J Comput Aided Mol Des (2014) 28:1049–1055
DOI 10.1007/s10822-014-9759-6

SPECIAL SERIES: STATISTICS IN MOLECULAR MODELING Guest Editor: Anthony Nicholls

Count on kappa

Paul Czodrowski

Received: 8 January 2014 / Accepted: 31 May 2014 / Published online: 11 July 2014

Abstract In the 1960s, the kappa statistic was introduced for the estimation of chance agreement in inter- and intra-rater reliability studies. The kappa statistic was strongly pushed by the medical field where it could be successfully applied via analyzing diagnoses of identical patient groups. Kappa is well suited for classification tasks where ranking is not considered. The main advantage of kappa is its simplicity and the general applicability to multi-class problems which is the major difference to receiver operating characteristic area under the curve. In this manuscript, I will outline the usage of kappa for classification tasks, and I will evaluate the role and uses of kappa in specifically machine learning and cheminformatics.

Keywords Kappa statistics · Cheminformatics · Machine learning · Classification

Introduction

When generating classification models, it is helpful to compare the generated models with a random model. By such means, it is possible to deduce if the modeling attempts are better than, equal to or even worse than random. The kappa statistic addresses this question and gives a numerical value to compare the trained model with a random model. Originally, the kappa statistic was published in the context of inter- and intra-rater reliability relationship [1]. It can be easily generalized to any classification problem and is therefore perfectly suited for classification tasks in the discipline of cheminformatics.

In the current publication, I will outline the strengths and the limitations of the kappa statistic. Real-world examples based on classification models and synthetic data will be given and the performance of kappa statistic will be explained in depth. Any program or algorithm mentioned throughout this manuscript is implemented in open source tools which technically enables any interested reader to re-run the in silico experiments by himself/herself.

P. Czodrowski (✉)
Merck KGaA, Frankfurter Strasse 250, 64293 Darmstadt, Germany
e-mail: paul.czodrowski@merckgroup.com

Method

A typical use case in the daily work life of a cheminformatician/molecular modeler is the training of activity models. It is often the case that simple binary classification models are sufficient for the raised question, e.g. is this compound active or inactive on target XYZ. The standard format to express the output of such classification tasks is the so-called confusion matrix. It relates the predicted classes with the experimentally determined classes. An example confusion matrix for a binary classification task is given in Table 1.

The successful predictions can be found on the diagonal of the matrix: the values true positive (TP) and true negative (TN) reflect the correct prediction of active and inactive compounds. If the ratio of the sum of those two (i.e. TP + TN) in relation to the overall sum of values (N) is formed, one has derived the most basic quality measure for classification tasks, the accuracy:

$$accuracy = \frac{TN + TP}{N}$$

If a "perfect" model was trained, the accuracy would be 1.00. In such a case, there would no false negatives (FN) or

Table 1 Typical confusion matrix

	Prediction		Total
	Inactive compound	Active compound	
Experiment			
Inactive compound	TN (true negative)	FP (false positive)	a1x
Active compound	FN (false negative)	TP (true positive)	a2x
Total	ax1	ax2	N

Table 2 Typical quality measures for classification problems

recall	Recall is also named sensitivity and true positive rate (TPR). It reveals how many of the positives are missed by the prediction. The "missed" predictions are expressed as false negatives (FN): those data points are actually positive, but they are predicted as negative.
precision	The precision considers the false positives (FP): Out of all predictions of the positives, it reveals the proportion of the "real" positives/true positives (TP).
specificity	Specificity is similar to recall, the only difference is the focus on the true negatives (TN).

false positives (FP). Since such cases rarely occur, other measures need to be used. Table 2 gives an overview of these measures.

Another commonly used statistical measure is ROC-AUC: it corresponds to the area under the curve (AUC) of a receiver operating characteristic (ROC). The ROC is obtained by plotting (1-specificity) on the x-axis, whereas sensitivity is plotted on the y-axis. Thus, there is an indirect relationship between the ROC-AUC and the actual confusion matrix. Although ROC-AUC accounts for a potential success by random guesses, the advantage of kappa is its straightforward application to classifications which are more than bimodal.

Such a measure was introduced already in 1960 by Jacob Cohen: it is called Cohen's kappa. The formula is as follows:

$$kappa = \frac{accuracy - baseline}{1 - baseline}$$

Given the example confusion matrix from Table 1, baseline is calculated as follows:

$$baseline = \sum_{i=1}^{k} \frac{a_{ix} \cdot a_{xi}}{N^2}$$

The intriguing aspect of kappa statistic in comparison to ROC-AUC is the inherent capability to be directly applicable to multiple classes (>2) which can be seen from the definition of the baseline.

Table 3 Different scenarios for the extreme outer values of kappa and complete randomness: (a) complete agreement (b) complete disagreement (c) random agreement/random disagreement (d) majority class classifier

	Prediction		Total
	Inactive compound	Active compound	
(a) Complete agreement			
Experiment			
Inactive compound	α	0	α
Active compound	0	β	β
Total	α	β	$\alpha + \beta$
(b) Complete disagreement			
Experiment			
Inactive compound	0	α	α
Active compound	β	0	β
Total	β	α	$\alpha + \beta$
(c) Random agreement/random disagreement			
Experiment			
Inactive compound	α	α	2α
Active compound	α	α	2α
Total	2α	2α	4α
(d) Majority class classifier			
Experiment			
Inactive compound	0	71	71
Active compound	0	623	623
Total	0	694	694

What kappa simply does is compute the ratio between the chance-corrected agreement of the accuracy in the numerator and the chance-corrected perfect agreement in the denominator. This ratio yields an estimate of how much better the actual agreement is over chance agreement .

The values for kappa range between -1 and $+1$: a perfect model gives a kappa value of $+1$, whereas kappa values lower than 0 indicate models performing worse than random. The following examples have been taken from a publication by Ben-David [2]. The perfect agreement can be seen in Table 3(a). Kappa is calculated as follows for this scenario:

$$accuracy = \frac{\alpha}{\alpha + \beta} + \frac{\beta}{\alpha + \beta} = 1.00$$

$$baseline = \frac{\alpha^2}{(\alpha + \beta)^2} + \frac{\beta^2}{(\alpha + \beta)^2} = \frac{\alpha^2 + \beta^2}{(\alpha + \beta)^2} \geq 0$$

$$kappa = \frac{1 - baseline}{1 - baseline} = 1.00$$

If experiment and theory are in total disagreement (see Table 3(b)), then:

$$accuracy = \frac{0}{\alpha + \beta} = 0.00$$

$$baseline = \frac{\alpha\beta}{(\alpha+\beta)^2} + \frac{\alpha\beta}{(\alpha+\beta)^2} = \frac{2\alpha\beta}{(\alpha+\beta)^2}$$

$$kappa = \frac{0 - baseline}{1 - baseline} = \frac{0 - \frac{2\alpha\beta}{(\alpha+\beta)^2}}{1 - \frac{2\alpha\beta}{(\alpha+\beta)^2}} = \frac{-2\alpha\beta}{\alpha^2 + \beta^2}$$

when $\alpha = \beta$, then kappa $= -1.00$.

Given a random scenario (see Table 3(c)), then:

$$accuracy = \frac{2\alpha}{4\alpha} = 0.50$$

$$baseline = \frac{2\alpha \cdot 2\alpha \cdot}{(4\alpha)^2} + \frac{2\alpha \cdot 2\alpha \cdot}{(4\alpha)^2} = \frac{8\alpha^2}{16\alpha^2} = 0.50$$

$$kappa = \frac{0.5 - 0.5}{1 - 0.5} = 0.00$$

This shows the power of kappa: a random "prediction" gives a kappa value of 0.00. Thus, any sophisticated model ending up with a kappa value of 0.00 is no better than random!

In Table 3(d), one finds an example of a simple majority class classifier. The actual "prediction" is based on the preponderant class. In this example more active than inactive compounds are inside the dataset, and therefore the calculation is as follows:

$$accuracy = \frac{623}{694} = 0.90$$

$$baseline = \frac{0 \times 71}{694^2} + \frac{694 \times 623}{694^2} = 0.90$$

$$kappa = \frac{accuracy - baseline}{1 - baseline} = \frac{0.90 - 0.90}{1} = 0.00$$

Again, kappa gives a value of 0.00: the majority class classifier is neatly uncovered by the kappa statistic!

It is possible to derive the error bar for kappa. In the original publication by Jacob Cohen [1], the error estimation was not described. It took 9 years until Fleiss, Cohen and Everitt [3] derived the error estimation for kappa. The formula for the variance of kappa is as follows:

$$var(kappa) = \sum_{i=1}^{k} \frac{1}{N(1 - baseline)^2} \left\{ \sum_{i=1}^{k} a_{xi} a_{ix} [1 - (a_{xi} + a_{ix})^2] + \sum_{i=1}^{k} \sum_{j=1, i \neq j}^{k} a_{xi} a_{jx} (a_{xi} + a_{jx})^2 - baseline^2 \right\}$$

Landis and Koch [4] derived an interpretation scheme for kappa which can be found in Table 4.

Table 4 Interpretation of kappa values [4]

< 0.00	Poor
0.00–0.20	Slight
0.21–0.40	Fair
0.41–0.60	Moderate
0.61–0.80	Substantial
>0.81–0.99	Almost perfect
1.00	Perfect

Table 5 Synthetic data to introduce the concept of prevalence (for the calculation of prevalence, the diagonal elements are needed and therefore highlighted)

	Prediction		Total
	Inactive compound	Active compound	
(a) kappa = 0.6, accuracy = 0.80, precision = 0.80, recall = 0.80			
Experiment			
Inactive compound	**40**	10	50
Active compound	10	**40**	50
Total	50	50	100
(b) kappa = 0.37, accuracy = 0.80, precision = 0.50, recall = 0.50			
Experiment			
Inactive compound	**70**	10	80
Active compound	10	**10**	20
Total	80	20	100

Weighted kappa

This variant of kappa can be used to penalize some misclassifications over others [5]:

$$kappa_{weighted} = \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} w_{ij} a_{ij} - \sum_{i=1}^{k} \sum_{j=1}^{k} w_{ij} a_{ix} a_{xi}}{1 - \sum_{i=1}^{k} \sum_{j=1}^{k} w_{ij} a_{ix} a_{xi}}$$

In other terms, the weighted kappa can be seen as a cost-sensitive classifier.

What's in my data model: prevalence and bias

The single assessment of the kappa value may be misleading, which was pointed out by Feinstein and Cicchetti [6] and by Byrt, Bishop and Carlin [7]. I will outline some of their examples to introduce the basic concepts.

In Table 5, two example confusion matrices are shown. In both cases, accuracy amounts to 0.80. For the example in Table 5(a), precision and recall amount to 0.80. For the example in Table 5(b), precision and recall are 0.50. Based on these values, one would judge the models as almost equally good. If the astute reader then calculates the kappa values, he or she might be puzzled by the fact that these

Table 6 Synthetic data to introduce the concept of bias for the calculation of bias, the off-diagonal elements are needed and therefore highlighted)

	Prediction		Total
	Inactive compound	Active compound	
(a) kappa = 0.17, accuracy = 0.60, precision = 0.50, recall = 0.50			
Experiment			
Inactive compound	40	**20**	60
Active compound	**20**	20	40
Total	60	40	100
(b) kappa = 0.26, accuracy = 0.60, precision = 0.33, recall = 1.00			
Experiment			
Inactive compound	40	**40**	80
Active compound	**0**	20	20
Total	40	60	100

values differ substantially. For Table 5(a), kappa amounts to 0.60, whereas it amounts to 0.37 for Table 5(b).

In order to explain such a scenario, Byrt, Bishop and Carlin introduced the concept of prevalence [7]. The prevalence accounts for the differences between the overall proportion of "Yes" and "No" assessments in the context of inter- and intra-rater relationship. Certainly, this can be directly transferred to the overall proportion of "active compound" and "inactive compound" assessment. Alternatively, this can be re-formulated as the degree of "balancedness" or "im-balancedness" of a dataset.

The prevalence is calculated as follows:

$$prevalence = \frac{|TP - TN|}{N}$$

For the two examples in Table 5, prevalence is 0.00 (Table 5(a)) and 0.60 (Table 5(b)). The rather poor performance of the model in Table 5(b) is directly reflected by the large prevalence inherent in the distribution of the data set.

Another scenario might occur, in which the observers largely disagree. Or to state it for the cheminformaticians amongst us, experiment and model deviate to a large extent. Synthetic data for this scenario is given in Table 6.

In both models, accuracy amounts to 0.60. Recall and precision deviate to some extent for both models. A mere look at the confusion matrices shows that the model in Table 6(b) is biased towards the prediction of active compounds. The index expressing such a situation is named bias, accordingly and calculated as follows:

$$bias = \frac{|FP - FN|}{N}$$

For the two examples in Table 6, bias is 0.00 (Table 6(a)) and 0.40 (Table 6(b)).

Prevalence and bias can only be calculated for 2-class models.

Byrt, Bishop and Carlin [7] and Lantz and Nebenzahl [8] introduced the concept of PABAK (**p**revalence **a**djusted and **b**ias **a**djusted **k**appa):

$$\begin{aligned} PABAK &= 2p_0 - 1 \\ &= k\left(1 - prevalence^2 + bias^2\right) + prevalence^2 \\ &\quad - bias^2 \end{aligned}$$

Kappa is related to PABAK by the following formula [7]

$$kappa = \frac{PABAK - prevalence^2 + bias^2}{1 - prevalence^2 + bias^2}$$

However, Hoehler states that this index is unsuitable [9]. Hoehler rather recommends to set kappa into the context of sensitivity and specificity. In my opinion, prevalence and bias are more a feature than a bug: they are helpful metrics. They relate the obtained models with the composition of the data set and a potential bias for one class.

How to practically calculate kappa

Kappa is implemented in the Python module pystatsmodels [10], for an exemplary iPython notebook see Fig. 1. Kappa is also implemented in the program weka [11]. There are several R implementations such as irr [12] or PresenceAbsence [13].

```
In [1]: import statsmodels.cohenskappa as irr
        import numpy as np

In [2]: bla = np.asarray([[12,28],[18,42]])

In [3]: print irr.cohens_kappa(bla)
        Simple Kappa Coefficient
        --------------------------------
        Kappa 0.0000
        ASE 0.0976
          95% Lower Conf Limit -0.1913
          95% Upper Conf Limit 0.1913

        Test of H0: Simple Kappa = 0

        ASE under H0 0.0976
        Z 0.0000
        One-sided Pr > Z 0.5000
        Two-sided Pr > |Z| 1.0000
```

Fig. 1 Example iPython notebook demonstrating how to calculate kappa. In addition, kappa can be calculated online via http://caddstat.eyesopen.com/kappa/

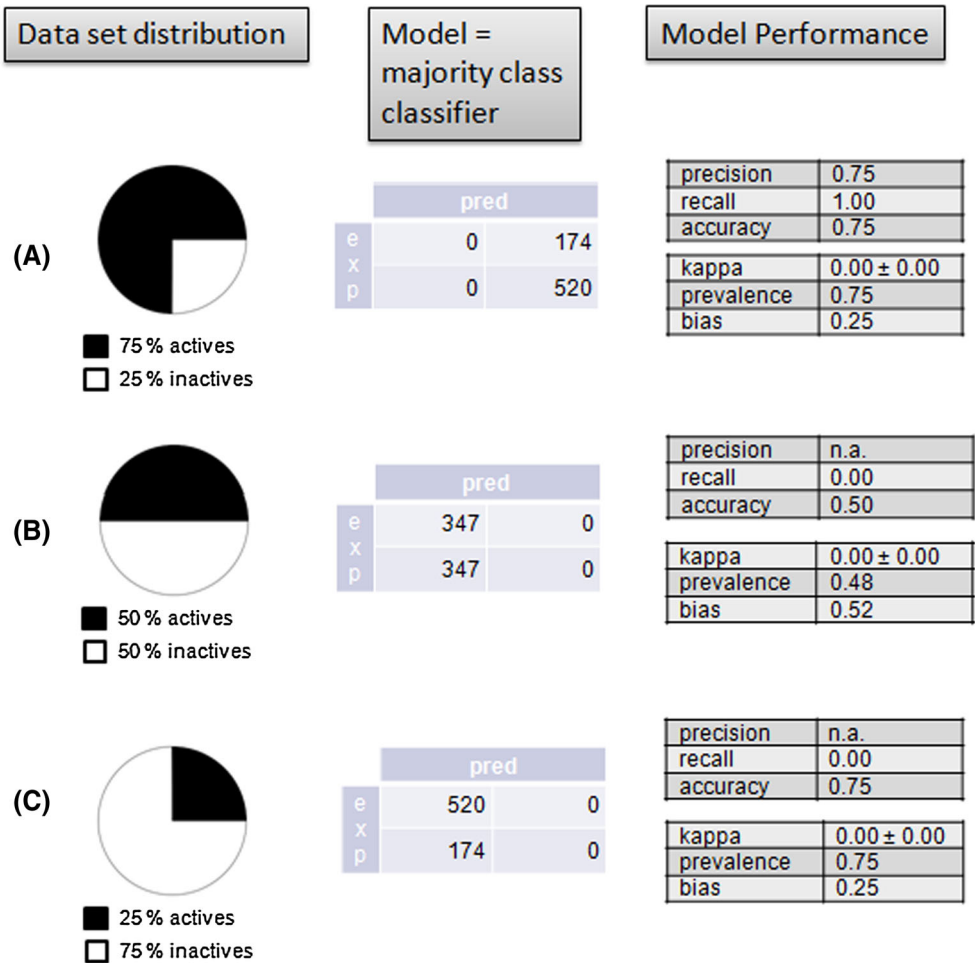

Fig. 2 What if… just the majority class classifier would have been used? Three different scenarios—given different distributions of actives and inactives—are considered (*left part*). The majority class classifier always predicts the preponderant class. (Table in the *middle part*). The statistical measures are given in the table on the right hand side

Results and discussion

Let's examine a simple experiment: A binary classification is done for the distinction between active and inactive compounds. In Fig. 2, more examples on the majority class classifier can be found. In Table 3(d), the kappa value was already calculated for one such scenario.

For the three examples in Fig. 2, the other statistical measures (accuracy, precision and recall) are also given. It becomes apparent from Fig. 2, that for all such majority class classifiers, kappa is 0.00. This also holds true for the case in which there is a tied vote.

It is worthwhile to describe example (A) from Fig. 2 in more detail: All three "non-kappa" measures (accuracy, precision and recall) indicate that the performance of the model is strong. Here, the power of kappa can be grasped: one learns from the kappa value of 0.00 that this model is useless. The fact that prevalence shows a large value should not be over-interpreted. Given this synthetic data set, the large prevalence values indicates the large inactive/active ratio.

It should be mentioned that the example from Table 3(d) also has high values for accuracy, precision and recall: 1.00, 0.90 and 1.00. As shown above, the kappa value is 0.00 indicating a pure agreement by chance. This is the real power of kappa!

For the other two examples, the accuracy indicates that there is at least some "signal" in the majority class classifier. In both cases, recall amounts to 0.00, which already indicates the poor performance of the majority class classifier. However, one should not forget that recall was completely misleading for the two above discussed examples!

The value of bias for the examples from Fig. 2(B) and (C) indicates the difference of the marginal totals (the sum of columns and the rows; ax1, ax2, a1x and a2x in the sample confusion matrix in Table 1). In the case of the example from Fig. 2(B), the consideration of prevalence is misleading, since the actual data is equally distributed. Bias and prevalence are not helpful for the further interpretation of kappa: since kappa is 0.00, this is actually not necessary.

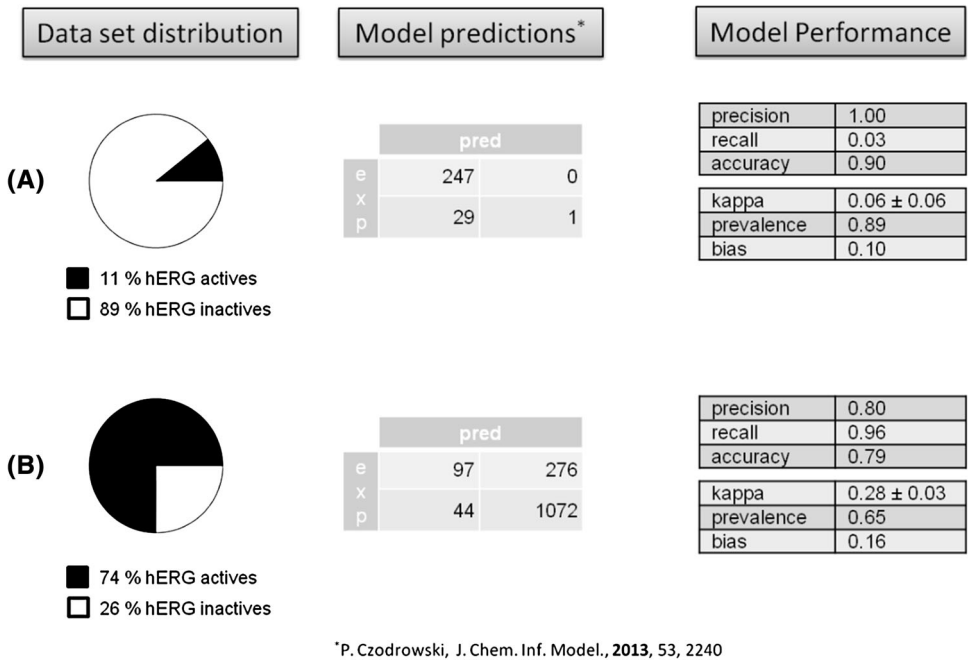

Fig. 3 Two real-world examples [15] in which accuracy, precision and/or recall are misleading

The majority class classifier is implemented in Weka [11] and in scikit-learn [14] (two popular machine learning tools): it is called "Zero R" in Weka and "dummy classifier" in scikit-learn.

- In a recent publication on hERG (human ether-à-go–go-Related Gene) classification models [15], several models were trained. We will inspect two of these models in more detail. The examples indicate that it is necessary to include kappa into considerations for the quality of the model (see Fig. 3). In example (A), the model acts almost like a majority class classifier and predicts nearly all compounds with the stronger populated class. Such a prediction appears reasonable from a probabilistic perspective: the likelihood of being a hERG inactive compound is high. This is also reflected by the large value for prevalence, it amounts to 0.89. However, the actual model performs bad in the prediction of the hERG actives: only one out of 30 hERG active compounds is correctly predicted. Accordingly, the kappa value is 0.00. Although the low value of recall already points into the right direction of interpretation of the model, the sole consideration of accuracy and precision would be misleading.
- In the case of example (B), the precision, recall and accuracy values evaluate this model as showing a reasonable predictive power. But is such a model significantly better than random? Although the kappa value at least has a positive sign, it is still only a fair model (kappa is between 0.21 and 0.40). The prevalence value indicates that the data set has some inherent mis-proportion. The consideration of prevalence can therefore be consulted for the analysis of the dataset and it can be used as an indicator of the difficulty to train a model performing equally well for both classes.

Conclusions

It was shown that the kappa statistic is a helpful measure for the interpretation of classification models. Based on simple examples, the added value of kappa versus other statistical measures such as accuracy, precision and recall could be shown. By means of kappa, it becomes clear how much better the trained model is over the random model.

I also pointed out that kappa needs to be carefully inspected. Based on an interpretation of kappa, one can learn about the performance gain over random guessing. The two indices bias and prevalence are helpful for an additional and independent examination of the data set. These indices are valuable when it comes to verify the model performance given the composition of the data set.

It can be concluded that the kappa statistic should be included in the toolbox of every cheminformatician.

Acknowledgments I thank Christian Kramer (University of Innsbruck, Austria) for critical proof-reading, making useful suggestions and the discussions initiated by this manuscript and my GRC talk. Furthermore, the fantastic assistance by Kim Branson (Hessian Informatics, San Francisco, USA) is acknowledged. Without Kim, this paper and my GRC talk would have been less instructive. I would also like to thank Georgia McGaughey (Vertex Pharmaceuticals, Boston, USA) for her intense proof-reading. Lastly, I would like to

express my deepest gratitude to Anthony Nicholls (OpenEye Scientific Software, Santa Fe, USA) who reviewed the initial GRC contribution and this manuscript in great detail: this was really a heroic effort!.

References

1. Cohen J (1960) Edu Psychol Meas 20:37–46
2. Ben-David A (2008) Expert Syst Appl 34:825–832
3. Fleiss JL, Cohen J, Everitt BS (1969) Psychol Bull 72:323–327
4. Landis JR, Koch GG (1977) Biometrics 33:159–174
5. Fleiss JL (1981) Statistical methods for rates and proportions, (2nd ed.)Wiley: New York
6. Feinstein AR, Cicchetti DV (1990) J Clin Epidemiol 43:543–549
7. Byrt T, Bishop J, Carlin JB (1993) J Clin Epidemiol 46:423–429
8. Lantz CA, Nebenzahl E (1996) J Clin Epidemiol 49:431–434
9. Hoehler FK (2000) J Clin Epidemiol 53:499–503
10. pystatsmodels https://github.com/yarikoptic/pystatsmodels (accessed Dec 8, 2013)
11. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, I. H. W. *SIGKDD Explor.* 2009, *11*
12. irr R package http://cran.r-project.org/web/packages/irr/index.html (accessed Dec 8, 2013)
13. PresenceAbsence R package http://cran.r-project.org/web/packages/PresenceAbsence/index.html (accessed Dec 8, 2013)
14. Pedregosa F, Weiss R, Brucher M (2011) J Mach Learn Res 12:2825–2830
15. Czodrowski P (2013) J Chem Inf Model 53:2240–2251