# Step-by-step user instructions to the hamlet-package

Teemu Daniel Laajala

January 27, 2015

## Contents

`Hamlet` is an R package intended for the statistical analysis of pre-clinical studies. This document is a basic introduction to the functionality of `hamlet` and a general overview to the analysis workflow of preclinical studies.

This document is structured as follows: First, a general overview to inputting and processing the raw data is presented. Second, functionality is presented for the processing of pre-intervention data. Finally, functionality is presented for the post-intervention period, along with brief discussion on the differences between non-matched and matched statistical approaches. Each section comes with a list of useful functions specific for the subtask.

Latest version of `hamlet` is available in the Comprehensive R Archive Network (CRAN, http://cran.r-project.org/). CRAN mirrors are by default available in the installation of R, and the `hamlet` package is installable using the R terminal command: `install.packages("hamlet")`. This should prompt the user to select a nearby CRAN mirror, after which the installation of `hamlet` is automatically performed. After the `install.packages`-call, the `hamlet` package can be loaded with either command `library("hamlet")` or `require("hamlet")`.

The following notation is used in the document: R commands, package names and function names are written in `typewriter font`. The notation of format `pckgName::funcName` indicates that the function `funcName` is called from the package `pckgName`. If only the function name is given, this indicates that it is located in the base package in R and is thus always available.

# 1 Analysis workflow

Two different types of case-control setups for the analysis of pre-clinical are presented in Fig. **??**.

The type A experiment design in Fig. **??** is preferred, as matching is performed before allocation to the experiment groups, and therefore improves the balance and power of the experiment. The alternate experiment type B requires the bipartite matching task, where suitable pairs of individuals are identified over two or more groups that existed prior to matching. This document presents a dataset where experiment design type A was used.

# 2 Loading data into R

The `hamlet` package comes pre-installed with the VCaP dataset, which is used here to illustrate the workflow. Two different formats of the data are provided. First one is available in `data(vcapwide)`, which includes the data in the so-called *wide* format. In this data format the columns are indicators for different variables available for the experimental unit (here animal). For example, the two first rows of observations are extracted with:

```
> require(hamlet)
> data(vcapwide)
> vcapwide[1:2,]
```

```
      CastrationDate CageAtAllocation   Group TreatmentInitiationWeek
ID003         100413            13489 Vehicle                  Week10
```

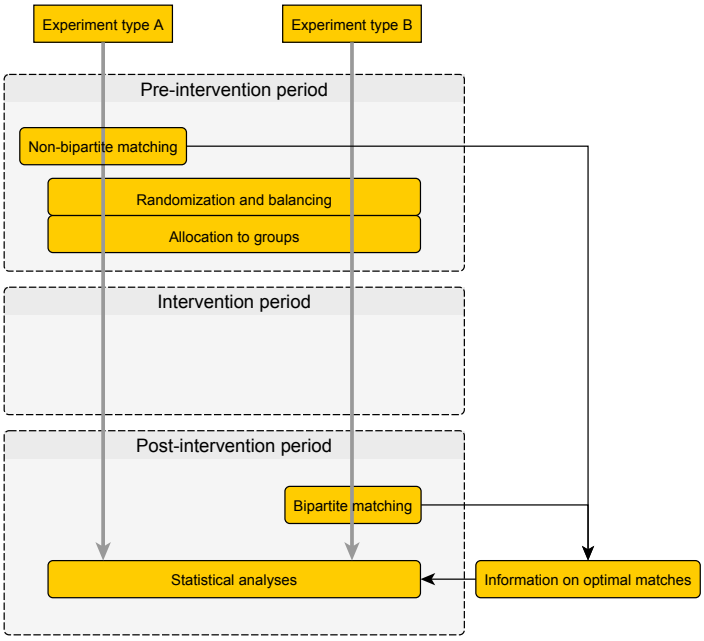


Figure 1: Analysis workflow for pre-clinical experiments

```
ID007         170413             13810      MDV                  Week10
         Submatch     ID PSAWeek2 PSAWeek3 PSAWeek4 PSAWeek5 PSAWeek6 PSAWeek7
ID003  Submatch_1 ID003      7.67    14.76    24.78     2.03     5.97     8.16
ID007 Submatch_10 ID007      2.01     5.17     8.59    14.62     1.99     2.81
      PSAWeek8 PSAWeek9 PSAWeek10 PSAWeek11 PSAWeek12 PSAWeek13 PSAWeek14
ID003    13.72    16.57     21.30     45.69     54.50     53.55     27.64
ID007     4.23     5.38      7.55      9.70     17.45     22.79     21.88
      BWWeek0 BWWeek1 BWWeek2 BWWeek3 BWWeek4 BWWeek5 BWWeek6 BWWeek7 BWWeek8
ID003    30.5    31.7    32.6    33.8    33.9    32.2    32.6    32.6    33.2
ID007    28.8    30.0    30.6    31.6    32.9    32.4    32.0    31.1    30.3
      BWWeek9 BWWeek10 BWWeek11 BWWeek12 BWWeek13 BWWeek14
ID003    34.2     35.0     36.1     37.9     37.5     39.7
ID007    30.5     31.6     31.7     32.4     33.5     33.3
```

An another format of the same dataset is provided in `data(vcaplong)`. This is the data from the same experiment in the so-called *long* format, where only few column variables are available (here PSA or body weight), and the different observations belonging to a single experimental unit (here animal) are distinguished using the measurement time (variable *Week* or *DrugWeek*). Again, first few rows of the dataset:

```
> data(vcaplong)
> vcaplong[1:3,]
```

```
    log2PSA   BW   Submatch    ID Week DrugWeek   Group Vehicle ARN MDV
11 4.412782 35.0 Submatch_1 ID003   10        0 Vehicle       1   0   0
```

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Animal | PSA week 10 [ug/l] | PSA week 9 [ug/l] | Body weight week 10 [g] |
| 2 | ID003 | 21,3 | 16,57 | 35 |
| 3 | ID007 | 7,55 | 5,38 | 31,6 |
| 4 | ID008 | 23,58 | 17,4 | 33,6 |
| 5 | ID009 | 13,17 | 11,14 | 31,7 |
| 6 | ID010 | 9,9 | 9,33 | 34,1 |
| 7 | ID016 | 15,05 | 15,29 | 39,6 |
| 8 | ID018 | 13,53 | 12,14 | 34 |
| 9 | ID025 | 13,13 | 10,91 | 33,3 |
| 10 | ID027 | 9,59 | 8,79 | 32 |
| 11 | ID031 | 7,04 | 6,95 | 36,6 |
| 12 | ID032 | 8,49 | 8,02 | 34,9 |
| 13 | ID037 | 13,74 | 13,38 | 32,4 |
| 14 | ID040 | 23,62 | 19,15 | 35,9 |
| 15 | ID045 | 14,27 | 9,8 | 34,8 |
| 16 | ID047 | 6,57 | 6,28 | 31,9 |
| 17 | ID054 | 34,72 | 27,14 | 32,1 |
| 18 | ID056 | 28,15 | 22,05 | 32,2 |
| 19 | ID058 | 9,74 | 7,68 | 34 |

Figure 2: Example Excel-format data, where rows correspond to individuals and columns to different characteristics at baseline. The single sheet data can be easily exported in a text-based format such as CSV.

```
12 5.513807 36.1 Submatch_1 ID003   11        1 Vehicle       1   0   0
13 5.768184 37.9 Submatch_1 ID003   12        2 Vehicle       1   0   0
```

The former *wide* format is useful for summarizing multiple variables when constructing distance matrices for the data. The latter *long* format is typically used for longitudinal mixed-effects modeling where observations are correlated through time.

## 2.1 Excel format data

An example view of a pre-clinical dataset is given in Fig. **??**. Such a dataset can be saved in an R-friendly format by selecting option `File > Save As` and `CSV (Comma delimited)` as the save format in MS Excel.

## 2.2 CSV-files

CSV (Comma Delimited Values) is a suitable text-based format for the data to be read into R using either the function `read.table` or `read.csv`. The above presented example CSV file can be opened with the following command:

```
> ex <- read.table(file="example.csv", sep=";", dec=",", stringsAsFactors=F, header=T)
> ex

   Animal PSA.week.10..ug.l. PSA.week.9..ug.l. Body.weight.week.10..g.
1   ID003              21.30             16.57                    35.0
2   ID007               7.55              5.38                    31.6
```

| | | | | |
|---|---|---|---|---|
| 3 | ID008 | 23.58 | 17.40 | 33.6 |
| 4 | ID009 | 13.17 | 11.14 | 31.7 |
| 5 | ID010 | 9.90 | 9.33 | 34.1 |
| 6 | ID016 | 15.05 | 15.29 | 39.6 |
| 7 | ID018 | 13.53 | 12.14 | 34.0 |
| 8 | ID025 | 13.13 | 10.91 | 33.3 |
| 9 | ID027 | 9.59 | 8.79 | 32.0 |
| 10 | ID031 | 7.04 | 6.95 | 36.6 |
| 11 | ID032 | 8.49 | 8.02 | 34.9 |
| 12 | ID037 | 13.74 | 13.38 | 32.4 |
| 13 | ID040 | 23.62 | 19.15 | 35.9 |
| 14 | ID045 | 14.27 | 9.80 | 34.8 |
| 15 | ID047 | 6.57 | 6.28 | 31.9 |
| 16 | ID054 | 34.72 | 27.14 | 32.1 |
| 17 | ID056 | 28.15 | 22.05 | 32.2 |
| 18 | ID058 | 9.74 | 7.68 | 34.0 |

The above presented CSV file was read into R using `read.table` with the following parameters: `file="example.csv"` is the first parameter and indicates the input file from our current working directory. The working directory may be changed using the command `setwd` or by including its path in the file parameter, i.e. `file="D://my//current//windows//working//directory//example.csv"`. `sep=";"` indicates that the values on each line are separated with the symbol ';', as is the format defined for the CSV delimited files with ","-decimals. This could also be a value such as `\tab` or `" "` (space). `dec=","` indicates that the "," symbol is used for decimals. The default value for indicating decimals is "." otherwise. `stringsAsFactors=F` indicates that strings should not be handled as factors. Factors are an R class, where a character string may only take instances of a predetermined set of strings. As each of our animal IDs - which are read as strings - are unique, it is generally more flexible to conserve them as character strings. Lastly, `header=T` indicates that the text CSV file has a header row as the first row, which includes names for each column. If this value is set to `header=F` or `header=FALSE`, the first row of the text file is read as the first observation and the columns are left unnamed.

Depending on the country of origin, the CSV files may use "." decimals and "," separator, or alternatively (as assumed here) "," decimals" and ";" separators.

List of useful functions:

- `read.table`, `read.csv`
- `data`: `data(vcaplong)`, `data(vcapwide)`

# 3 Distance and dissimilarity functions

A distance or dissimilarity function is used to describe the amount of (dis-)similarity between two experimental units. Common choices for computing the amount of similarity between two vectors $\mathbf{x}$ and $\mathbf{y}$ include:

- Euclidean distance: $\mathrm{d}(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$.

- Standardized Euclidean distance: $\sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{s_i^2}}$
- Mahalanobis distance: $\sqrt{(x-y)^T S^{-1} (x-y)}$

Here, $\mathbf{x}$ and $\mathbf{y}$ are expected to be observation vectors of length $N$, where each dimension describes the measured value for a particular covariate. $S$ describes the covariance-variance matrix between covariates, and therefore incorporates inter-correlations between variables. The standard deviation $s$ may be used to standardize differences in variation over the dimensions.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.00 | 18.05 | 2.80 | 10.32 | 13.53 | 7.87 | 9.00 | 10.08 | 14.38 | 17.28 | 15.40 | 8.61 | 3.58 | 9.76 | 18.23 | 17.33 | 9.21 | 14.62 |
| 2 | 18.05 | 0.00 | 20.14 | 8.05 | 5.23 | 14.78 | 9.34 | 8.04 | 3.99 | 5.27 | 4.33 | 10.15 | 21.60 | 8.66 | 1.36 | 34.81 | 26.51 | 3.98 |
| 3 | 2.80 | 20.14 | 0.00 | 12.29 | 15.89 | 10.64 | 11.35 | 12.30 | 16.50 | 19.79 | 17.82 | 10.70 | 2.89 | 12.08 | 20.39 | 14.87 | 6.67 | 16.92 |
| 4 | 10.32 | 8.05 | 12.29 | 0.00 | 4.44 | 9.12 | 2.53 | 1.62 | 4.29 | 8.90 | 6.47 | 2.42 | 13.82 | 3.55 | 8.20 | 26.84 | 18.54 | 5.39 |
| 5 | 13.53 | 5.23 | 15.89 | 4.44 | 0.00 | 9.61 | 4.59 | 3.68 | 2.19 | 4.48 | 2.08 | 5.83 | 16.97 | 4.45 | 5.02 | 30.61 | 22.33 | 1.66 |
| 6 | 7.87 | 14.78 | 10.64 | 9.12 | 9.61 | 0.00 | 6.60 | 7.91 | 11.39 | 11.95 | 10.86 | 7.56 | 10.10 | 7.33 | 14.57 | 24.16 | 16.49 | 10.84 |
| 7 | 9.00 | 9.34 | 11.35 | 2.53 | 4.59 | 6.60 | 0.00 | 1.47 | 5.54 | 8.71 | 6.57 | 2.04 | 12.43 | 2.58 | 9.34 | 26.03 | 17.75 | 5.85 |
| 8 | 10.08 | 8.04 | 12.30 | 1.62 | 3.68 | 7.91 | 1.47 | 0.00 | 4.33 | 7.98 | 5.70 | 2.70 | 13.59 | 2.19 | 8.15 | 27.04 | 18.73 | 4.73 |
| 9 | 14.38 | 3.99 | 16.50 | 4.29 | 2.19 | 11.39 | 5.54 | 4.33 | 0.00 | 5.57 | 3.20 | 6.20 | 17.87 | 5.55 | 3.93 | 31.12 | 22.81 | 2.29 |
| 10 | 17.28 | 5.27 | 19.79 | 8.90 | 4.48 | 11.95 | 8.71 | 7.98 | 5.57 | 0.00 | 2.48 | 10.19 | 20.60 | 7.98 | 4.77 | 34.56 | 26.32 | 3.82 |
| 11 | 15.40 | 4.33 | 17.82 | 6.47 | 2.08 | 10.86 | 6.57 | 5.70 | 3.20 | 2.48 | 0.00 | 7.91 | 18.81 | 6.05 | 3.96 | 32.58 | 24.30 | 1.58 |
| 12 | 8.61 | 10.15 | 10.70 | 2.42 | 5.83 | 7.56 | 2.04 | 2.70 | 6.20 | 10.19 | 7.91 | 0.00 | 11.96 | 4.34 | 10.10 | 25.09 | 16.82 | 7.14 |
| 13 | 3.58 | 21.60 | 2.89 | 13.82 | 16.97 | 10.10 | 12.43 | 13.59 | 17.87 | 20.60 | 18.81 | 11.96 | 0.00 | 13.27 | 21.73 | 14.19 | 6.53 | 18.11 |
| 14 | 9.76 | 8.66 | 12.08 | 3.55 | 4.45 | 7.33 | 2.58 | 2.19 | 5.55 | 7.98 | 6.05 | 4.34 | 13.27 | 0.00 | 8.95 | 26.95 | 18.69 | 5.07 |
| 15 | 18.23 | 1.36 | 20.39 | 8.20 | 5.02 | 14.57 | 9.34 | 8.15 | 3.93 | 4.77 | 3.96 | 10.10 | 21.73 | 8.95 | 0.00 | 35.04 | 26.73 | 4.05 |
| 16 | 17.33 | 34.81 | 14.87 | 26.84 | 30.61 | 24.16 | 26.03 | 27.04 | 31.12 | 34.56 | 32.58 | 25.09 | 14.19 | 26.95 | 35.04 | 0.00 | 8.31 | 31.72 |
| 17 | 9.21 | 26.51 | 6.67 | 18.54 | 22.33 | 16.49 | 17.75 | 18.73 | 22.81 | 26.32 | 24.30 | 16.82 | 6.53 | 18.69 | 26.73 | 8.31 | 0.00 | 23.42 |
| 18 | 14.62 | 3.98 | 16.92 | 5.39 | 1.66 | 10.84 | 5.85 | 4.73 | 2.29 | 3.82 | 1.58 | 7.14 | 18.11 | 5.07 | 4.05 | 31.72 | 23.42 | 0.00 |

Table 1: Euclidean distance matrix D for 18 animals

Table **??** shows the Euclidean distance matrix for the 18 animals presented in Figure **??**.

List of useful functions:

- `dist` includes many common distance and dissimilarity functions (Euclidean by default, others: `method="manhattan"`, `method="maximum"`, `method="minkowski"`
- `cluster::daisy`, daisy includes Gower's dissimilarity for mixed data (parameter `metric="gower"`)

# 4 Non-bipartite optimal matching of animals at baseline

The non-bipartite optimal matching problem may be solved using the provided branch and bound algorithm:

```
> sol <- match.bb(d, g=3)

[1] "Performing initial sorting for a good initial guess"
[1] "Computing boundaries for minimum distances in possible combinations..."
[1] "Starting branch and bound"
[1] "Branches: 272"
[1] "Bounds: 7140"
[1] "Ends visited: 25"
```

```
[1] "Solution cost 169.62"
[1] "Solution: 5,3,5,6,4,5,6,4,3,2,2,6,1,4,3,1,1,2"
```

```
> submatches <- paste("Submatch_", LETTERS[1:6][sol$solution], sep="")
> names(submatches) <- names(sol$solution)
> submatches
```

```
          1           2           3           4           5           6
"Submatch_E" "Submatch_C" "Submatch_E" "Submatch_F" "Submatch_D" "Submatch_E"
          7           8           9          10          11          12
"Submatch_F" "Submatch_D" "Submatch_C" "Submatch_B" "Submatch_B" "Submatch_F"
         13          14          15          16          17          18
"Submatch_A" "Submatch_D" "Submatch_C" "Submatch_A" "Submatch_A" "Submatch_B"
```

The `match.bb` function returns the solution to the optimal matching task. It takes as input a distance matrix `d`, as is indicated in the function call `match.bb(d, g=3)` (notice that `d` was defined before). Furthermore, the size of the submatches is defined using the parameter `g=3`. This value indicates that the optimal matching algorithm minimizes edges within triplets. Each observation has to belong to a triplet called a submatch.

List of useful functions:

- Multigroup non-bipartite matching: `hamlet::match.bb`
- Paired non-bipartite matching: `hamlet::match.bb`, `nbpMatching::nonbimatch`
- Paired bipartite matching: `optmatch::fullmatch`

# 5 Randomization based on matched individuals

The submatches identified in the above section should not be mistaken for the randomly allocated intervention groups. The final intervention groups are obtained by dividing members of each submatch in the found solution to a separate treatment arm. Since the within-submatch distances are minimized, this guarantees that comparable individuals are randomly divided to separate arms:

```
> ex[,"Submatch"] <- submatches
> set.seed(1) # for reproducibility
> ex[,"AllocatedGroups"] <- match.allocate(ex[,"Submatch"])
```

List of useful functions:

- Multigroup non-bipartite matching: `hamlet::match.bb`
- Paired non-bipartite matching: `hamlet::match.bb`, `nbpMatching::nonbimatch`
- Paired bipartite matching: `optmatch::fullmatch`

```
> boxplot(PSA.week.10..ug.l. ~ AllocatedGroups, data = ex, range=0,
+ xlab="Group", ylab="PSA week 10 ul/g")
```

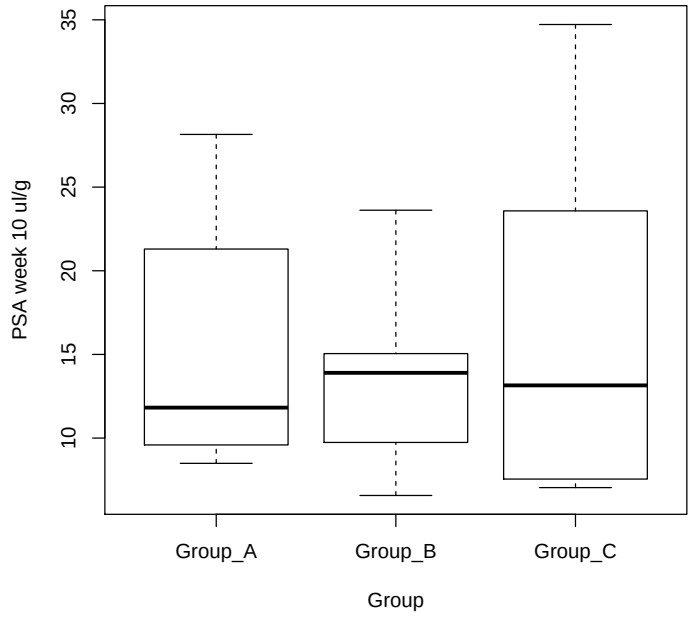


Figure 3: Boxplots for the week 10 PSA in the example allocation

| | Animal | PSA.week.10..ug.l. | PSA.week.9..ug.l. | Body.weight.week.10..g. | Submatch | AllocatedGroups |
|---|---|---|---|---|---|---|
| 1 | ID003 | 21.30 | 16.57 | 35.00 | Submatch_E | Group_A |
| 2 | ID007 | 7.55 | 5.38 | 31.60 | Submatch_C | Group_C |
| 3 | ID008 | 23.58 | 17.40 | 33.60 | Submatch_E | Group_C |
| 4 | ID009 | 13.17 | 11.14 | 31.70 | Submatch_F | Group_C |
| 5 | ID010 | 9.90 | 9.33 | 34.10 | Submatch_D | Group_A |
| 6 | ID016 | 15.05 | 15.29 | 39.60 | Submatch_E | Group_B |
| 7 | ID018 | 13.53 | 12.14 | 34.00 | Submatch_F | Group_B |
| 8 | ID025 | 13.13 | 10.91 | 33.30 | Submatch_D | Group_C |
| 9 | ID027 | 9.59 | 8.79 | 32.00 | Submatch_C | Group_A |
| 10 | ID031 | 7.04 | 6.95 | 36.60 | Submatch_B | Group_C |
| 11 | ID032 | 8.49 | 8.02 | 34.90 | Submatch_B | Group_A |
| 12 | ID037 | 13.74 | 13.38 | 32.40 | Submatch_F | Group_A |
| 13 | ID040 | 23.62 | 19.15 | 35.90 | Submatch_A | Group_B |
| 14 | ID045 | 14.27 | 9.80 | 34.80 | Submatch_D | Group_B |
| 15 | ID047 | 6.57 | 6.28 | 31.90 | Submatch_C | Group_B |
| 16 | ID054 | 34.72 | 27.14 | 32.10 | Submatch_A | Group_C |
| 17 | ID056 | 28.15 | 22.05 | 32.20 | Submatch_A | Group_A |
| 18 | ID058 | 9.74 | 7.68 | 34.00 | Submatch_B | Group_B |

Table 2: The result table in variable `ex` after performing the optimal matching and allocation.

# 6 Visualizations for pre-clinical data

Various visualization functions are available to illustrate baseline balance. For example, the boxplots in respect to allocation groups can be plotted using a command such as `boxplot`, which is illustrated in Figure **??**.

Mixed variable scatterplots with annotations for the submatches or allocation groups are plotted using the function `hamlet::mixplot`, which can be seen in Figures **??** or **??** respectively.

A common way to illustrate distance matrices is through heatmaps, along with hierarchical clustering to connect similar individuals (Figure **??**).

List of useful functions:

- Scatterplots etc: `hamlet::mixplot`, `plot`, `boxplot`
- Heatmaps: `hamlet::hmap`, `heatmap`, `gplots::heatmap.2`

# 7 Paired vs. non-paired testing

Pairing of samples aims to increase statistical power of tests through connecting the same measurement over two conditions. In the pre-clinical context, we incorporate the matching information from prior to interventions (Figure **??**) in order to find the connecting measurements. As the submatches were used to couple measurements prior to interventions, it is natural to use this information for pairing the observations in the post-intervention tests.

If pairing is not performed, the statistical power may not be sufficient in the tests, as the populations are assumed to relatively homogeneous and no prognostic information is incorporated to the testing procedure:

```
> veh <- vcapwide[vcapwide[,"Group"]=="Vehicle",
+         c("Submatch","PSAWeek10","BWWeek10","PSAWeek14")]
> mdv <- vcapwide[vcapwide[,"Group"]=="MDV",
```

```
> mixplot(ex[,2:5], pch=16)
```

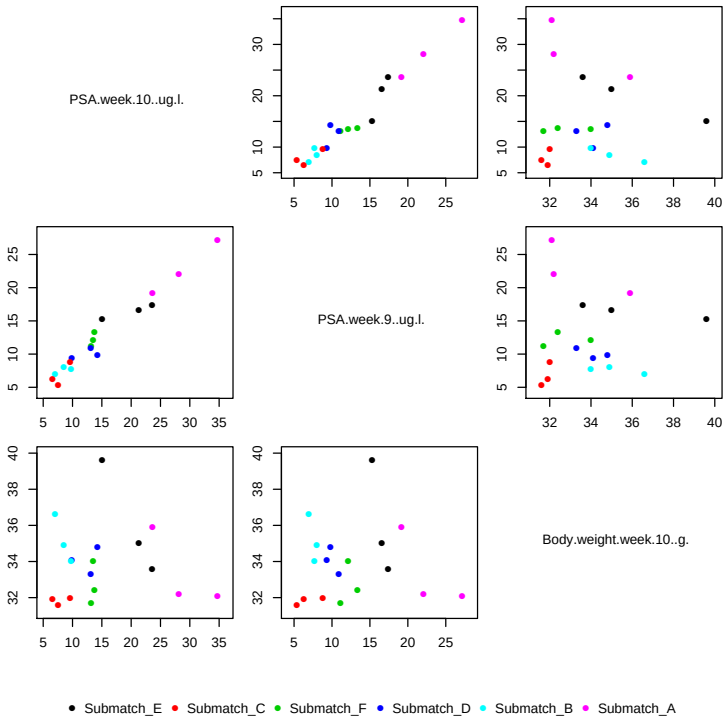


Figure 4: Test mixplot with submatch labels

```
> mixplot(ex[,c(2:4,6)], pch=16)
```

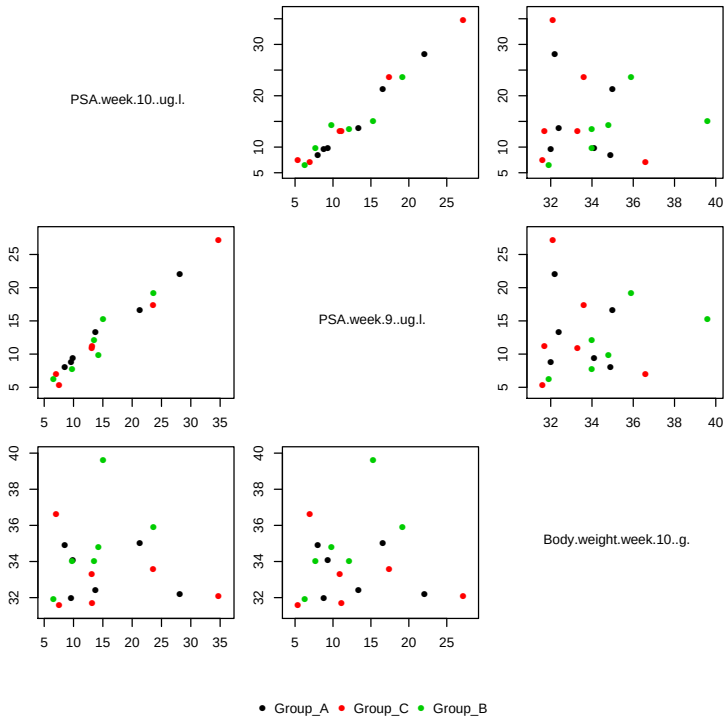


Figure 5: Test mixplot with allocation group labels

```
> heatmap(d)
```

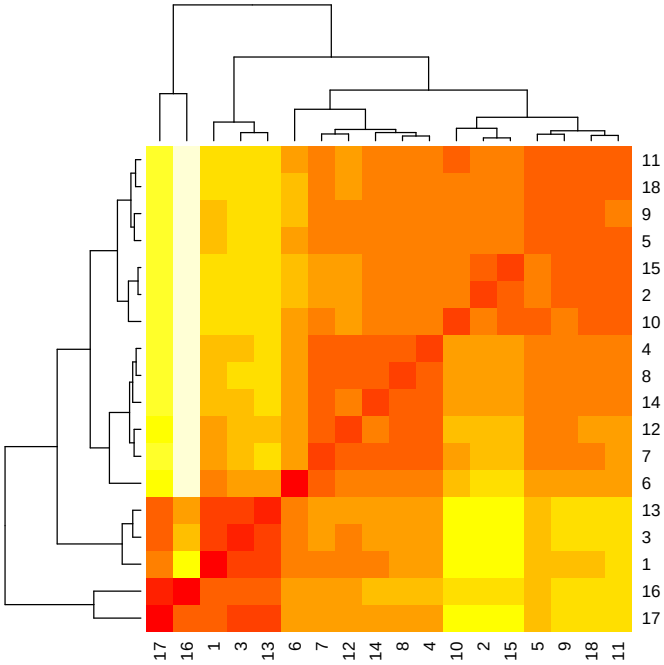


Figure 6: Heatmap for the 18x18 distance matrix

```
+         c("Submatch","PSAWeek10","BWWeek10","PSAWeek14")]
> t.test(veh[,"PSAWeek14"], mdv[,"PSAWeek14"])

        Welch Two Sample t-test

data:  veh[, "PSAWeek14"] and mdv[, "PSAWeek14"]
t = 1.7385, df = 26.796, p-value = 0.0936
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.18889 38.49813
sample estimates:
mean of x mean of y
 53.39533  35.74071
```

However, the submatches are used to couple best comparable measurements. For example, in the VCaP experiment the week 10 was baseline and its PSA and body weight contain prognostic information necessary for weighting post-intervention tests. During the experiment weeks 11 to 14 interventions took place. The column "Submatch" given in the datasets `vcapwide` and `vcaplong` are the submatches that occurred in the actual matching and randomization task when the experiment itself was performed. The week 10 information in connection to the submatches were:

```
> veh <- veh[order(veh[,"Submatch"]),]
> mdv <- mdv[order(mdv[,"Submatch"]),]
```

| | Veh.PSAWeek10 | MDV.PSAWeek10 |
|---|---|---|
| Submatch_1 | 21.30 | 16.25 |
| Submatch_10 | 8.13 | 7.55 |
| Submatch_11 | 8.36 | 13.53 |
| Submatch_12 | 15.05 | 9.96 |
| Submatch_13 | 7.03 | 6.57 |
| Submatch_14 | 34.72 | 27.16 |
| Submatch_16 | 9.13 | 10.78 |
| Submatch_2 | 13.17 | 13.13 |
| Submatch_3 | 22.29 | 23.58 |
| Submatch_4 | 14.27 | 9.74 |
| Submatch_5 | 13.74 | 9.59 |
| Submatch_6 | 8.49 | 9.89 |
| Submatch_7 | 16.29 | 19.51 |
| Submatch_8 | 20.99 | 16.39 |
| Submatch_9 | 14.69 | 18.71 |

Table 3: Submatches in the real VCaP experiment, per PSA at week 10 in tumors allocated to the Vehicle and MDV groups

Tables **??** and **??** show the prognostic baseline information in terms of PSA and body weight that is connected to the submatches. These submatches are used for pairing observations in comparing MDV intervention to the vehicle, which results in a noticeable increase in statistical power in even conventional tests:

| | Veh.BWWeek10 | MDV.BWWeek10 |
|---|---|---|
| Submatch_1 | 35.00 | 34.80 |
| Submatch_10 | 26.70 | 31.60 |
| Submatch_11 | 35.60 | 34.00 |
| Submatch_12 | 39.60 | 33.40 |
| Submatch_13 | 30.00 | 31.90 |
| Submatch_14 | 32.10 | 28.30 |
| Submatch_16 | 29.90 | 29.20 |
| Submatch_2 | 31.70 | 33.30 |
| Submatch_3 | 31.10 | 33.60 |
| Submatch_4 | 34.80 | 34.00 |
| Submatch_5 | 32.40 | 32.00 |
| Submatch_6 | 34.90 | 38.30 |
| Submatch_7 | 30.00 | 29.90 |
| Submatch_8 | 28.40 | 30.10 |
| Submatch_9 | 28.30 | 33.30 |

Table 4: Submatches in the real VCaP experiment, per body weight at week 10 in tumors allocated to the Vehicle and MDV groups

```
> t.test(veh[,"PSAWeek14"], mdv[,"PSAWeek14"], paired=TRUE)

        Paired t-test

data:  veh[, "PSAWeek14"] and mdv[, "PSAWeek14"]
t = 2.393, df = 13, p-value = 0.03251
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  1.895046 37.097811
sample estimates:
mean of the differences
               19.49643
```

For more refined statistical testing, the longitudinal profiles of a pre-clinical experiment can be modeled using the `lme4`-package. Functions are provided in the `hamlet`-package for this purpose.

List of useful functions:

- MEM-modeling packages: `lme4`, `lmerTest`, `nlme`
- MEM-modeling in `hamlet`: `hamlet::mem.getcomp`, `hamlet::mem.plotran`, `hamlet::mem.plotresid`