Example on using the tiger-package (TIme series of Grouped ERrors)

Dominik Reusser

March 26, 2009

1 Introduction

This document walks you step by step through the data analysis with the tiger package. For more information on the method, please see the relevant publication by Reusser et al. [2008]. In fact, the example presented here will reproduce the Weisseritz case study from [Reusser et al., 2008].

2 Data

In this example, we are looking at the difference between an observed river discharge time series and the model output from a hydrological model, simulating the river discharge from meteorolgical input data. The data is provided in the package and is shown in figure 1.

```
> library(tiger)
> data(tiger.example)
> measured <- tiger.res$measured
> modelled <- tiger.res$modelled
```

```
> plot(d.dates, measured, type = "l", col = "blue")
> lines(d.dates, modelled)
> legend("topright", legend = c("measured", "modelled"), lty = 1,
+     col = c("blue", "black"))
```

3 Doing the calculations

First of all, we will generate our synthetic peak errors which will help to better understand the error groups. The synthetic peak errors are shown in figure 2.

```
> peaks2 <- synth.peak.error(rise.factor = 2, recession.const = 0.02,
+     rise.factor2 = 1.5, err1.factor = c(1.3, 1.5, 2), err2.factor = c(0.02,
+         0.03, 0.06), err4.factor = c(9, 22, 40), err5.factor = c(0.2,
+         0.3, 0.5), err6.factor = c(2, 3, 5), err9.factor = c(1.5,
+         3, 6))
```

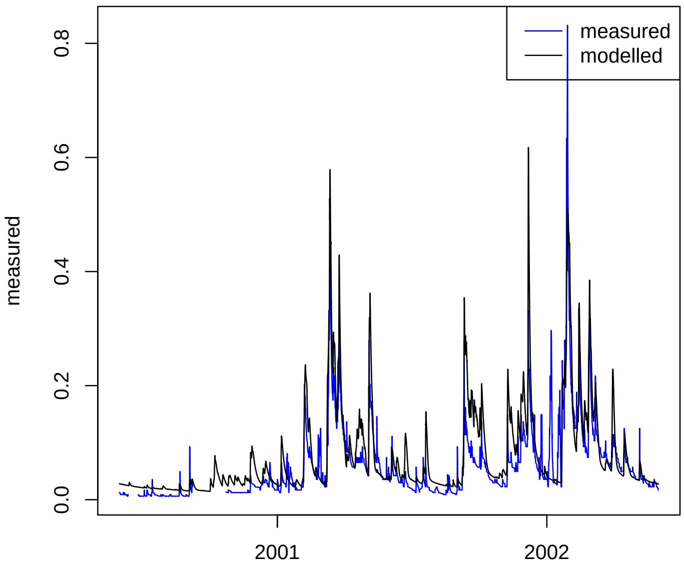

Figure 1: Measured and modelled river discharge.

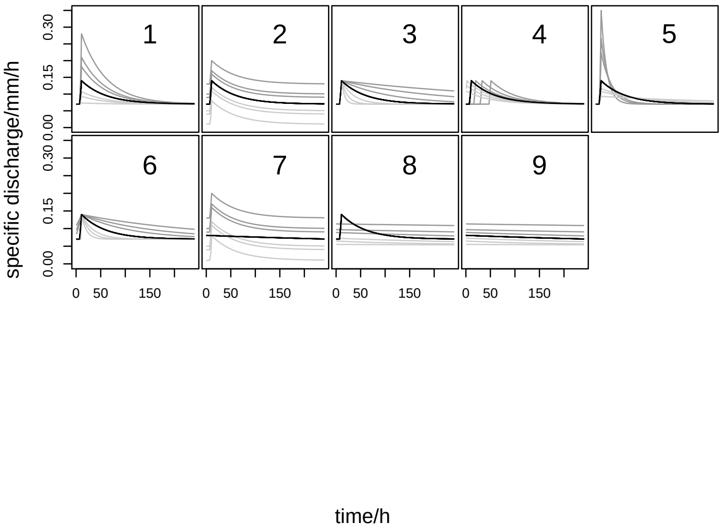

Figure 2: Synthetic peak errors.

```
[1] "Check if peak volumes are correct for 'volume-optimized'\npeaks:"
[1] "Volume for reference peak:  20.3895471993771"
[1] "Volumes for error peaks:"
[1] 20.38916 20.38863 20.38802 20.38132 20.38950 20.36542
```

The synthetic peak error number 5 overestimates the peak, but the total volume is kept correct. The recession constant is optimized to obtain a correct volume and the package asks you to check whether the optimization was successful.

The command bellow plots the synthetic peak errors.

```
> p.synth.peak.error(peaks2)
```

Then, we will call the function that does all the computation. The object returned (`result`) is equivalent to `tiger.res` provided in `data(tiger.example)`. This result object will then be further processed by plotting and summarizing methods.

```
> result <- tiger(modelled = d.qgko.calib, measured = d.ammelsdorf_interp,
+     window.size = 240, synthetic.errors = peaks2)
```

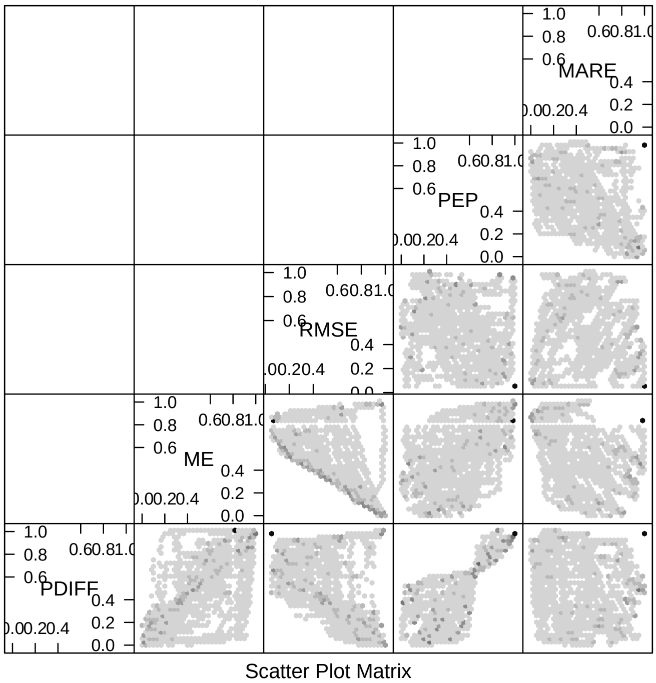

Figure 3: Scatter plots of the performance measures.

4 Assessing performance measures used to build error groups

About 50 performance measures are used to split the time series into error groups. Some of these performance measures are highly correlated and not of much interest for further interpretation. Therefore, we will exclude those from further plots. Note that we want to keep the CE and RMSE measures in any case. We will also create a scatter plot of the remaining measures to get an impression of their interdependence (Figure 3 - here, we are only showing the scatter plots for the first five measures).

```
> correlated <- correlated(result, keep = c("CE", "RMSE"))
> print(scatterplot(result$measures.uniform, show.measures = correlated$measures.uniform$t
+     2, 3, 5, 6)]))
```

To get an impression of how the performance measures react to the synthetic peaks, we can create a number of plots (figure 4). Nine plots show the response of some exemplary measures (y-axis) to the synthetic peak errors, each of which is shown with a different symbol. On the x-axis, no error would be in the centre and the severity of the error increases to each side. The variable `do.out` determines whether to exclude outliers from the plot.

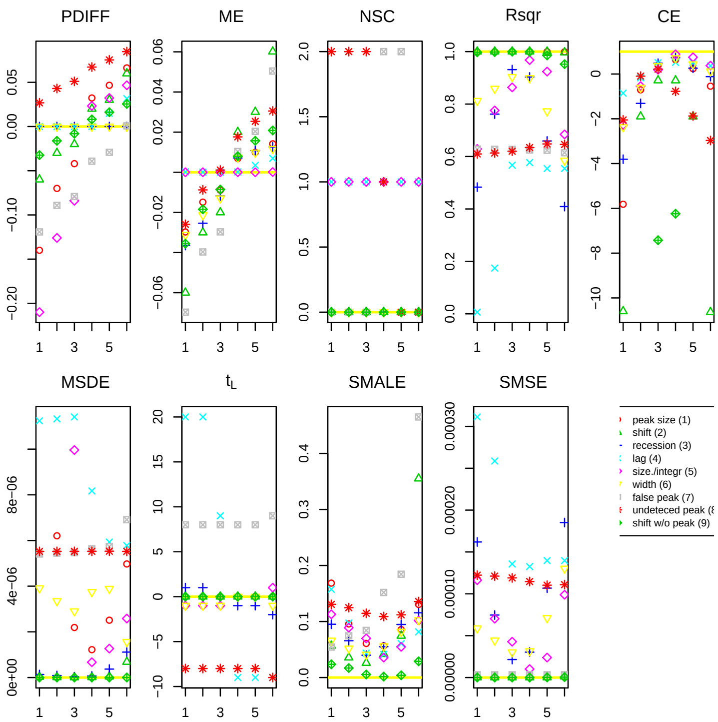

Figure 4: Performance measures for synthetic peak errors.

```
> show.measures <- which(names(result$measures) %in% c("CE", "PDIFF",
+     "ME", "MSDE", "SMSE", "Rsqr", "SMALE", "lagtime", "NSC"))
> peaks.measures(result, show.measures = show.measures, mfrow = c(2,
+     5), do.out = c(rep(FALSE, 4), TRUE, TRUE, rep(FALSE, 3)))
```

5 How many clusters to use?

In order to determine the optimum number of error groups during the c-means clustering, we try to minimize the validity index (figure 5).

```
> validity.max <- 10
> par(mar = c(4, 4, 1, 1) + 0.1)
> xmax <- 1
> while (any(result$validity[xmax:length(result$validity)] < validity.max)) {
+     xmax <- xmax + 1
+ }
> plot(result$validity[1:xmax], ylab = expression(V[XB]), xlab = "Number of clusters",
+     type = "b", lty = 2, ylim = c(0, validity.max))
```

The while loop determines the maximum number to plot on the x-axis.

The 2 cluster solution combines clusters A-C and D-F from the 6 cluster solution, while the 5 cluster solutions combines clusters B and D from the 6

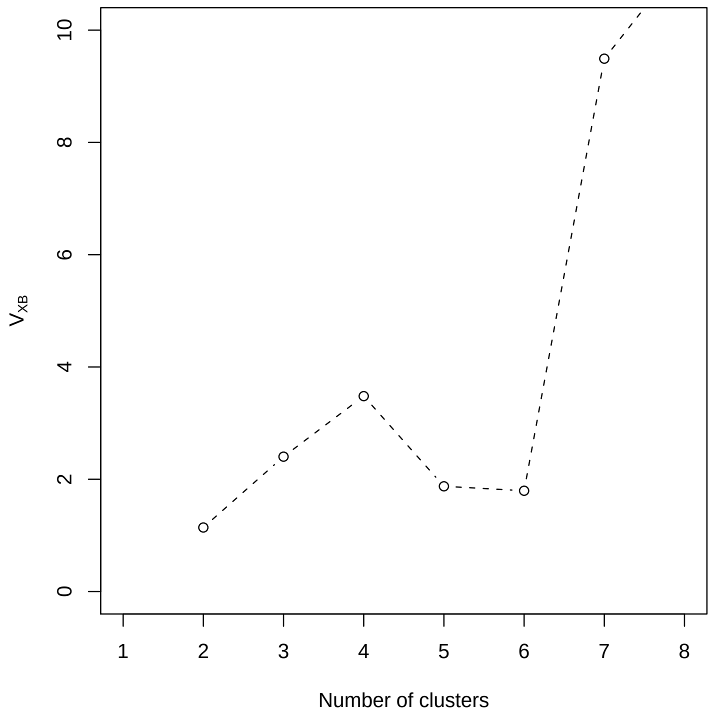

Figure 5: Validity index for the identification of the optimal cluster number for c-means clustering.

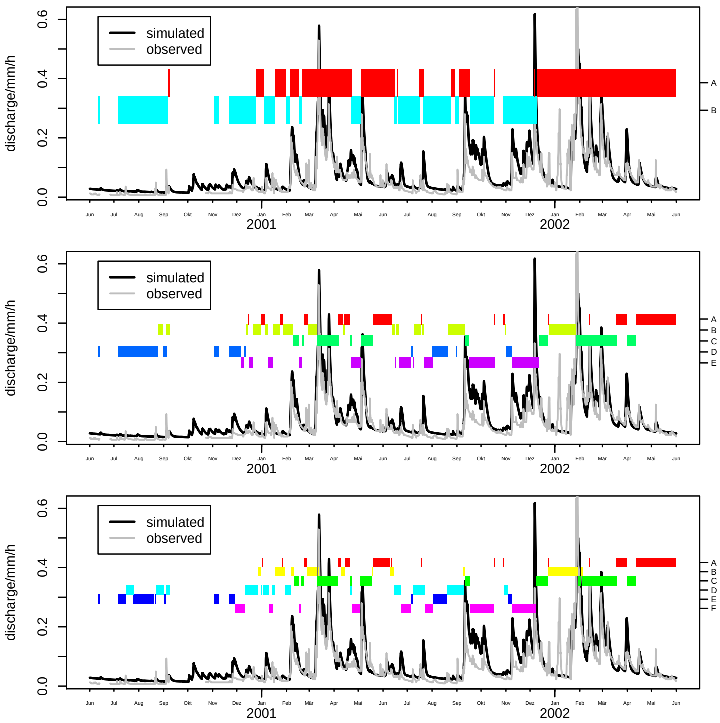

Figure 6: Simulated and observed discharge series. The colour bars indicate the error class during this time period. Plots are show for 2, 5, and 6 classes.

cluster solution (figure 6). Therefore the 6 cluster solution also represents the 2 and 3 cluster solutions. In this example, we are using the 6 group (cluster) solution.

```
> par(mfrow = c(3, 1), mar = c(2, 4, 1, 2) + 0.1)
> errors.in.time(d.dates, result, solution = 2, show.months = TRUE)
> errors.in.time(d.dates, result, solution = 5, show.months = TRUE,
+     new.order = c(3, 4, 5, 2, 1))
> errors.in.time(d.dates, result, solution = 6, show.months = TRUE,
+     new.order = c(4, 6, 5, 2, 1, 3))
> solutions <- 6
```

6 Time pattern of error groups

The temporal occurence of the error groups is shown in figure 7. The new.order parameter helps to reassign the color pattern, such that it is equivalent to the figures in [Reusser et al., 2008].

```
> new.order <- c(4, 6, 5, 2, 1, 3)
> par(mar = c(2, 4, 1, 2) + 0.1)
```

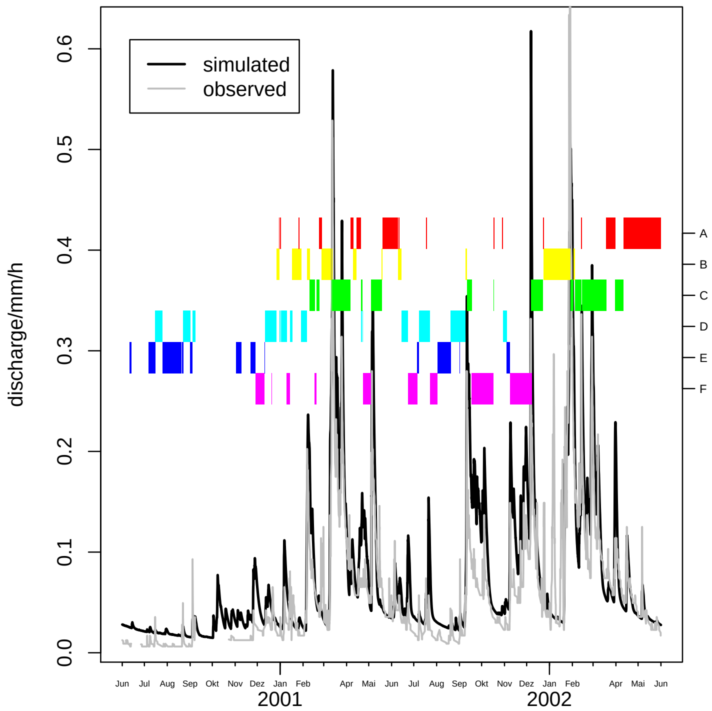

Figure 7: Simulated and observed discharge series. The colour bars indicate the error class during this time period.

```
> errors.in.time(d.dates, result, solution = solutions, show.months = TRUE,
+     new.order = new.order)
```

7 Characterizing the error groups

In order to characterize the error groups, we can check which of the synthetic peaks belong to which error groups. Note that the output is formated as LaTeXtables with & as the delimiter between columns and

as end of line delimiter.

```
> peaks.in.clusters(result, solution = solutions, new.order = new.order)
```

```
level & 1&2&3&4&5&6 \\
\hline
peak size (1) & C&A&A&A&A&A \\
shift (2) & F&A&A&A&A&D \\
recession (3) & F&A&A&A&A&A \\
lag (4) & B&A&A&A&A&A \\
size./integr (5) & A&A&A&A&A&A \\
width (6) & F&A&A&A&A&A \\
```

```
false peak (7) & F&F&F&A&D&D \\
undeteced peak (8) & C&A&A&A&A&B \\
shift w/o peak (9) & F&A&A&A&A&A \\


Cluster & Error & Level\\
A & peak size (1) & 2  3  4  5  6  \\
 & shift (2) & 2  3  4  5  \\
 & recession (3) & 2  3  4  5  6  \\
 & lag (4) & 2  3  4  5  6  \\
 & size./integr (5) & 1  2  3  4  5  6  \\
 & width (6) & 2  3  4  5  6  \\
 & false peak (7) & 4  \\
 & undeteced peak (8) & 2  3  4  5  \\
 & shift w/o peak (9) & 2  3  4  5  6  \\
B & lag (4) & 1  \\
 & undeteced peak (8) & 6  \\
C & peak size (1) & 1  \\
 & undeteced peak (8) & 1  \\
D & shift (2) & 6  \\
 & false peak (7) & 5  6  \\
F & shift (2) & 1  \\
 & recession (3) & 1  \\
 & width (6) & 1  \\
 & false peak (7) & 1  2  3  \\
 & shift w/o peak (9) & 1  \\
```

But also, we can check what the values of the performance measures in each cluster are. This is done with box plots as shown in figure 8. The plotting command also produces a summary table of the findings as described by Reusser et al. [2008].

```
> par(mfrow = c(4, 6), mar = c(2, 2, 3, 1) + 0.1)
> summary.table <- box.plots(result, solution = solutions, show.measures = correlated$meas
+     new.order = new.order)

> print(summary.table)

[1] "A & {\bf best:} PDIFF, ME, RMSE, PEP, MARE, CE, IoAd, PI, t_test, r[d], DE, r[k], RSM
[2] "B & {\bf best:} CE, PI, t_test, t[L]; {\bf worst:} NSC, Rsqr, MSDE, DE, MAOE, LCS, RS
[3] "C & {\bf best:} Rsqr, CE, IoAd, r[d], DE, MAOE, LCS; {\bf worst:} RMSE, MSDE; {\bf lo
[4] "D & {\bf best:} PDIFF, RMSE, PEP, PI, DE; {\bf worst:} Rsqr, t[L], r[d], MAOE, LCS; {
[5] "E & {\bf best:} RMSE, MSDE, t[L], DE, RSMSGE; {\bf worst:} MARE, Rsqr, CE, IoAd, PI, 
[6] "F & {\bf best:} Rsqr, t[L], r[d], DE, LCS, RSMSGE; {\bf worst:} RMSE, CE, PI; {\bf lo
```

References

D. E. Reusser, T. Blume, B. Schaefli, and E. Zehe. Analysing the temporal dynamics of model performance for hydrological

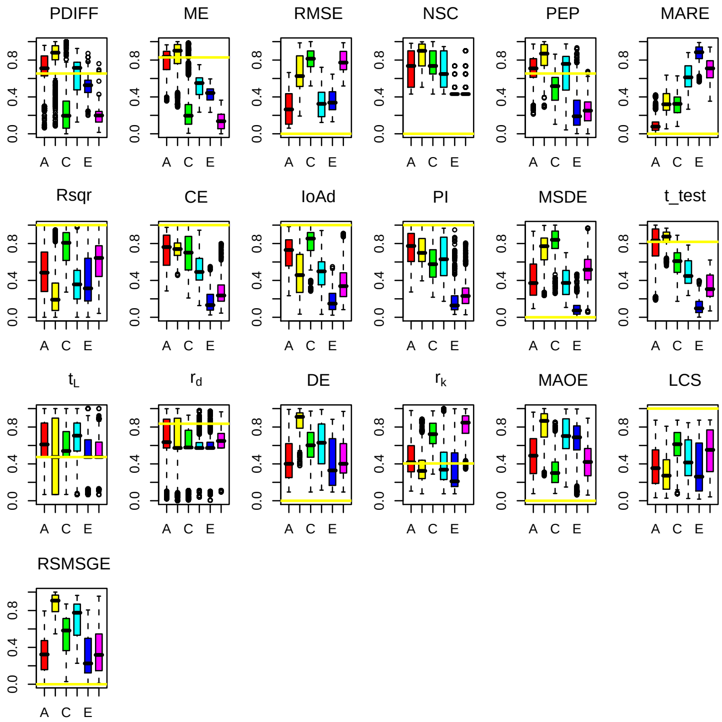

Figure 8: Matrix of box plots comparing the normalized performance measure values. The yellow line indicates the "perfect fit" for each of the performance measures.Simulated and observed discharge series. The colour bars indicate the error class during this time period.

models. *Hydrol. Earth Syst. Sci. Discuss.*, 5:1–43, 2008. URL `http://www.hydrol-earth-syst-sci-discuss.net/5/3169/2008/hessd-5-3169-2008.html`.