Processing the Clinical Information

Keith A. Baggerly, Shannon Neeley, and Kevin R. Coombes

October 9, 2007

1 Introduction

Here, we load the clinical information to be used in our analyses, and do some double-checking with data from related sources.

2 Options and Libraries

```
> options(width = 80)
```

3 Describing the Clinical Data

We make use of clinical data from three sources here.

- **OVCclinicalinfo.xls**. This is the clinical information supplied on the supplementary web page for Dressman et al, `http://data.genome.duke.edu/platinum.php`. This gives, for each of the 119 tumor samples, an ID to identify the sample, post-treatment survival in months (patients still alive have a "+" suffix), grade, stage, debulking status (optimal or suboptimal), CA 125 post-treatment level, and NR/CR status (NR = 0, CR = 1).
- **Ovarian_Clinical.xls**. This is the clinical information supplied on the supplementary web page for Bild et al, Nature 2006: `http://data.genome.duke.edu/oncogene.php`. This gives, for each of 135 samples, an ID to identify the sample (the same as above), post-treatment survival in months, alive/dead status (0=alive, 1=dead), and tumor stage.
- **array web site clinical data_2005.xls**. This is the clinical information supplied on the supplementary web page for Berchuck et al, Clinical Cancer Research 2005: `http://data.genome.duke.edu/clinicalcancerresearch.php`. This gives, for 65 samples, an ID to identify the array (different than the IDs used above), a tripartite classification of the tumors (early stage, long survivor (> 7 years), or short survivor (< 3 years)), age of the patient at diagnosis, stage, grade, and debulking status (optimal (O) or suboptimal (S) for patients who are not early stage).

All of the above files were acquired on October 1, 2007. All files were converted from xls to csv files to make it easier to load the data into R; white space characters in both the titles and the tables were also stripped if necessary.

4 Loading the Clinical Data

4.1 Dressman et al

We begin by loading the data from Dressman et al. We found it necessary to strip out some "empty" columns to the right of the data to get the file to load. We need to be a bit careful with the column names, as one of the names contains a comma.

```
> clinicalDressman <- read.table(file.path("DukeWebSite", "OVCclinicalinfo.csv"),
+     skip = 1, header = FALSE, sep = ",", quote = "", nrows = 119,
+     row.names = 1)
> clinicalDressman[1:3, ]
```

```
      V2 V3 V4 V5   V6 V7
0.08  14  4  3  S 72.3  0
860   17  4  3  0  133  0
872  185  3  3  S  3.1  1
```

```
> dim(clinicalDressman)
```

```
[1] 119   6
```

```
> tempHeaders <- read.table(file.path("DukeWebSite", "OVCclinicalinfo.csv"),
+     header = FALSE, sep = ",", nrows = 1)
> tempHeaders <- as.vector(apply(tempHeaders, 1, as.character))
> tempHeaders
```

```
[1] "OVC TumorID"          "Survival"             "Assigned Stage"
[4] "GRADE"                "Debulk"               "CA125 POST"
[7] "response 0=NR, 1=CR "
```

```
> tempHeaders[tempHeaders == "Assigned Stage"] <- "Stage"
> tempHeaders[tempHeaders == "GRADE"] <- "Grade"
> colnames(clinicalDressman) <- tempHeaders[2:7]
> clinicalDressman[1:3, ]
```

```
     Survival Stage Grade Debulk CA125 POST response 0=NR, 1=CR
0.08       14     4     3      S       72.3                   0
860        17     4     3      0        133                   0
872       185     3     3      S        3.1                   1
```

```
> rm(tempHeaders)
```

The basic data has now been loaded. Now we have to do some parsing to get the data into a consistent form. We'll deal with this one column at a time.

To deal with survival, we need to split the data into two pieces: the survival time in months (an integer) and alive/dead status. Currently, survival times for patients still living are given with a "+" suffix.

```
> tempSurv <- as.character(clinicalDressman$Survival)
> tempAlive <- rep("Dead", 119)
> tempAlive[grep("\\+", tempSurv)] <- "Alive"
> table(tempAlive)
```

```
tempAlive
Alive  Dead
   35    84
```

```
> tempMonths <- as.integer(unlist(strsplit(tempSurv, "\\+")))
> clinicalDressman$SurvMonths <- tempMonths
> clinicalDressman$Censoring <- as.factor(tempAlive)
> clinicalDressman[1:3, ]
```

```
     Survival Stage Grade Debulk CA125 POST response 0=NR, 1=CR  SurvMonths
0.08       14     4     3      S       72.3                   0          14
860        17     4     3      O        133                   0          17
872       185     3     3      S        3.1                   1         185
     Censoring
0.08      Dead
860       Dead
872       Dead
```

```
> rm(tempSurv, tempAlive, tempMonths)
```

Next, we look at stage.

```
> table(clinicalDressman$Stage)
```

```
 2  3  4
 1 98 19
```

```
> which(is.na((clinicalDressman$Stage)))
```

```
[1] 118
```

```
> rownames(clinicalDressman)[which(is.na((clinicalDressman$Stage)))]
```

```
[1] "M6199"
```

There's nothing that really needs fixing here; we should just be aware that there is one patient for whom this information is missing.

Now we look at grade.

```
> table(clinicalDressman$Grade)
```

```
   2/3    ?    1    2    3    4  UNK
  1    2    1    4   54   55    1    1
```

Here, we'll run with things as they are for the moment. There are 3 different ways of indicating unknown, but we'll wait to fix this until we make use of this data.

Now we take a look at debulking.

```
> tempDebulk <- as.character(clinicalDressman$Debulk)
> table(tempDebulk)
```

```
tempDebulk
                  O              Optimal                    S
                 42                   22                   49
          Suboptimal Suboptimal (10/3/97)
                  5                    1
```

```
> tempDebulk[grep("^Sub", tempDebulk)] <- "S"
> tempDebulk[grep("^Opt", tempDebulk)] <- "O"
> table(tempDebulk)
```

```
tempDebulk
 O  S
64 55
```

```
> clinicalDressman$Debulk <- as.factor(tempDebulk)
> table(clinicalDressman$Debulk)
```

```
 O  S
64 55
```

```
> levels(clinicalDressman$Debulk)
```

```
[1] "O" "S"
```

All we've done here is enforce a more consistent notation.
We will pass on checking the CA125 levels for now.
Now let's fix up the response column.

```
> colnames(clinicalDressman)[6]
```

```
[1] "response 0=NR, 1=CR "
```

```
> colnames(clinicalDressman)[6] <- "Response"
> tempResp <- clinicalDressman$Response
> tempResp[clinicalDressman$Response == 0] <- "NR"
> tempResp[clinicalDressman$Response == 1] <- "CR"
> tempResp[1:3]
```

```
[1] "NR" "NR" "CR"
```

```
> clinicalDressman$Response <- as.factor(tempResp)
> clinicalDressman[1:3, ]
```

```
     Survival Stage Grade Debulk CA125 POST Response SurvMonths Censoring
0.08       14     4     3      S       72.3       NR         14      Dead
860        17     4     3      O        133       NR         17      Dead
872       185     3     3      S        3.1       CR        185      Dead
```

```
> rm(tempResp)
```

Things are clean enough now.

4.2 Bild et al

Next, we load in the clinical information for Bild et al. We expect the 135 samples described here to be a superset of those used in Dressman et al, and we can use this for checking.

```
> clinicalBild <- read.table(file.path("OtherData", "BildNature06",
+     "Ovarian_Clinical.csv"), header = TRUE, row.names = 1, nrows = 135,
+     sep = ",")
> dim(clinicalBild)

[1] 135   3

> clinicalBild[1:3, ]

     Survival..months. Status..0.alive..1.dead. STAGE
0.08                14                        1    IV
1623               147                        0  IIIC
1447                75                        1  IIIC

> colnames(clinicalBild)

[1] "Survival..months."        "Status..0.alive..1.dead."
[3] "STAGE"

> colnames(clinicalBild) <- c("SurvMonths", "Censoring", "Stage")
> clinicalBild[1:3, ]

     SurvMonths Censoring Stage
0.08         14         1    IV
1623        147         0  IIIC
1447         75         1  IIIC

> tempCensor <- clinicalBild$Censoring
> tempCensor[clinicalBild$Censoring == 0] <- "Alive"
> tempCensor[clinicalBild$Censoring == 1] <- "Dead"
> clinicalBild$Censoring <- as.factor(tempCensor)
> clinicalBild[1:3, ]

     SurvMonths Censoring Stage
0.08         14      Dead    IV
1623        147     Alive  IIIC
1447         75      Dead  IIIC
```

This is now in usable form.

4.3 Berchuck et al

Finally, we load in the clinical information for Berchuck et al. Of these 65 samples, we expect 54 (the ones that aren't early stage) to be among those used in Dressman et al. There are some difficulties here with the header row, so we load that separately.

```
> clinicalBerchuck <- read.table(file.path("OtherData", "BerchuckClinCancRes05",
+     "arrayWebSiteClinicalData_2005.csv"), skip = 2, header = FALSE,
+     nrows = 65, sep = ",")
> dim(clinicalBerchuck)

[1] 65  6

> clinicalBerchuck[1:3, ]

    V1          V2 V3 V4 V5 V6
1 2536 Early stage 42 IA  1
2 1761 Early stage 72 IC  2
3 1762 Early stage 52 IA  3

> tempHeader <- read.table(file.path("OtherData", "BerchuckClinCancRes05",
+     "arrayWebSiteClinicalData_2005.csv"), skip = 1, header = FALSE,
+     nrows = 1, sep = ",")
> colnames(clinicalBerchuck) <- c("ArrayID", "CancerType", "AgeDx",
+     "Stage", "Grade", "Debulk")
```

It is important to note that the ID in the first column is not the same as the ID that has been used in the other clinical data files. If we look at the parsing outlined in the header line, where the template CEL file name is "0074_GenomeID_h133a_2802.cel", we see that it is the second text block that is being reported. In the other tables, the fourth block has been used. To get the appropriate fourth block, we need the names of the CEL files that were used. These are also available from `http://data.genome.duke.edu/clinicalcancerresearch.php` as "survival_CEL_files.zip" (downloaded Oct 1, 2007). All we need here is the file names, note the actual contents.

```
> berchuckCELNames <- dir(file.path("OtherData", "BerchuckClinCancRes05",
+     "CEL_Files"))
> berchuckCELNames[1:10]

 [1] "0074_1761_h133a_1564.cel" "0074_1762_h133a_2537.cel"
 [3] "0074_1763_h133a_2783.cel" "0074_1764_h133a_2883.cel"
 [5] "0074_1765_h133a_3098.cel" "0074_1772_h133a_872.cel"
 [7] "0074_1773_h133a_922.cel"  "0074_1774_h133a_1451.cel"
 [9] "0074_1775_h133a_1526.cel" "0074_1776_h133a_1784.cel"
```

Looking at some of the CEL file names, we see some samples (872, 922, etc) that we've looked at in more detail before, so we're dealing with many of the same CEL files.

```
> temp <- unlist(strsplit(berchuckCELNames, ".cel"))
> berchuckCELNames[1:2]

[1] "0074_1761_h133a_1564.cel" "0074_1762_h133a_2537.cel"

> temp[1:2]

[1] "0074_1761_h133a_1564" "0074_1762_h133a_2537"
```

```
> temp <- strsplit(temp, "_")
> berchuckArrayIDs <- unlist(lapply(temp, function(x) {
+     x[2]
+ }))
> berchuckSampleIDs <- unlist(lapply(temp, function(x) {
+     x[4]
+ }))
> names(berchuckSampleIDs) <- berchuckArrayIDs
> setdiff(berchuckArrayIDs, as.character(clinicalBerchuck$ArrayID))

[1] "2485"

> setdiff(as.character(clinicalBerchuck$ArrayID), berchuckArrayIDs)

[1] "2484"
```

Ok, there's one ID that doesn't match, going in each direction. Let's look at these in more detail.

```
> berchuckCELNames[grep("2485_h133a", berchuckCELNames)]

[1] "0074_2485_h133a_1976.cel"

> load(file.path("RDataObjects", "celFiles.Rda"))
> celFiles[grep("2484_h133a", celFiles)]

                      3249
"0074_2484_h133a_3250.cel"
```

Here, we see that the sample ID from the Berchuck et al CEL file 2485 is 1976, for which we don't have any clinical information, but if we look at CEL file 2484 from Dressman et al, we find that the sample ID is 3249 (as opposed to 3250, as we noted earlier). Our conjecture here is that CEL file 2485 was posted by mistake, and 2484 was desired. Thus, we append the 2484/3249 pair to the list of berchuckSampleIDs.

```
> berchuckSampleIDs <- c(berchuckSampleIDs, "3249")
> berchuckSampleIDs[64:66]

  2485   2536
"1976" "1357" "3249"

> names(berchuckSampleIDs)[64:66]

[1] "2485" "2536" ""

> names(berchuckSampleIDs)[66] <- "2484"
```

Having added this last pair, we're almost done. One outstanding issue is that sample ".08" is recorded as sample "0.08" elsewhere, so we need to alter this name for consistency.

```
> berchuckSampleIDs[berchuckSampleIDs == ".08"] <- "0.08"
```

we can now define rownames for clinicalBerchuck which we can use to link over to the other clinical data.

```
> rownames(clinicalBerchuck) <- berchuckSampleIDs[as.character(clinicalBerchuck$ArrayID)]
```

We note in passing that the sample IDs from Berchuck et al are all of the strictly numeric type; there are no "D" or "M" prefixes as with the other samples used in Dressman et al.

5 Checking Clinical Data

Given the assorted clinical information, it is useful to check the overall consistency. We look first at checking the Dressman results against those from Bild et al. The first step here is simply to match the sample ids and see how much overlap there is.

```
> tempOverlap <- intersect(rownames(clinicalDressman), rownames(clinicalBild))
> length(tempOverlap)

[1] 110

> setdiff(rownames(clinicalDressman), rownames(clinicalBild))

[1] "D1837" "D1859" "D2342" "D2358" "D2421" "D2733" "M1054" "M1390" "M3514"

> setdiff(rownames(clinicalBild), rownames(clinicalDressman))

 [1] "D2251" "1858"  "M4171" "D2791" "M806"  "D2711" "D2736" "D1462" "D2732"
[10] "M1025" "D2612" "D2457" "D2171" "D2651" "D2287" "D2443" "M3035" "D2247"
[19] "D2159" "D2528" "D2578" "D2775" "M1777" "D2679" "D2734"
```

Actually, the Bild set is not a superset of the Dressman set, but the overlap is pretty large (110/119).

Let's check the assessment of stage.

```
> table(clinicalDressman[tempOverlap, "Stage"], clinicalBild[tempOverlap,
+     "Stage"])

    IC IIC IIIA IIIB IIIC IIIIC IV Unstage
  2  0   1    0    0    0     0  0       0
  3  0   0    2    5   84     1  0       0
  4  0   0    0    0    0     0 16       0
```

Things look pretty good here, modulo the one typo in the Bild results (IIIIC instead of IIIC).

Next, we check on survival. We look first at survival times.

```
> sum(clinicalDressman[tempOverlap, "SurvMonths"] == clinicalBild[tempOverlap,
+     "SurvMonths"])

[1] 108

> tempOverlap[clinicalDressman[tempOverlap, "SurvMonths"] != clinicalBild[tempOverlap,
+     "SurvMonths"]]

[1] "D2432" "D2668"

> clinicalDressman[c("D2432", "D2668"), ]

      Survival Stage Grade Debulk CA125 POST Response SurvMonths Censoring
D2432       34     3     2      S          9       CR         34      Dead
D2668       40     3           S                  CR         40      Dead

> clinicalBild[c("D2432", "D2668"), ]
```

```
      SurvMonths Censoring Stage
D2432         20      Dead  IIIC
D2668         41      Dead  IIIC
```

Here, we have two discrepancies. The difference for D2668, from 40 to 41 months, is not that bothersome. The other difference, with D2432 going from 20 to 34 months, is a bit disturbing. Nonetheless, the overlap is overall very high.

Finally, let's check the censoring status.

```
> table(clinicalDressman[tempOverlap, "Censoring"], clinicalBild[tempOverlap,
+     "Censoring"])

        Alive Dead
  Alive    31    1
  Dead     14   64
```

```
> censorBad <- tempOverlap[clinicalDressman[tempOverlap, "Censoring"] !=
+     clinicalBild[tempOverlap, "Censoring"]]
> censorBad
```

```
 [1] "872"   "922"   "1590"  "1623"  "1846"  "1929"  "2046"  "2204"  "2419"
[10] "2479"  "2505"  "2673"  "2739"  "3102"  "D2572"
```

This is quite disturbing; censoring status should not be ambiguous. Let's take a look at where the disagreements occur.

```
> clinicalDressman[censorBad, ]
```

```
      Survival Stage Grade Debulk CA125 POST Response SurvMonths Censoring
872        185     3     3      S        3.1       CR        185      Dead
922        183     3     2      S          9       CR        183      Dead
1590       148     3     2      O          7       CR        148      Dead
1623       147     3     1      O          4       CR        147      Dead
1846       142     3     2      O          9       CR        142      Dead
1929       134     3     4      S          7       CR        134      Dead
2046       127     3     2      O          6       CR        127      Dead
2204       118     3     2      O          7       CR        118      Dead
2419       107     3     3      O         10       CR        107      Dead
2479        95     3     2      S       16.5       CR         95      Dead
2505        95     3     2      S         12       CR         95      Dead
2673        74     3     2      S       11.5       CR         74      Dead
2739        67     3     2      O         10       CR         67      Dead
3102       10+     3     ?      S      127.3       NR         10     Alive
D2572       37     3     2      S         24       CR         37      Dead
```

```
> clinicalBild[censorBad, ]
```

```
      SurvMonths Censoring Stage
872          185     Alive  IIIC
922          183     Alive  IIIC
1590         148     Alive  IIIC
```

```
1623         147     Alive  IIIC
1846         142     Alive  IIIB
1929         134     Alive  IIIC
2046         127     Alive  IIIC
2204         118     Alive  IIIC
2419         107     Alive  IIIC
2479          95     Alive  IIIC
2505          95     Alive  IIIC
2673          74     Alive  IIIC
2739          67     Alive  IIIC
3102          10      Dead  IIIC
D2572         37     Alive  IIIC
```

```
> sort(clinicalDressman$SurvMonths)[100:119]
```

```
 [1]  95  95  95  98 101 103 107 108 110 118 118 119 127 132 134 142 147 148 183
[20] 185
```

In going from clinicalBild to clinicalDressman, 14 patients, all CR, shift from Alive to Dead. Of these 14, 11 are among the 20 longest survival times recorded here, including the very top 7. One patient shifts from Dead to Alive (rather odd), and this patient is NR. This type of shift will alter the shapes of the survival curves rather substantially. One other point is that all but one of these shifts apply to samples with strictly numerical labels. These samples are among those examined by Berchuck et al, so we can check that information as well.

```
> clinicalBerchuck[censorBad, ]
```

```
     ArrayID CancerType AgeDx Stage Grade Debulk
872     1772       Long    33  IIIC     3      S
922     1773       Long    68  IIIC     2      S
1590    1901       Long    50  IIIC     2      O
1623    1902       Long    67  IIIC     1      O
1846    1778       Long    66  IIIB     2      O
1929    1907       Long    73  IIIC     4      S
2046    2020       Long    58  IIIC     2      O
2204    1780       Long    63  IIIC     2      O
2419    1781       Long    73  IIIC     3      O
2479    2021       Long    62  IIIC     2      S
2505    2033       Long    57  IIIC     2      S
2673    2032       Long    58  IIIC     2      S
2739    2031       Long    63  IIIC     2      O
3102    2005      Short    47  IIIC     ?      S
NA        NA       <NA>    NA  <NA>  <NA>   <NA>
```

We see that we have shifted the status of 13 long survivors from Alive to Dead, and the status of one short survivor from Dead to Alive.

Let's take a closer look at some of the clincial data from Berchuck et al now, contrasting it with that presented for Dressman et al.

```
> tempOverlapBerDre <- intersect(rownames(clinicalBerchuck), rownames(clinicalDressman))
> length(tempOverlapBerDre)
```

```
[1] 53
```

```
> clinicalBerchuck[setdiff(rownames(clinicalBerchuck), rownames(clinicalDressman)),
+     ]
```

```
     ArrayID  CancerType AgeDx Stage Grade Debulk
1357    2536 Early stage    42    IA     1
1564    1761 Early stage    72    IC     2
2537    1762 Early stage    52    IA     3
2783    1763 Early stage    58   IIC     2
2883    1764 Early stage    48    IC     3
3098    1765 Early stage    45    IA     2
1859    1903 Early stage    73   IIC     3
1478    2390 Early stage    66    IA     2
3396    2391 Early stage    77    IA     3
1605    2392 Early stage    46   IIC     3
3449    2393 Early stage    49   IIC     2
1858    1832       Short    59  IIIC     2      S
```

As predicted almost all of the non-Early-Stage samples from Berchuck et al are also used in Dressman et al. There is one outlier, which we can't explain, but the overlap is still quite large. We can directly contrast data for Stage, Grade, and Debulking status.

```
> table(clinicalDressman[tempOverlapBerDre, "Stage"], clinicalBerchuck[tempOverlapBerDre,
+     "Stage"])
```

```
    IA IC IIC IIIA IIIB IIIC IV
  3  0  0   0    1    3   43  0
  4  0  0   0    0    0    0  6
```

```
> table(clinicalDressman[tempOverlapBerDre, "Grade"], clinicalBerchuck[tempOverlapBerDre,
+     "Grade"])
```

```
       ?  1  2  3  4
       0  0  0  0  0
   2/3  0  0  0  0  0
  ?     1  0  0  0  0
  1     0  1  0  0  0
  2     0  0 30  0  0
  3     0  0  0 20  0
  4     0  0  0  0  1
  UNK   0  0  0  0  0
```

```
> table(clinicalDressman[tempOverlapBerDre, "Debulk"], clinicalBerchuck[tempOverlapBerDre,
+     "Debulk"])
```

```
       0  S
  0  0 25  0
  S  0  0 28
```

All of the above information lines up pretty much perfectly, causing no real problems. There is one last contrast we want to try, involving CancerType (from Berchuck et al) and Censoring (from Dressman et al).

```
> table(clinicalDressman[tempOverlapBerDre, "Censoring"], clinicalBerchuck[tempOverlapBerDre,
+     "CancerType"])

       Early stage Long Short
  Alive           0    0     1
  Dead            0   24    28
```

There does not appear to be any major separation by CancerType, but the important point is that the clinical information from Dressman et al says that essentially all of the Berchuck et al patients (and all of the long survivors) are Dead. Given that Berchuck et al (p.3687) states that "Twelve long-term survivors remain alive", this strongly suggests that the Censoring status from the Bild et al clinical information is correct, and that from Dressman et al incorrect.

6 Assembling Clinical Info

Combining data across the various files, we are for the most part running with the clinical information from Dressman et al, with one addition. We include two columns for Censoring, one corresponding to that from Dressman et al, and the other corresponding to that from Bild et al. We repeat the information from Dressman et al for the 9 samples not described by Bild et al.

```
> tempCensor <- clinicalDressman[, "Censoring"]
> names(tempCensor) <- rownames(clinicalDressman)
> tempCensor[censorBad]

  872   922  1590  1623  1846  1929  2046  2204  2419  2479  2505  2673  2739
 Dead  Dead  Dead  Dead  Dead  Dead  Dead  Dead  Dead  Dead  Dead  Dead  Dead
 3102 D2572
Alive  Dead
Levels: Alive Dead

> tempCensor[censorBad] <- "Alive"
> tempCensor["3102"] <- "Dead"
> tempCensor[censorBad]

  872   922  1590  1623  1846  1929  2046  2204  2419  2479  2505  2673  2739
Alive Alive Alive Alive Alive Alive Alive Alive Alive Alive Alive Alive Alive
 3102 D2572
 Dead Alive
Levels: Alive Dead

> clinicalInfo <- cbind(clinicalDressman[, c("SurvMonths", "Censoring")],
+     CensoringBild = tempCensor, clinicalDressman[, c("Response",
+         "Stage", "Grade", "Debulk", "CA125 POST")])
> clinicalInfo[1:5, ]

     SurvMonths Censoring CensoringBild Response Stage Grade Debulk CA125 POST
0.08         14      Dead          Dead       NR     4     3      S       72.3
```

```
860          17      Dead          Dead       NR     4     3      O        133
872         185      Dead         Alive       CR     3     3      S        3.1
922         183      Dead         Alive       CR     3     2      S          9
1024         13      Dead          Dead       CR     4     3      S         12
```

This last data frame is the one we shall make most use of.

7 Summary

1. There are three sources of clinical data, corresponding to the papers by Dressman et al. (119 samples), Bild et al. (135 samples), and Berchuck et al. (54 samples).
2. Dressman and Bild describe 110 common samples (instead of the 119 that we would expect). Only two of the common samples are recorded with different survival times.
3. Fifteen of the samples common to Dressman and Bild have different survivla outcomes recorded (14 dead and 1 alive in Dressman that changes to 14 alive and 1 dead in Bild). The 14 all are listed as CR, and the odd 1 is listed as NR.
4. Berchuck and Dressman describe 53 common samples (instead of the 54 that we would expect). Dressman lists 52/53 of these patients as dead, even though Berchuck makes a point that 12 patients remained alive.

8 Appendix

8.1 Saves

```
> save(clinicalDressman, file = paste("RDataObjects", "clinicalDressman.Rda",
+     sep = .Platform$file.sep))
> save(clinicalBild, file = paste("RDataObjects", "clinicalBild.Rda",
+     sep = .Platform$file.sep))
> save(clinicalBerchuck, file = paste("RDataObjects", "clinicalBerchuck.Rda",
+     sep = .Platform$file.sep))
> save(clinicalInfo, file = paste("RDataObjects", "clinicalInfo.Rda",
+     sep = .Platform$file.sep))
```

8.2 SessionInfo

```
> sessionInfo()

R version 2.5.1 (2007-06-27)
i386-pc-mingw32

locale:
LC_COLLATE=English_United States.1252;LC_CTYPE=English_United States.1252;LC_MONETARY=English_United Sta

attached base packages:
[1] "splines"   "tools"     "stats"     "graphics"  "grDevices" "utils"
[7] "datasets"  "methods"   "base"
```

```
other attached packages:
       survival  ClassDiscovery         cluster ClassComparison      PreProcess
         "2.32"         "2.5.0"        "1.11.7"         "2.5.0"         "2.5.0"
      oompaBase     geneplotter         lattice        annotate            affy
        "2.5.0"        "1.14.0"       "0.15-11"        "1.14.1"        "1.14.2"
         affyio         Biobase
        "1.4.1"        "1.14.1"
```