contains the information about the weights obtained for each data set. Under `first dataset output`, the features in data set 1 with non-zero weights are listed (sorted by weight). Any identifiers loaded into `Correlate` for data set 1 will be used. A large (absolute value) weight indicates a feature that is very important to the correlation found. A small (absolute value) weight indicates a feature that is less important. Features that have zero weight are not listed. The results under `second dataset output` can be interpreted the same way.

	first dataset output		second dataset output	
	labels	weights	labels	weights
L4	SF3B4	splicing fa 0.101921	Gene82	156282 1 0.137347251
L5	HSPC003	HSPC003 g 0.097829	Gene83	157693 1 0.137347251
L6	RAB3-GAP150	rab3 GTPa 0.091918	Gene84	157812 1 0.137347251
L7	MYO14	hypotheti 0.09133	Gene85	159634 1 0.137347251
L8	GNPAT	glycerone 0.090716	Gene86	159925.95 1 0.137347251
L9	B4GALT3	UDP-Gal:t 0.090573	Gene87	160212.83 1 0.137347251
L20	NDUFS2	NADH del 0.09023	Gene88	163241 1 0.137347251
L21	FLJ12671	hypotheti 0.087871	Gene89	164403 1 0.137347251
L22	MRPL24	mitochoni 0.087298	Gene90	164792 1 0.137347251
L23	LOC51107	CGI-78 prc 0.085159	Gene91	164980 1 0.137347251
L24	HSPC155	hypotheti 0.08502	Gene92	165280 1 0.137347251
L25	PEX11B	peroxison 0.084839	Gene93	165442 1 0.137347251
L26	TPR	translocat 0.083614	Gene94	167345 1 0.137347251
L27	GGPS1	geranylge 0.083482	Gene111	198434 1 0.137232269
L28	C1orf27	chromoso 0.083434	Gene112	199181.29 1 0.137232269

Figure 10: Close-up of new data1 vs data 2 worksheet.

11.2 Output figures

`Correlate` produces two types of output figures: permutation plots and factor plots. Depending on the settings used to run `Correlate`, neither, one, or both of these plots may be produced.

If output figures are produced, then they can be found in a new worksheet called `data1 vs data2 plots` that will be created when `Correlate` is run.

Note that when multiple plots are output, they may appear on top of each other and so it will

be necessary to move around the plots in order to view all of them. Also, if `Correlate` is run repeatedly on the same data sets and the previous `data1 vs data2 plots` worksheet is not removed, then additional plots will be added to the same worksheet. In order to avoid confusion, after you run `Correlate`, copy and paste any plots that you wish to keep into a separate file, and delete the plots in `data1 vs data2plots` or delete the worksheet itself.

11.2.1 Permutation plots

If `Correlate` is run with at least one data set of type `Standard` with automatic tuning parameter selection, then permutations are performed in order to select a tuning parameter value. (For more details on automatic tuning parameter selection and permutations, see Section 14.3.) If permutations are performed, then two plots will be produced, as shown in Figures 11 and 12.

In both plots, the x-axis shows the tuning parameter or penalty used on the data set(s). The penalty must lie between 0 and 1; usually a grid of ten values from 0.1 to 0.7 will be used.

The first plot shows the correlations between the linear combinations of variables obtained on the real and permuted data. For each value of the tuning parameter, the correlation between the linear combinations of the variables in the two data sets are shown as small black circles. For each value of the tuning parameter, the small green circles indicate the correlations between the linear combinations of the variables obtained using the *permuted* data. A good tuning parameter value is one for which the true (black) correlation is large relative to the permuted (green) correlations.

In order to further assess which tuning parameter is best, the second plot shows the *z-statistics* obtained for each tuning parameter value. For each tuning parameter value, the z-statistic is a measure of the correlation obtained on the real data relative to the correlation on the permuted data. A large z-statistic indicates a good choice of tuning parameter. `Correlate` will automatically perform sparse CCA using the tuning parameter corresponding to the largest z-statistic. However, it is best to examine this plot and choose a penalty that gives a large z-statistic and also a reasonable number of non-zero weights. For instance, sometimes the highest z-statistic will be obtained when all but 2 weights are non-zero (corresponding to a tiny penalty) and a z-statistic that is almost as large can be obtained using a larger penalty. In this case, one might want to use a larger penalty. In other words, this plot should be used for guidance, but one should always feel free to re-run `Correlate` using a tuning parameter other than the one corresponding to the largest z-statistic.

11.2.2 Factor plots

When a data set is of type `Ordered`, the worksheet `data1 vs data2 plots` will contain *factor plots* showing the resulting weights for that data set. If the `Blocking Variable` was used for a given data set, then the weights for that data set will be separated into the individual blocks. Note that the weights within each block are expanded/shrunk to fit the space available in the figure, and so one should compare weights *within* a block but not *between* blocks. Positive weights are plotted in red, and negative weights are plotted in green. For each data set of type

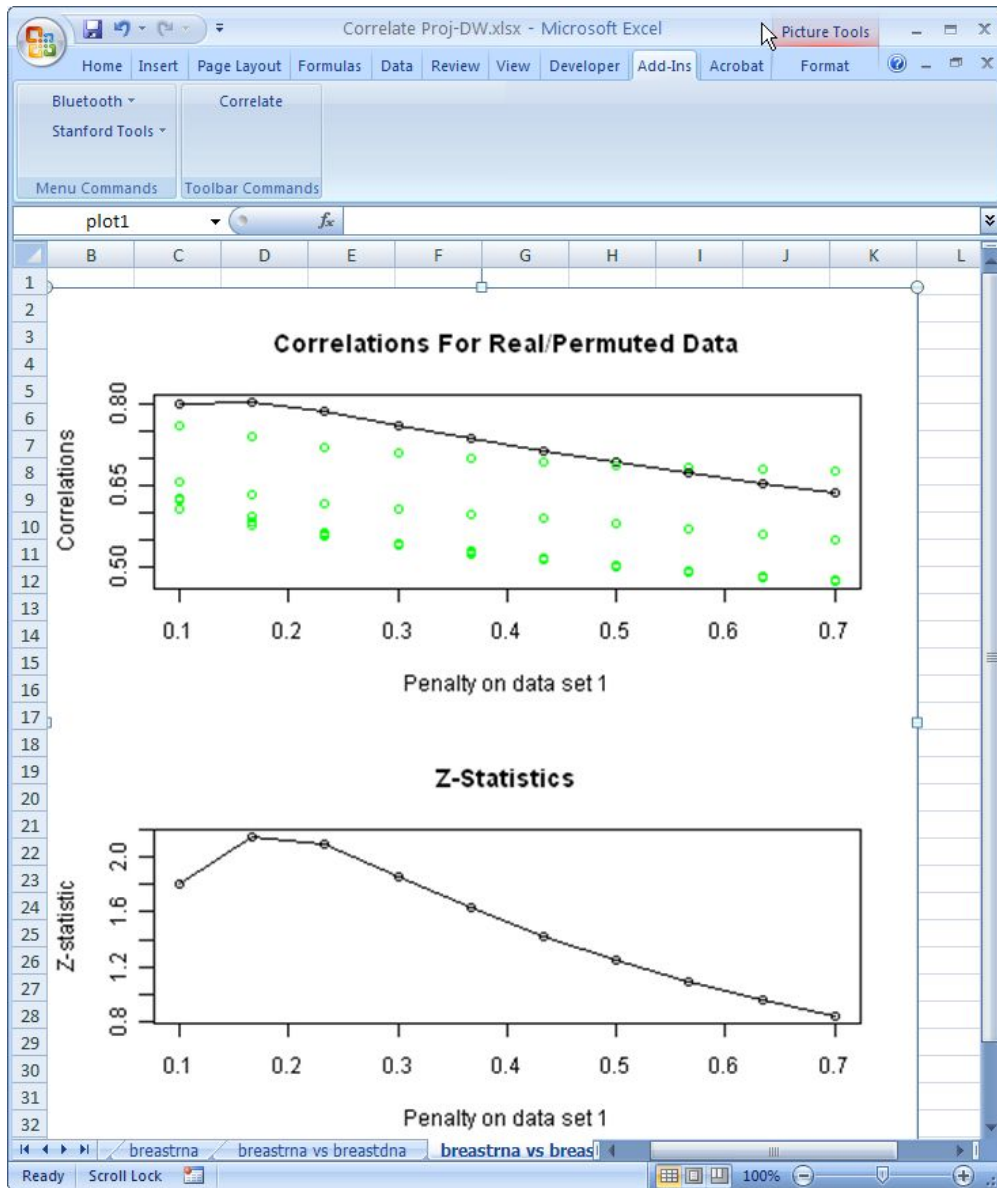


Figure 11: The data1 vs data 2 plots worksheet: permutation plots when automatic tuning parameter selection is performed on data set 2, which is of type Standard.

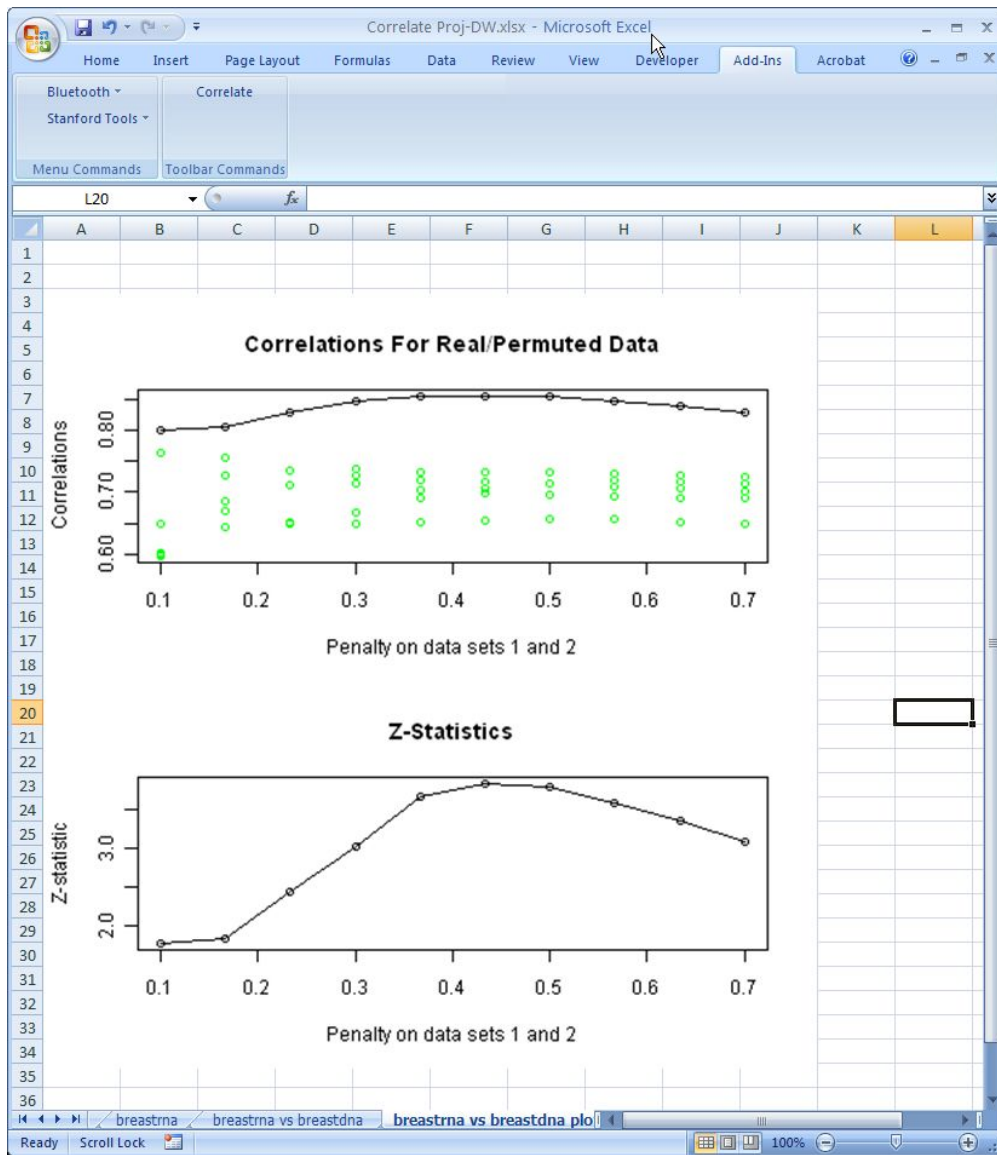


Figure 12: The data1 vs data 2 plots worksheet: permutation plots when automatic tuning parameter selection is performed on two Standard data sets.

Ordered, a plot will be made for each component requested. That is, if three components were requested on the Run Form, then three plots will be made. An example of a factor plot is shown in Figure 13.

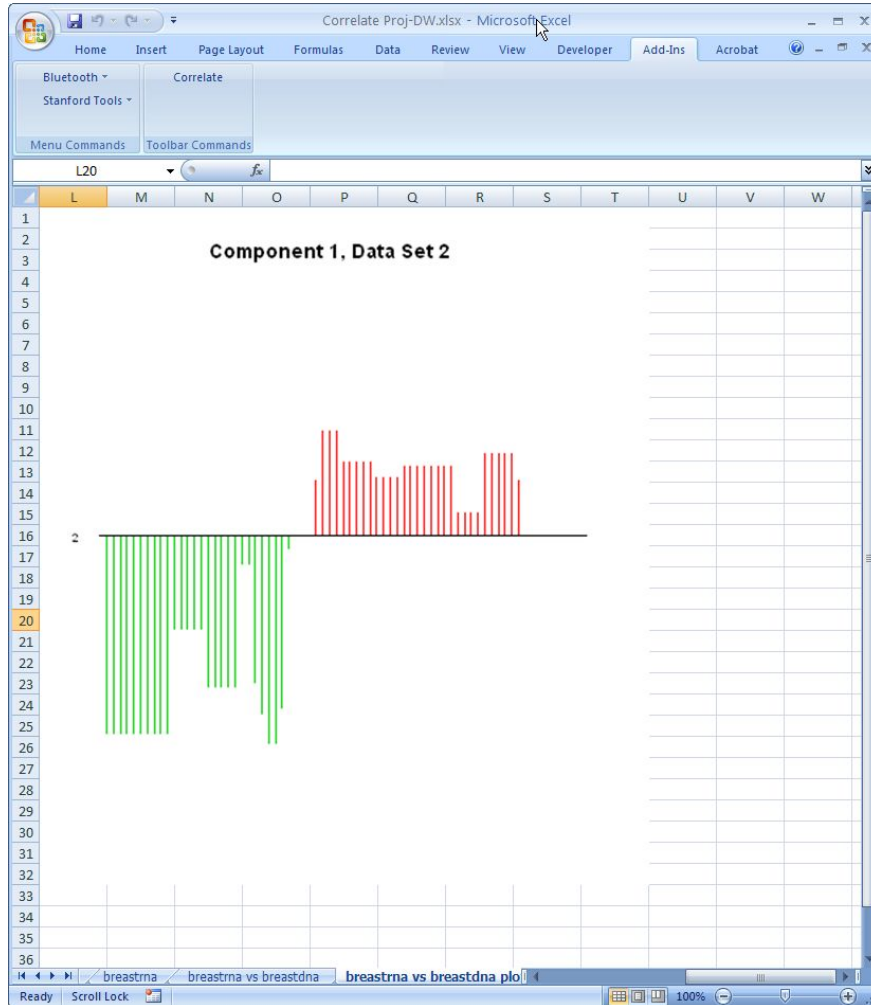


Figure 13: The data1 vs data 2 plots worksheet: A sample factor plot.

The order of the weights in the factor plot is the same as the order of the variables in the original data worksheet (and should correspond to some sort of spatial location, since type Ordered was used for the analysis).

11.3 Next steps

After the output figures have been examined and a final choice of tuning parameter values has been made, return to the `Run Form` and run `Correlate` using the `Manual` tuning parameter option and the chosen tuning parameter values. This time, use a large number of permutations (50 or 100 permutations will give meaningful p-values). This gives a final list of weights for each gene and a p-value that quantifies whether a correlation as high as what was observed on this data would have been expected by chance.

12 A typical analysis for `Correlate`

A typical analysis for `Correlate` might proceed as follows:

1. Load in the data.
2. Run `Correlate` using automatic tuning parameter selection and a small number of permutations (say, 5).
3. If one or both data sets is `Standard`, inspect the resulting permutation plots in order to choose a tuning parameter that gives a large z-statistic and has a desirable number of non-zero weights. If one or both data sets is `Ordered` then inspect the factor plot(s) output in order to determine if a larger or smaller tuning parameter value is desired. (Recall that for `Standard`, a larger tuning parameter yields more non-zero weights, whereas for `Ordered`, a larger tuning parameter yields fewer non-zero weights.)
4. Run `Correlate` using the tuning parameters chosen in the previous step (`Manual` setting), and a large number of permutations in order to get a meaningful p-value (at least 20, but 50 or 100 permutations would be better).

13 Some more ideas for using `Correlate`

Here are some ideas for how `Correlate` can be run in the case of gene expression data (data set 1) and DNA copy number data (data set 2):

1. Run `Correlate` using all of the genes in data set 1 (type `Standard`), against all of the probes on chromosome k for data set 2 (type `Ordered`). Repeat for all k . One can identify genes whose expression is correlated with copy number change on chromosome k .
2. Run it using all of the genes in data set 1 that *are not located* on chromosome k (type `Standard`), and all of the probes in data set 2 that *are located* on chromosome k (type `Ordered`; be sure to use a blocking variable). Repeat for all k . One can identify genes not

on chromosome k whose expression is correlated with copy number change on chromosome k - these might be *trans* effects.

Some other ideas are as follows:

1. Let data set 1 be gene expression measurements on chromosome k and let data set 2 be gene expression measurements on chromosome l . Run `Correlate` with both data sets of type `Standard` in order to find co-regulated sets of genes on different chromosomes.
2. Let data set 1 be DNA copy number measurements on chromosome k and let data set 2 be DNA copy number measurements on chromosome l . Run `Correlate` with both data sets of type `Ordered` in order to identify regions of copy number change on chromosome k that are correlated with regions of copy number change on chromosome l . Are significant correlations found (is the p-value small?)

14 Technical details of the `Correlate` procedure

14.1 Imputation of missing data

When the data is loaded in, missing data is imputed using K-nearest neighbors [2]. For more information, refer to that paper or to Section 9 of this document.

14.2 Sparse canonical correlation analysis

Suppose that we wish to run `Correlate` on two data sets with measurements (features) on n samples. The first data set consists of p_1 feature and the second data set consists of p_2 features. `Correlate` first standardizes the features in each data set to have mean zero and standard deviation one. Let \mathbf{X}_1 denote the $n \times p_1$ data matrix of data set 1 with standardized features, and let \mathbf{X}_2 denote the corresponding matrix for data set 2.

Sparse canonical correlation analysis [6] seeks vectors $\mathbf{u}_1 \in \mathbb{R}^{p_1}$ and $\mathbf{u}_2 \in \mathbb{R}^{p_2}$ that maximize

$$\mathbf{u}_1^T \mathbf{X}_1^T \mathbf{X}_2 \mathbf{u}_2 \text{ subject to } P_1(\mathbf{u}_1) \leq c_1, P_2(\mathbf{u}_2) \leq c_2, \|\mathbf{u}_1\|^2 \leq 1, \|\mathbf{u}_2\|^2 \leq 1, \quad (14.1)$$

where c_1 and c_2 are tuning parameters and P_1 and P_2 are penalties on the elements of \mathbf{u}_1 and \mathbf{u}_2 . The form of the penalties P_1 and P_2 depend on the types of the data sets 1 and 2 (`Standard`, `Ordered`, or `Unpenalized`).

In the equation (14.1), \mathbf{u}_1 and \mathbf{u}_2 are *weight vectors* that define a linear combination of features in \mathbf{X}_1 that is correlated with a linear combination of features in \mathbf{X}_2 . Elements of \mathbf{u}_1 and \mathbf{u}_2 that equal zero indicate features in \mathbf{X}_1 and \mathbf{X}_2 that are not involved in the linear combinations. If sparse CCA is applied to DNA copy number data (data set 1) and gene expression data (data set 2) then the results can be interpreted as *this linear combination of copy number changes is correlated with*

this linear combination of gene expression values. This allows the scientist to identify genomic regions that regulate gene expression, etc.

14.2.1 Standard data sets

Suppose that data set 1 has many features - that is, p_1 is large relative to n - and we want each feature in data set 1 to be considered individually. This is the `Standard` setting that will be used for *SNP data, gene expression data, and most other types of data*. In this case, the penalty P_1 is an L_1 , or lasso [3], penalty: the sparse CCA criterion becomes

$$\mathbf{u}_1^T \mathbf{X}_1^T \mathbf{X}_2 \mathbf{u}_2 \text{ subject to } \|\mathbf{u}_1\|_1 \leq c_1 \sqrt{p_1}, \|\mathbf{u}_2\|_1 \leq c_2 \sqrt{p_2}, \|\mathbf{u}_1\|^2 \leq 1, \|\mathbf{u}_2\|^2 \leq 1, \quad (14.2)$$

where the notation $\|\mathbf{u}_1\|_1$ indicates the sum of the absolute values of the elements in \mathbf{u}_1 . When the tuning parameter c_1 is small, many of the elements of \mathbf{u}_1 will be exactly equal to zero, and so the resulting linear combination can be interpreted as a weighted average of a small subset of the features in data set 1. The tuning parameters c_1 and c_2 are required to be between 0 and 1.

When data set 1 is `Standard` the user has the option to constrain the sign of the elements of \mathbf{u}_1 to be `Positive` or `negative`. In this case, (14.2) is modified to include an additional constraint $u_{i1} \geq 0$ or $u_{i1} \leq 0$.

14.2.2 Ordered data sets

Now, suppose that data set 1 has many features - p_1 is large relative to n - and the features have some sort of spatial ordering such that we expect adjacent features to be correlated and want them to have similar weights in the linear combination. For instance, if data set 1 consists of *DNA copy number data* then we would like to find a *region of copy number change* that is correlated with the features in data set 2. In particular, we want nearby copy number probes to be given similar weights by sparse CCA. To do this, we run `Correlate` with data set 1 of type `Ordered`, which results in a `fused lasso` penalty being imposed on the elements of \mathbf{u}_1 [4, 5]. The fused lasso penalty takes the form

$$P_1(\mathbf{u}_1) = \|\mathbf{u}_1\|_1 + \sum_{i=2}^{p_1} |u_{i1} - u_{(i-1)1}| \quad (14.3)$$

and encourages sparsity (many weights exactly equal to zero) and smoothness (adjacent weights have similar values) in the weight vector \mathbf{u}_1 . When the data is of type `Ordered`, it is important to use the `Blocking` variable (Section 8) in order to indicate that smoothness should not extend between blocks. For instance, in the context of DNA copy number data, the blocking variable can indicate the chromosome number, so that `Correlate` does not require that the weights be smooth between the end of one chromosome and the beginning of the next. Also, it is most important that within a block, the features be listed in the data worksheet in the *correct spatial ordering* since this is the only information that `Correlate` uses in order to determine the feature order.

When a data set is `Ordered`, the criterion used to perform sparse CCA isn't exactly of the form (14.1). Details are given in [6], but the important point is that the Lagrange form rather than the bound form of the fused lasso penalty is used, and so *a large value of the tuning parameter corresponds to a lot of sparsity and smoothness, whereas a small value corresponds to less sparsity and less smoothness* in the weights. (On the other hand, with type `Standard`, a small tuning parameter results in more sparsity and a large tuning parameter results in less sparsity.) Also, unlike the `Standard` case, the `Ordered` tuning parameters will tend to take on very small positive values (whereas the `Standard` tuning parameters will always range from 0 to 1).

Note that running `Correlate` with the data type `Ordered` for a given tuning parameter is much more computationally intensive than running it using `Standard`. However, see Section 14.3 for more information about automatic tuning parameter selection for both methods, which greatly affects the run time of `Correlate`.

14.2.3 Unpenalized data sets

If data set 1 has few features - p_1 is small relative to n - then `Correlate` can be run using the `Unpenalized` option for data set 1. In this case, the P_1 penalty to the elements of \mathbf{u}_1 simply will not be applied, and the optimization criterion (14.1) becomes

$$\mathbf{u}_1^T \mathbf{X}_1^T \mathbf{X}_2 \mathbf{u}_2 \text{ subject to } P_2(\mathbf{u}_2) \leq c_2, \|\mathbf{u}_1\|^2 \leq 1, \|\mathbf{u}_2\|^2 \leq 1. \quad (14.4)$$

In this case, all elements of \mathbf{u}_1 will be non-zero - that is, each feature in \mathbf{X}_1 will be involved in the linear combination found. The `Unpenalized` option can be useful if the features in \mathbf{X}_1 are clinical measurements that one wishes to cross-correlate with the features in \mathbf{X}_2 .

When data set 1 is `Unpenalized` the user has the option to constrain the sign of the elements of \mathbf{u}_1 to be positive or negative. In this case, (14.4) is modified to include an additional constraint $u_{i1} \geq 0$ or $u_{i1} \leq 0$.

14.3 Selection of Tuning Parameters and Computation of P-values

Once the types of data sets 1 and 2 have been chosen, the penalty tuning parameters c_1 and c_2 in (14.1) must be selected. (However, if a data set is of type `Unpenalized` then no tuning parameter is required for that data set, so this discussion of tuning parameter selection applies only to the `Standard` and `Ordered` cases.) If the user knows which tuning parameters he or she would like to use, then he can select the `Manual` radio button in the `Run Form` and can enter the tuning parameter into the box. However, in most cases, one will opt for automatic tuning parameter selection. The automatic tuning parameter selection method is as follows:

1. If a data set is of type `Ordered` and automatic tuning parameter selection is requested, then the tuning parameter for that data set is chosen by inspection of the first eigenvector of the matrix $\mathbf{X}_1^T \mathbf{X}_2$. In particular, automatic selection of the tuning parameter for `Ordered` data

sets is *very fast* because it *does not require permutations* to be performed. (However, if the other data set is of type `Standard` and automatic tuning parameter selection for that data set is required, then permutations will be performed. Also, running a large data set of type `Ordered` can be quite time-consuming even without performing permutations.)

2. If a data set is of type `Standard` and automatic tuning parameter selection for that data set is requested, then the tuning parameter for that data set is chosen using a permutation approach, described below. A grid of ten tuning parameters is considered, and five permutations are performed by default. We recommend running 50 or so permutations for any final analyses. Automatic tuning parameter selection for `Standard` data sets can be quite time-consuming since it entails performing sparse CCA 10 times for each permutation performed. A progress bar will pop up to indicate the fraction of permutations that have been completed.

The algorithm for automatic tuning parameter selection for `Standard` data sets is as follows.

Automatic tuning parameter selection for `Standard` data set(s):

1. A grid of ten candidate tuning parameter values between 0 and 1 is created.
2. Sparse CCA is run ten times, once for each tuning parameter value in the grid. If the data consist of two `Standard` data sets for which tuning parameters must be chosen, then at each point c in the grid, sparse CCA is performed with both tuning parameters equal to c . If only one data set of type `Standard` requires tuning parameter selection, then at each point c in the grid, sparse CCA is performed with the tuning parameter that needs to be automatically selected equal to c . Record $\text{Cor}(\mathbf{X}_1\mathbf{u}_1, \mathbf{X}_2\mathbf{u}_2)$ for each tuning parameter value considered.
3. For $i = 1, \dots, k$, where k is the number of permutations requested (default is 5, but we recommend running a larger k for final analyses):
 - (a) Permute the samples in \mathbf{X}_1 in order to obtain a permuted matrix \mathbf{X}_1^* .
 - (b) Sparse CCA is run ten times on data $(\mathbf{X}_1^*, \mathbf{X}_2)$, once for each tuning parameter value in the grid. Record $\text{Cor}(\mathbf{X}_1^*\mathbf{u}_1, \mathbf{X}_2\mathbf{u}_2)$ for each tuning parameter value considered.
4. For each tuning parameter value, compute the following:
 - (a) Compute the fraction of permuted data sets that have a correlation that is greater than or equal to the correlation obtained on the real data. This fraction is the *p-value* for that tuning parameter value. Note that the only possible p-values are $0, \frac{1}{k}, \frac{2}{k}, \dots, \frac{k-1}{k}, 1$ and so it is important to use a large number of permutations k in order to get accurate p-values.

- (b) Compute a z-statistic comparing the correlation obtained on the real data set to the correlations obtained on the permuted data. To do this, first apply a Fisher transformation to each correlation r : $f(r) = \frac{1}{2} \log \frac{1+r}{1-r}$. Let $f(r)$ denote the transformed correlation on the real data and let $\{f(r_i^*)\}_{i=1}^k$ denote the transformed correlations on the permuted data. Then, the z-statistic is given by $z = \frac{f(r) - \frac{1}{k} \sum_{i=1}^k f(r_i^*)}{\text{sd}(f(r_i^*))}$.
5. Then, the optimal tuning parameter value corresponds to the largest z-statistic, and the p-value for that tuning parameter value is a measure of how high the correlation obtained is relative to what one would expect if the features in \mathbf{X}_1 and \mathbf{X}_2 are not truly correlated with each other.

Now, one might want to obtain a p-value for the correlation obtained even if permutations are not being performed for tuning parameter selection. For instance, if the two data sets are of type `Ordered` or `Unpenalized`, or of type `Standard` with the tuning parameter specified, then a p-value will not be output automatically - but one might want one anyway. In this case, the number of permutations to be performed can, as usual, be specified in the `RUN FORM`. In this case, the permutations will run much more quickly since the computations will be run only for a single pair of tuning parameters, rather than over an entire grid of tuning parameters. In this case, the p-value is computed as follows.

Computation of p-values when tuning parameters are specified:

1. Perform sparse CCA using the specified tuning parameters.
2. For $i = 1, \dots, k$, where k is the number of permutations requested (default is 5, but we recommend running a larger k (on the order of at least 50 or 100) in order to get meaningful p-values:
 - (a) Permute the samples in \mathbf{X}_1 in order to obtain a permuted matrix \mathbf{X}_1^* .
 - (b) Sparse CCA is run on data $(\mathbf{X}_1^*, \mathbf{X}_2)$, with the specified tuning parameters. Record $\text{Cor}(\mathbf{X}_1^* \mathbf{u}_1, \mathbf{X}_2 \mathbf{u}_2)$.
3. Compute the fraction of permuted data sets that have a correlation that is greater than or equal to the correlation obtained on the real data. This fraction is the *p-value* for that tuning parameter value. Note that the only possible p-values are $0, \frac{1}{k}, \frac{2}{k}, \dots, \frac{k-1}{k}, 1$ and so it is important to use a large number of permutations k to get accurate p-values.

14.4 Obtaining multiple sets of feature weights

Equation 14.2 gave the criterion for finding a single set of feature weights. To obtain a second set of feature weights, we notice that the criterion (14.1) can be thought of as maximizing

$$\mathbf{u}_1^T \mathbf{Z}_1 \mathbf{u}_2 \text{ subject to } P_1(\mathbf{u}_1) \leq c_1, P_2(\mathbf{u}_2) \leq c_2, \|\mathbf{u}_1\|^2 \leq 1, \|\mathbf{u}_2\|^2 \leq 1, \quad (14.5)$$

where $\mathbf{Z}_1 = \mathbf{X}_1^T \mathbf{X}_2$, a $p_1 \times p_2$ data matrix. We compute $\mathbf{Z}_2 = \mathbf{Z}_1 - d \mathbf{u}_1 \mathbf{u}_2^T$ where $d = \mathbf{u}_1^T \mathbf{Z}_1 \mathbf{u}_2$ and perform the optimization problem (14.5), this time using \mathbf{Z}_2 as the data instead of \mathbf{Z}_1 . The resulting factors are the second set of feature weights. This process of subtracting out the current set of feature weights and performing sparse CCA on the residuals can be repeated to obtain as many sets of feature weights as desired.

15 Frequently Asked Questions

15.1 General Questions

1. Is there a version of `Correlate` that works on Macintosh computers?

Since the Excel version of `Correlate` makes extensive use of Microsoft Component architecture on Windows (COM), it is not easy to port directly to a Mac.

However one can run `Correlate` on a MAC using a PC emulator such as Parallels. We have had good experience with that configuration.

15.2 `Correlate` Usage Questions

1. `Correlate` generates an error when I run it on my dataset. What should I do?

Most often, errors are due to improper data formats.

- Please make sure that your data is formatted exactly as described in Section 8 and is loaded in following the instructions in Section 10.1. In particular, be sure that the sample labels match up and that the data begins in the fourth column (if a blocking column is present) or in the third column (if not present).
- Is there a feature with only one or zero *non-missing* values? If so, the imputation will fail.
- Are there any features with 0 standard deviation? If so then sparse CCA will fail.

Sometimes `Correlate` will run out of memory, especially if the dataset is large. `Correlate` can run very slowly on large data sets, especially if many permutations are requested. Also, note that computations for the `Ordered` data type are much more demanding than for `Unpenalized` or `Standard`.

2. Why does the random number seed stay the same? Can you not generate a new seed automatically?

The random number seed allows one to reproduce an analysis. However, if one uses the default seed for every analysis, then the *same sequence of permutations* are generated. This is not always desirable. It would appear that generating a seed randomly using the clock or some such mechanism without bothering the user for input might be better. Not necessarily. If reproducibility is important, then asking the user to set the seed is preferable so that any analysis can be rerun to confirm results. We have come down on the side of reproducibility. The user can change the random seed in the `Run Form` (Section 10.2). Please also note that the random number generator seed used in any analysis is always listed in the output to ensure reproducibility of results.

3. How large a dataset can `Correlate` handle?

There is really no hard limit *per se* in `Correlate`. Excel itself has some limit on the number of rows and columns it can handle. There are additional overheads involved in marshalling the data between Excel and the core of `Correlate`. Therefore, the practical limit is lower. In general, the more memory you have, the larger problems you can handle.

However, when permutations are being performed and the data sets are large, `Correlate` may take hours to run (easily). In this case, we recommend doing something else while it is running...

4. Where is the `Correlate` manual?
5. Where can I go for help if I just cannot get `Correlate` to work?

We are very interested in making `Correlate` work for all users. However, before reporting problems or bugs, we'd really like you to make sure that the problem is really with `Correlate`. The following checklist should help.

- Please make sure you have installed all the prerequisites. See section 3.
- If the problem is with `Correlate` usage, please make sure that you have formatted your data exactly as mentioned in the `Correlate` manual.
- If you are having problem on a particular type of data, please make sure that you have formatted the response labels appropriately and have chosen the correct applicable data type.

If you still cannot get `Correlate` to work, send email to `correlate-bug@stat.stanford.edu` with complete details including

- (a) The error message

- (b) The system you are using (Windows 2000, Windows XP Home, Windows XP Pro)
- (c) The version of R you are using
- (d) The dataset you used that generated the error.

References

- [1] K. Chin, S. DeVries, J. Fridlyand, P.T. Spellman, R. Roydasgupta, W.-L. Kuo, A. Lapuk, R.M. Neve, Z. Qian, T. Ryder, F. Chen, H. Feiler, T. Tokuyasu, C. Kingsley, S. Dairkee, Z. Meng, K. Chew, D. Pinkel, A. Jain, B.M. Ljung, L. Esserman, D.G. Albertson, F.M. Waldman, and J.W. Gray. Genomic and transcriptional aberrations linked to breast cancer pathophysiologies. *Cancer Cell*, 10:529–541, 2006.
- [2] T. Hastie, O. Alter, G. Sherlock, M. Eisen, R. Tibshirani, D. Botstein, and P. Brown. Imputation of missing values in dna microarrays. Technical report, 1999. Working draft.
- [3] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc. B.*, 58:267–288, 1996.
- [4] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. Royal. Statist. Soc. B.*, 67:91–108, 2005.
- [5] R. Tibshirani and P. Wang. Spatial smoothing and hotspot detection for CGH data using the fused lasso. *Biostatistics*, 9:18–29, 2008.
- [6] D.M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.