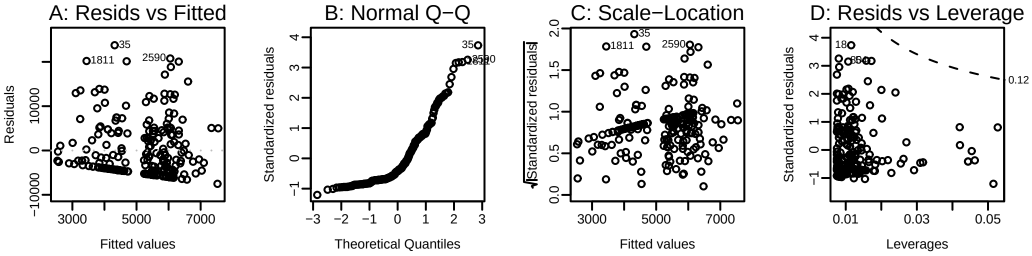

Figure 11.8: Diagnostic plots for the model (`A.lm`) that uses the scores as the only covariate.

11.5 *High-dimensional data, classification, and plots

Data sets that have many more variables than observations are now common in a number of application areas. In the data that will be considered in this section, the observations are divided up into groups, just as for the `possum` data in Subsection 11.2.4.

In Subsection 11.2.4 the groups were sites at which the possums were taken. For the data that will be discussed here, conveniently called the “Golub” data after the first author of the paper that was based on it (Golub et al., 1999), the observations are tissue samples that have been taken from cancer patients, grouped into three different cancer types. For the possum data, the 9 variables were body measurements. For the Golub data, there are 7129 variables, each an attempt to measure the biological activity of a particular gene. Technically, the measurements are “gene expression indices”.

The following table compares the relative number of observations and variables in the `possum` data with the numbers in the data that will now be examined:

		Number of	
Data source	Type of measurement	observations	variables
Possum data	Body dimension	102 (1 missing)	9
Golub cancer tissue data	Gene expression index	72	7129

For the possum data, the only groupings of interest were by `sex` and by `site`. For the Golub data, there are several groupings that are relevant to any examination of the data. A major interest is in finding a discrimination rule that can discriminate between different cancer types: AML, ALL B-type and ALL T-type. Here AML is Acute Myoblastic Leukemia (myoblastic = producing muscle tissue), while ALL is Acute Lymphoblastic Leukemia (lymphoblastic = producing lymph tissue). An effective discrimination rule, using a small subset of the features, would allow the design of a diagnostic device (a “probe”) that could be used to determine cancer type. (Note however that any classification of cancers is likely to conceal large individual differences that, in many cancers, arise from random differences in the timing and outcome of trigger points in a cascade of genomic damage and disruption.)

Use of these data for discrimination between cancer types is complicated by the potential effects of other factors. As well as different sexes, there are two different types of body tissue (bone marrow and peripheral blood). Another potential source of variation is that the tissues came from four different hospitals; this will not be pursued here. Because of the complications that these other factors create for the attempt to find a rule that will be effective in classifying new samples, graphical exploration of the data is more than otherwise important. The high dimensionality offers however a huge challenge for creating graphs that are both revealing and unlikely to mislead. What views of the data may help in revealing subgroups in the data, or points that may have been misclassified, or between group differences in the variance-covariance structure?

In the discussion that follows, variables will be called features, following the terminology that is common in this area. Each of the 7129 features (or variables) is a value of an expression index, measuring the expression of one of 7129 genes or putative genes.

For further discussion of the types of analysis that are described here, see Maindonald and Burden (2005) and Ambroise and McLachlan (2002).

What groups are of interest?

The data frame `golub.info` has information on the tissue samples, which are the observations. The two classifications that will be pursued are (1) according to cancer type (AML, ALL B-type, ALL T-type), given by the factor `cancer`, and (2) according to sex and tissue type, given by the factor `tissue.mf`. The frequencies in the two-way classification according to these factors are:

```
> with(golub.info, table(cancer, tissue.mf))
      tissue.mf
cancer BM:NA BM:f BM:m PB:NA PB:f PB:m
  allB   4    19   10    2     1    2
  allT   0     0    8    0     0    1
  aml   16     2    3    1     1    2
```

The different tissue/sex combinations may well affect the comparison between different cancer types. Thus the `allB` is predominately `BM:f`, while `aml` is predominately `BM` of unknown sex. For `allB`, there are enough samples that it is worthwhile to investigate the split of `tissue.mf` into `BM:f`, `BM:m` and `PB:m`. Thus, for these data, a sensible preliminary analysis is to investigate whether the tissue type (`BM` or `PB`) and gender (`Female` or `Male`) affects expression values, limiting attention to `allB` tissues. The following preliminary calculations separate out the relevant subset (`GolubB`) of the data, and derive a factor (`tissue.mfB`) that identifies the groups of interest (`BM:f`, `BM:m` and `PB:m`):

```
attach(golub.info)
## Identify allB samples for that are BM:f or BM:m or PB:m
subsetB <- cancer=="allB" & tissue.mf%in%c("BM:f","BM:m","PB:m")
## Form vector that identifies these as BM:f or BM:m or PB:m
tissue.mfB <- factor(tissue.mf[subsetB])
## Separate off the relevant columns of the matrix Golub
GolubB <- Golub[, subsetB]
detach(golub.info)
```

The vector `tissue.mf[subsetB)` is a factor that retains all the levels of `tissue.mf`. Use of the function textttfactor() returns a factor that has only the levels that are present in the data.

11.5.1 Classifications and associated graphs

In the following discussion, interest will be on the graphical view that can be associated with one or other discriminant rule, rather than in the discriminant rules themselves. The idea is to obtain graphs that are targeted toward giving a visual representation of classifications of interest. Different classifications will lead to different graphical views – what is seen depends, inevitably, on which clues are pursued in the search for relevant graphical views. Care is required to ensure that graphs present a fair view of the data, not showing differences between known groups where there are none or exaggerating such differences as may exist.

Discrimination will use the relatively simple and readily understood linear discriminant function methodology that was introduced and used earlier, in Section 11.2.2. Analyses will, again, use the `lda()` function (*MASS*). Linear discriminant functions may be as complicate as is sensible, given the large amount of noise in current expression array data sets.

The statistical information given in the output from the function `lda()` assumes that the variance-covariance matrix is the same in all groups. Even where this is not plausible, a useful graphical view may still result. If there are pronounced between group differences in the variance-covariance structure, this will be obvious in the graphs that result. The emphasis will be on insight rather than on the use of methods that are arguably optimal.

The function `qda()` is an alternative to `qda()` that allows for different variance-covariance matrices in different groups. Use of `qda()` does however restrict the number of features that can be used. For use of p features, each group must have at least $p + 1$ observations. The methodology for deriving plots that will be described here does not readily adapt for use with `qda()`.

Preliminary data manipulation

The version of the Golub data that is used here, and the functions used, are available from `http://www.maths.anu.edu.au/~johnm/r/cvplot`.) These will shortly be incorporated into a package that has the name *hddplot*. The data were derived from the *golubEsets* package that is available from the Bioconductor web site, with further processing applied that is additional to the processing that preceded the incorporation of the data into the `golubEsets` package.

All data for an observation (a tissue sample) comes from a single "chip", leading potentially to systematic differences between observations. It is therefore usual to apply procedures that align the feature values for the different observations. For the present data, the procedure has been to ensure that the median and standard deviation is the same across the different slides. Full details will be included in an Sweave file that will be placed on the site noted above.

Before proceeding further the distribution for individual observations across features, and the distribution for a selection of features across observations, should be checked.[6]The

[6]
```
## Try e.g.
boxplot(data.frame(GolubB[, 1:20]))  # First 20 columns (observations)
## Random selection of 20 rows (features)
boxplot(data.frame(GolubB[sample(1:7129, 20), ]))
```

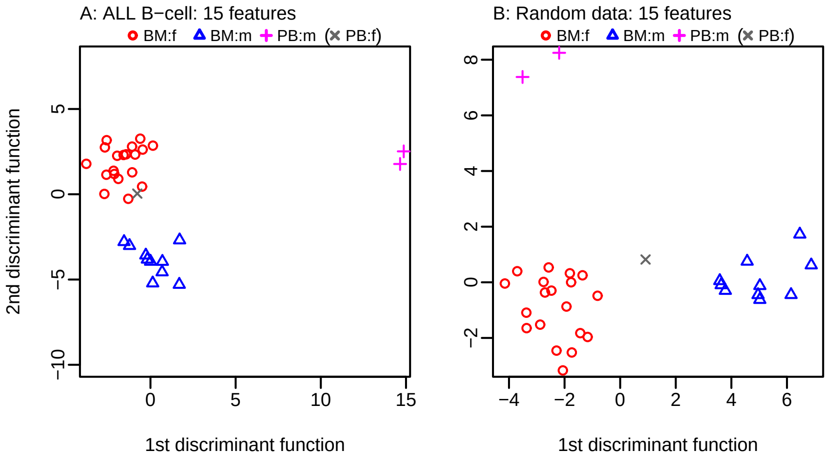

Figure 11.9: The left panel used the subset of ALL B-cell observations for which `Gender` was known. The one `PB:f` observation was excluded for purposes of analysis. An anova F-statistic calculation identified the 15 features that, individually, “best” separated the data into three groups. These 15 features were then used in a linear discriminant analysis. Scores were then determined for each of the two available discriminant axes. Additionally, a predicted score was determined for the `PB:f` observation. For the right panel, the same procedure was followed, but now using a matrix where the “expression values” were random normal data.

distributions are positively skewed. The effect on results requires investigation.

11.5.2 Flawed graphs

Figure 11.9A is a first, but flawed, attempt at a graph that shows the separation of the 31 observations into the 3 specified groups. It uses discriminant axes that were determined using 15 features that individually, as indicated by an analysis of variance F-test, gave the best separation into the 3 groups. It is flawed because no account is taken of the effect of selecting the 14 “best” features, out of 7129. Figure 11.9B, which was obtained by applying the same procedure to random normal data, from 7129 independent normal variables that all has the same mean and variance, shows the potential for getting an entirely spurious separation into groups. In spite of its evident flaws, it is important to understand the procedure that was followed, as the later discussion will extend and adapt it to give graphs that are not subject to the same flaws.

In summary, Figure 11.9 was obtained as follows:

- The 15 features (from 7129) were selected that, as measured by an analysis of variance F-statistic, gave the best separation of the remaining 40 observations into the three groups `BM:f`, `BM:m`, `PB:f`;[7]
- The first two discriminant functions, and associated discriminant scores, were calculated and plotted;

[7]
```
## Uses order.features() (DAAG); see below
ord15 <- order.features(GolubB, cl=tissue.mfB)[1:15]
```

- Discriminant scores were predicted for the one `PB:f`, so that it could be plotted.[8]

Figure 11.9B used the same approach, but replaced the measured expression values by random normal values. It can be obtained with:

```
simscores <- simulate.scores(nfeatures=7129, cl=rep(1:3, c(19,10,2)),
                             cl.other=4, nf.use=15, seed=41)
  # Returns list elements: scores, cl, scores.other & cl.other
with(simscores, scoreplot(scores, cl, scores.other, cl.other))
```

Readers are encouraged to experiment with this function, trying different values for `nfeatures`, for `nf.use`, and different groupings of the data.[9]

The selection of 15 features from a total of 7129, selected to have the largest F-statistics, makes it impossible to give much credence to the clear separation achieved with expression array data in Figure 11.9A. The extent of separation in Figure 11.9B from use of random normal data indicates the potential severity of the selection effect, for the data used for 11.9A. The 15 most extreme F-statistics out of 7129, from a null distribution in which there is no separation between groups, will all individually show some separation. Choice of the two “best” two discriminant axes that use these 15 features will achieve even clearer separation than is possible with any of the features individually. Clearer apparent separation still, both for random data and for the Golub data, can be achieved by choosing more than 15 features.

Distributional extremes

There will in some data sets be a small number of F-statistics that are so large that they are unlikely to be extremes from the null distribution. Plots such as Figure 11.9A can then be based on these features, with no concern about possible effects of selection bias. Correlations between features vitiates use of a theory that assumes that genes are independent. Additionally, the distributions for individual features may not be normal, in a context where the interest is in the distributional extremes of F-statistics and normality is likely to matter.

Permutation methods, implemented in the package *multtest*, can however be used to determine a relevant reference distribution. (There are two ways to install this package.

[8]
```
dfB.ord <- data.frame(t(GolubB[ord15, ]))
## Calculations for the left panel
## Transpose to observations by features
dfB15 <- data.frame(t(GolubB[ord15, ]))
library(MASS)
dfB15.lda <-  lda(dfB15, grouping=tissue.mfB)
scores <- predict(dfB15.lda, dimen=2)$x
## Scores for the single PB:f observation
attach(golub.info)
df.PBf <- data.frame(t(Golub[ord15, tissue.mf=="PB:f"
                                    & cancer=="allB", drop=FALSE]))
scores.PBf <- predict(dfB15.lda, newdata=df.PBf, dimen=2)$x
detach(golub.info)
## Warning! The plot that now follows may be misleading!
scoreplot(scores=scores, cl=tissue.mfB, scores.other=scores.PBf,
          cl.other="PB:f")
```

[9]
```
## Use the function simulate.scores()
## Alternatively, set:
## dimen <- dim(GolubB); rsetB <- array(rnorm(prod(dimen)), dim=dimen)
## Then replace GolubB with rsetB, and repeat the lines above.
```

A "stand-alone" version can be installed from CRAN, or it can alternatively be installed as one component of a minimal BioConductor installation. The stand-alone version from CRAN is adequate for present purposes.)

The function `mt.maxT()` determines the needed empirical distribution, as part of its implementation of a multiple testing procedure – the "step-down" method. Our interest here is not in the multiple testing procedure, but in the empirical distribution of the F-statistics, which can be obtained as`qf(1-GolubB.maxT$rawp, 2, 28)`. The F-statistics for the assignment of labels as in the actual sample are stored in `GolubB.maxT$teststat`, i.e., the values are the same as those obtained from `calc.matrixrows.f(GolubB, tissue.mfB)`.

The code used for the calculation is:

```
## The calculation may take tens of minutes, even with adequate
## memory (e.g., 512MB) and a fast processor.
## If necessary, use a smaller value of B.
GolubB.maxT <- mt.maxT(GolubB, unclass(tissue.mfB)-1, test="f",
                       B=100000)
## Compare calculated F-statistics with permutation distribution
qqthin(qf(1-GolubB.maxT$rawp, 2, 28), GolubB.maxT$teststat)
## Compare calculated F-statistics with theoretical F-distribution
qqthin(qf(ppoints(7129), 2, 28), GolubB.maxT$teststat)
  # The theoretical F-distribution gives estimates of quantiles
  # that are too small
## NB also (not included in Figure 11.10) the comparison between
## the permutation distribution and the theoretical F:
qqthin(qf(ppoints(7129), 2, 28), qf(1-GolubB.maxT$rawp, 2, 28))
```

The parameter `B` sets the number of permutations that will be taken. This needs to be substantially larger than the number of features in order to get an accurate estimate of extreme upper quantiles of the distribution.

Figure 11.10 shows QQ-plots that compare the relevant distributions.

The plot on the left, which uses an appropriate reference distribution, suggests that the largest two features, and probably others as well, show differential expression, The theoretical F-distribution, used for the horizontal axis in the right plot, is clearly not an appropriate reference distribution. It is very unlikely that the null distribution would generate such extreme F-statistics. It would be safe to work with a version of Figure 11.9A that uses the two features that show the clearest evidence of differential expression. Beyond, this point, there are a number of features that show evidence of differential expression, and use of those that show the clearest evidence of differential expression may introduce a bias.

This brief interlude has had the aim of drawing attention to the distributional issues. Effective mechanisms for handling such issues are the subject of active research, and will be pursued further in this section.

The subsequent discussion will demonstrate adaptations of the procedure used for Figure 11.9A, designed to avoid its evident flaws and without necessarily limiting attention to a small number of features that show clear evidence of differential expression. Before proceeding to that discussion, brief details will be given of the function `order.features()` that was used above, and that will be used extensively in the re-

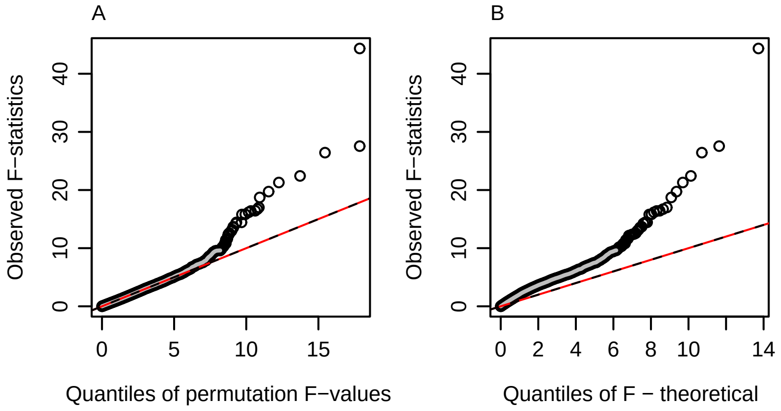

Figure 11.10: These QQ-plots are for the subset of ALL observations for which `Gender` was known, but excluding the one `PB:f` observation. The left plot compares the ordered F-statistics with the ordered statistics from the permutation distribution. The right plot compares the ordered F-statistics with F-distribution quantiles. Also shown, in both plots, is the line $y = x$.

mainder of this section.

The function `order.features()`

At this point, note our function `order.features()`, included in the `DAAG` package. This function will be used extensively in the sequel.[10]The function `order.features()` takes as parameters:

- `dset`: the matrix of expression values, in the features by observations layout that is usual in work with expression arrays;
- `cl`: a factor that classifies the observations;
- `subset`: if changed from its default (`NULL`), this identifies a subset of observations (columns of `dset`) that will be used for calculating the statistics.

Selection of features that discriminate well individually, in an analysis of variance F-statistic sense, is not guaranteed to select the features that will perform well in combination. It is akin to using, in a multiple regression, the variables that perform well when used as the only predictors. It may however be a reasonable strategy for use in an initial exploratory analysis, in the absence of a an obviously better alternative. Again, note that this section is intended to draw attention to important issues and ideas. The development of good variable selection methods, in the context considered in this section, is the subject of ongoing research.

[10]
```
## Simplified version. The DAAG version has an additional parameters
## subset (to extract a subset of observations)
"simple.order.features" <-
  function(dset, cl){
    ## Ensure that cl is a factor & has no redundant levels
    cl <- factor(cl)
    stat <- calc.matrixrows.f(dset, cl)
    order(-stat)     # Require largest first
  }
```

11.5.3 Accuracies and Scores for test data

One approach is to split the data into two sets, here named I and II. Then set I can be used to train discriminant functions and to determine discriminant scores for the test observations in set II. The key requirement is that the scores must relate to observations that are distinct from those used to develop the discriminant functions, and are free from the selection bias that affects the set I scores. The set II test data had no role in either the selection of features, or the determination of the discriminant functions and associated scores.

The process can then be reversed, with set II used for training and scores calculated for set I. Two plots are then available, the first of which shows scores for set II, and the second scores for set II. The two plots, conveniently identified as I/II and II/I, will use different features and different discriminant functions and cannot be simply superposed.

Because of the small number of observations (3) in the `PB:m` category, the data used for Figure 11.9A cannot satisfactorily be split into training and test data. We can however use this approach to examine the classification of the bone marrow (`BM.PB=="BM"`) samples into ALL B-cell, ALL T-cell, and AML. Two approaches that might be used to determine the optimum number of features, when developing a discriminant rule from set I, are:

- Use the predicted accuracies for set II.
- Use cross-validation on set I

Cross-validation can of course be used even if, as in the data used for Figure 11.9A, we do not have a set II. Cross-validation will be demonstrated later in this section.

The graphs now presented will work with the grouping of the 62 observations into ALL B-cell, ALL T-cell and AML categories. These will be split into training and test sets in which the relative numbers in the three groups are similar. The function `balanced.sample()` (*DAAG*) may conveniently be used for this purpose.

```
attach(golub.info)
Golub.BM <- Golub[, BM.PB=="BM"]
cancer.BM <- cancer[BM.PB=="BM"]
## Now split each of the BTM categories between two subsets
##  Uses balanced.sample(), from DAAG
gp.id <- balanced.sample(cancer.BM, nset=2, seed=31)
  # Set seed to allow exact reproduction of the results below
detach(golub.info)
```

Tabulating the division into two sets, we find:

```
> table(gp.id, cancer.BM)
     cancer.BM
gp.id allB allT aml
    1 17    4   11
    2 16    4   10
```

The maximum number of features for calculations using `lda()` with the set I data are 28 (= 17+4+11-3-1) for set I, and 26 for set II. Hence we will work with a maximum of 25 features, in each case. Steps in handling the calculations are:

1. Using set I as the training data, find the 25 features that, for a classification of the data

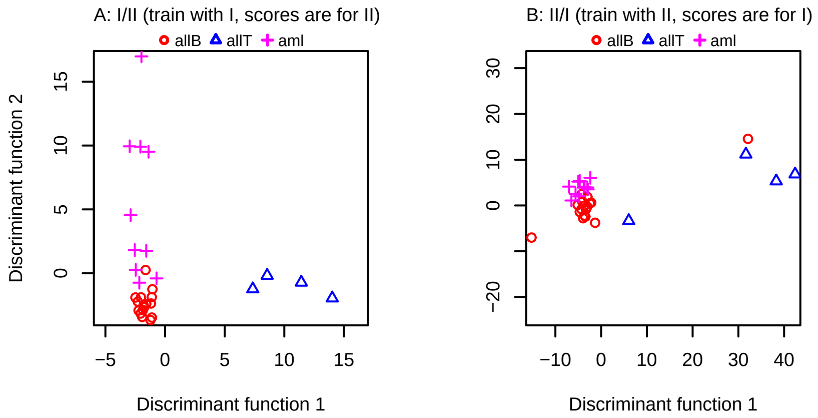

Figure 11.11: Panel A plots the test scores for the set II data in a calculation where the training data were set I (the I/II split), as described in the text. Panel B plots the test scores for the set I data in discriminant calculations where the roles of the two sets of the data were reversed i.e. the split was II/I.

into three groups according to levels of BTM, have the largest F-statistics.

2. For each value of $n_f = 1, 2, \ldots, 25$ in turn
   - Use the best n_f features to develop discriminant functions
   - Apply this function to the data in set II, and calculate the accuracy
3. The accuracies that result will be called the I/II accuracies.
4. Now make set II the training data and set I the test data, and repeat items 1 and 2. The accuracies that result will be called the II/I accuracies.

These calculations can be carried out using the function `acc.train.test()` (from *hddplot*).

```
> accboth <- acc.train.test(dset = Golub.BM, cl = cancer.BM,
                            traintest=gp.id)
```

```
Training/test split         Best accuracy, less 1SD   Best accuracy
I (training) / II (test)    0.89 (7 features)         0.93 (16 features)
II (training) / I (test)    0.77 (12 features)        0.81 (16 features)
```

Notice that, as well as giving the number of features that gives the maximum accuracy, the output gives the number which achieves the maximum accuracy, less one standard deviation. This gives a more conservative estimate of the optimum number of features. (The standard deviation is estimated as $p(1-p)/n$, where p is the estimated maximum accuracy, and n is the number of observations used for estimation of p.)

We now calculate both sets of test scores (I/II and II/I) for the more conservative choices of numbers of features (7 and 12 respectively), and plot the scores. The function `plot.train.test()` can conveniently be used for this purpose. Figure 11.11A shows the test scores for the I/II split, while Figure 11.11B shows the test scores for the

II/I split.[11] Readers should repeat the plots with other divisions of the data into training and test sets. To determine each new division, specify:

```
gp.id <- balanced.sample(cl=cancer.BM, nset=2, seed=NULL)
```

The graphs can vary greatly, depending on how the data have been split. The ALL T-cell points seem more dispersed than points for the other two categories.

It is instructive to compare the choices of features between I/II and II/I. The first twelve in the two cases are, by row number:

```
> rbind(accboth$sub1.2[1:12],accboth$sub2.1[1:12])
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
[1,] 6606 6510 4847 5542 4050 5543 4377 4342 6236  1694  3594  1268
[2,] 4050 2794 6696 4342 6510 1207 4055 2335 3252  6180  6236  5543
> ## Find the order of the first list in the second, if present
> match(accboth$sub1.2[1:12],accboth$sub2.1[1:12])
 [1] NA  5 NA NA  1 12 NA  4 11 NA NA NA
```

Note that the feature that is first in the first list is fifth in the second list, and that the feature that is first in the second list does not appear in the first list.

Cross-validation to determine the optimum number of features

We will demonstrate the use of cross-validation to determine how many features to choose. For the use of cross-validation, data are split into k sets. The cross-validation must be repeated for each choice of number of features that is under consideration.

Consider again the classification of a subset of the ALL B-cell Golub data for which gender is known into `BM:f`, `BM:m`, and `PB:m`, but omitting the one `PB:f` sample. There are 31 observations, divided into three groups, so that the maximum number of features that can be used for discrimination is 26 (=31-3-2). In order to choose the optimum number of features, the cross-validation must be repeated for each choice of g = number of features in the range 1, 2, ..., $g_{max} = 26$, calculating the cross-validation accuracy for each such choice. The number of features will be chosen to give an accuracy that is, or is close to, the maximum.

The full procedure is:

```
For $g$ = 1, 2, ..., $g_{max}$, do the following:
    For each fold $i = 1, \ldots k$ in turn ($k$ = number of folds) do the following:
        Split data: Take the $i$th set as the test data, and use the remaining data (all
            except the $i$th set) for training;
        Select: Choose the $g$ features that have the largest anova between group $F$-
            statistics;
        Classify: Determine discriminant functions, using the chosen features and
            the current training data;
        Prediction: Predict the group to which the current test observation belongs.
```

[11]
```
## Use function plot.train.test() from hddplot
plot.train.test(dset=Golub.BM, nfeatures=c(7,12),
                cl=cancer.BM, traintest=gp.id)
```

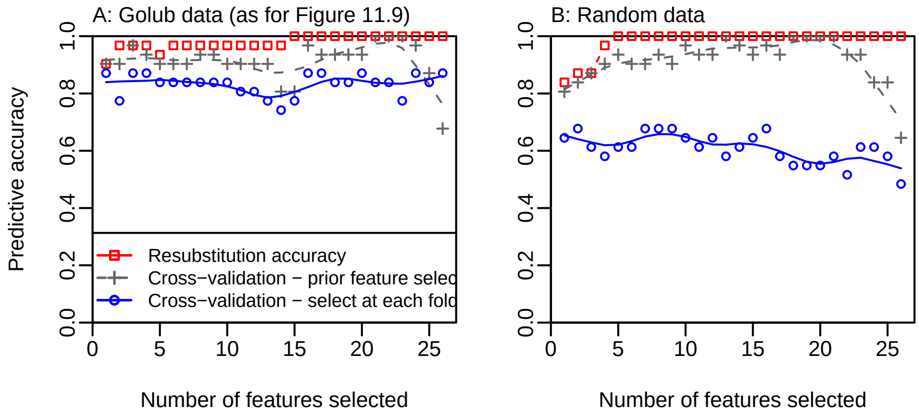

Figure 11.12: Comparison of different accuracy measures, in the development of a discriminant rule for the classification, into the categories `BM:f`, `BM:m` and `PB:m`, of the B-cell ALL data for which gender is known. The resubstitution measure (□) is a severely biased measure. Cross-validation, but with features selected using all the data (+), is less severely biased. An acceptable measure of predictive accuracy (○) requires re-selection of features at each fold of the cross-validation. The right panel shows the performance of each of these measures when the expression values are replaced by random data.

> Record, against the number g of features used, the proportion of correct predictions.

Accuracies are now available for all choices of number of features. Choose the smallest number of features that gives close to the maximum accuracy.

Leave-one-out cross-validation will be used in the subsequent discussion. This is slightly simpler to implement than general cross-validation, and the balancing of samples is automatic. It gives a more discrete view of the variability than is possible when the test set is a larger part of the total data.

Computations can be greatly reduced and simplified by determining the ordering of features in advance. This will use a matrix of character values, with as many columns as there are folds, and with the number of rows equal to the maximum number of features that will be considered for use. A rigid upper limit is the number of features that can be accommodated on the discriminant analysis, which is 36 (= 40 observations - 3 groups - 1). When the preliminary calculations are finished, the matrix will have stored, in column i, the 36 features that give the highest F-statistic for the observations that form the fold i training data. For selecting the "best" n_f features, one set for each different fold, the first n_f rows of this matrix will be used.

The key to getting the cross-validation to work correctly is the use, at each fold of the cross-validation, of the choice of features that is optimal for the training data at that fold. The blue curve in Figure 11.12A shows the result. Calculations are conveniently handled using the function `cvdisc()` (*hddplot*), thus:

```
tissue.mfB.acc <- cvdisc(GolubB, cl=tissue.mfB, nfeatures=1:26,
                         selectonce=TRUE, cv=TRUE)
```

```
  # Accuracy measures are cv: tissue.mfB.acc$acc.cv
  # Resubstitution (red points): tissue.mfB.acc$acc.resub
  # "Select once" (gray):  tissue.mfB.acc$acc.sel1
```

The red and gray points show biased and therefore inappropriate accuracy measures. The red points show the proportion of correct predictions when the discrimination rule is applied to the data used to develop the rule. The gray points show proportion of correct predictions when the same features, selected using all the data, are used at each fold, and do not change from one fold to the next.

Figure 11.12B applies the same calculations to random data. The bias in the red and gray curves is now very obvious. The grey points now do worse than chance for more than 3 or 4 features. At each fold, the rule has been tuned to be optimal for the training data. It therefore overfits, at each fold, relative to what is optimal for the one test observation.

Feature selection at each fold

Note the comparison between the simpler code that selects features initially prior to the cross-validation, and code that repeats feature selection at each fold of the cross-validation. The code for the calculation that selects features initially, prior to the cross-validation, is:

```
## This code does (less than) half the required task.
## It gives biased and therefore incorrect results.
maxfeatures <- 26
ord <- order.features(GolubB, source.mfB)
selectonce.df <- data.frame(t(GolubB[ord, , drop=F]))
acc.sel1 <- numeric(maxfeatures)
for(nf in 1:maxfeatures){
  hat.selB <- lda(tissue.mfB ~ .,
                  data=selectonce.df[, 1:nf, drop=FALSE],
                  CV=TRUE)$class
  tab1 <- table(hat.selB, tissue.mfB)
  acc.sel1[nf] <- sum(tab1[row(tab1)==col(tab1)])/sum(tab1)
}
```

For the correct use of cross-validation, the line that calculates `selectonce.df` disappears. The line that calculates `hat.sel1` expands into a `for` loop that is repeated once for each fold. Within each fold, there are steps akin to:

```
## Repeat for each fold of the cross-validation
  # Vector traini will be TRUE for training observations
  # Vector testi will be TRUE for test observations
  # cli <- tissue.mfB[traini]
  # ordi <- order.features(GolubB[, traini], cli)
  # dfi <- t(GolubB[ordi[1:ng], traini)
  # newdfi <- t(GolubB[ordi[1:ng], testi)
  # hati <- predict(lda(dfi, cli), newdata=newdfi)$class
```

Cross-validation: bone marrow (BM) samples only

. It turns out to be sufficient to calculate accuracies for 1, 2, . . . , 20 features.[12]The maximum is 92%, from use of 14 features. A more conservative assessment, based on the one standard deviation rule, suggests use of 11 features with an accuracy of 89%.

11.5.4 Graphs derived from the cross-validation process

With a methodology available for choosing the number of features, it is now possible to look for a better alternative to Figure 11.9A. Figure 11.12A suggested that the optimum number of features is, conservatively, either 1 or 3. The calculations that will be described here will use three features. This is more interesting, from the point of view of the methodology, than the use of a single feature. (Use of one feature allows just one discriminant axis, i.e. it does not lead to a scatterplot.)

It is of interest to see what features have been used at the different folds. This information is available from the list element `genelist`, in the object `tissue.mfB.acc` that was obtained earlier. As the interest is in working with three features, it is the first three rows that are relevant. The following table summarizes this information:

```
> tabf <- table(tissue.mfB.acc$genelist[1:3,])
> nam <- names(sort(-tabf))
> tab <- with(tissue.mfB.acc, table(genelist[1:3,],
+                                    row(genelist[1:3,])))
> tab[nam, ]
                  1  2  3
  M58459_at      30  0  1
  X54870_at       0 23  5
  U91327_at       0  4 23
  L08666_at       0  1  0
  S74221_at       0  1  0
  U29195_at       1  0  0
  U49395_at       0  1  0
  X00437_s_at     0  1  0
  X53416_at       0  0  1
  X62654_rna1_at  0  0  1
```

Observe that M58459_at is almost always the first choice.

Test scores can be calculated for the test data at each of the folds. However the different pairs of scores (pairs is all that is possible when there is are three groups) relate to different discriminant functions and to different choices of features, and are thus appropriately called “local” test scores. Local fold i training scores are similarly available.

These local training scores are used to make the connection between the different folds. Details of the way this is done are in the vignette that accompanies the package *hddplot*. The methodology is similar, but not identical to, that described in Maindonald and Burden (2005).

[12]
```
attach(golub.info)
BMonly.acc <- cvdisc(Golub, cl=cancer, nf.use=1:20, subset=BM.PB=="BM")
round(BMonly.acc$acc.cv, 2)
detach(golub.info)
```

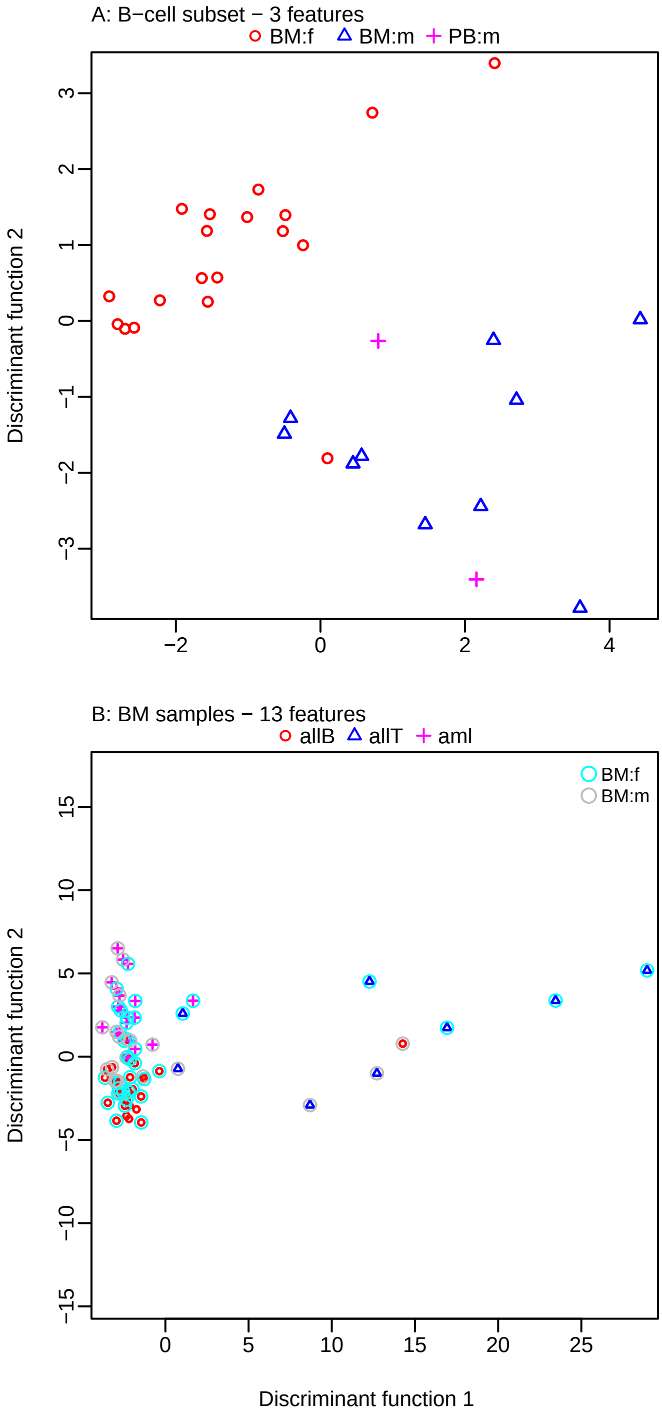

Figure 11.13: This plot is designed to fairly reflect the performance of a linear discriminant in distinguishing between the categories `BM:f`, `BM:m` and `PB:m`, from the ALL B-cell subset of the Golub data for which gender is known. The additional point `PB:f` is plotted on the same axes. In the right panel, scores were derived and plotted for B-cell samples only.

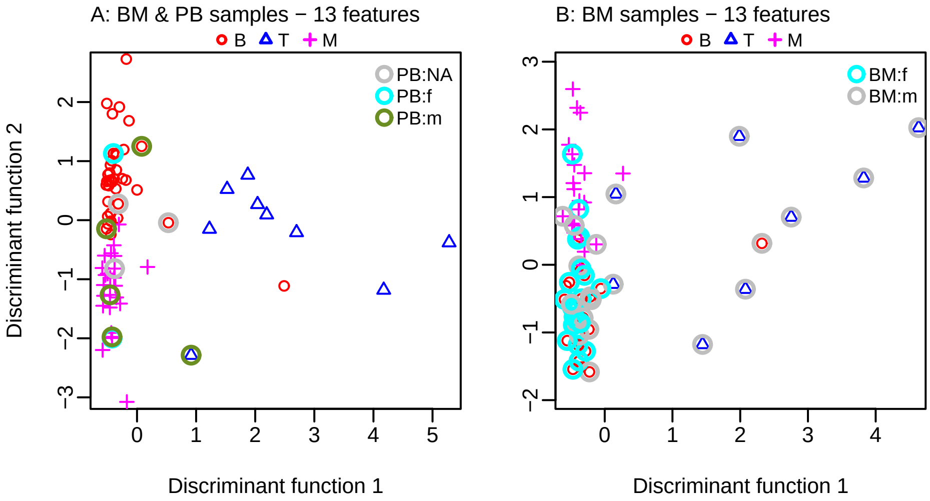

Figure 11.14: Scores are plotted for the classification into of bone marrow samples into ALL B-cell, ALL T-cell and AML. Points where `Gender` is known are identified as male or female.

Figure 11.13A shows the result.[13]Figure 11.13B has applied the same methodology to the classification of the bone marrow samples according to cancer type. Points where `Gender` is known are identified as male or female.[14]

Notice the clear clustering of points from females on the left of the graph. This complicates interpretation; is there a bias from the different gender balances in the three groups? Enough has been done to indicate that the heterogeneity of the samples is an important issue for the analysis of this data, and for the interpretation of the graphs of the graphs that have been presented. There is scope to extend further the lines of investigation that have been pursued in this section.

[13]
```
attach(golub.info)
## Uses tissue.mfB.acc from above
tissue.mfB.scores <-
   cvscores(cvlist = tissue.mfB.acc, nfeatures = 3, cl.other = NULL)
cvplot(scorelist = tissue.mfB.scores, cl.circle=NULL,
          prefix="B-cell subset -")
detach(golub.info)
```

[14]
```
BMonly.scores <- cvscores(cvlist=BMonly.acc, nfeatures=13,
                                  cl.other=NULL)
cvplot(scorelist=BMonly.scores, cl.circle=tissue.mf[BM.PB=="BM"],
          circle=source.mf[BM.PB=="BM"]%in%c("BM:f","BM:m"),
          circle.colr=c("cyan","gray"), prefix="B: BM samples -")
```