Supplement of Biogeosciences, 21, 1827–1846, 2024
https://doi.org/10.5194/bg-21-1827-2024-supplement

Biogeosciences Open Access EGU

Supplement of

Technical note: Flagging inconsistencies in flux tower data

Martin Jung et al.

Correspondence to: Martin Jung (mjung@bgc-jena.mpg.de)

S1 Summary table of constraints

Constraints	GPP_NT	GPP_DT	RECO_NT	RECO_DT	NEE	LE	H	NETRAD	SW_IN	PPFD_IN	TA	VPD
Machine learning	S	S	S	S	S	S	S	S	S	S	S	S
u*					S							
GPP_NT vs GPP_DT	H	H										
[1]GPP_NT*sqrt(VPD) vs LE	S					S						S
[1]GPP_DT*sqrt(VPD) vs LE		S										
RECO_NT vs RECO_DT			H	H								
RECO_NT_NIGHT vs NEE_NIGHT			H									
RECO_DT_NIGHT vs NEE_NIGHT				H								
NETRAD vs LE+H						S	S	S				
[2]NETRAD vs SW_IN								S	S			
[2]NETRAD vs PPFD_IN								S		S		
SW_IN vs PPFD_IN									H	H		
TA vs TA ERA-5											S	
VPD vs VPD ERA-5												S
# constraints for samples (within site)	3	3	3	3	2	3	2	4	3	3	2	3
# constraints for variable (between sites)	4	4	4	4	3	4	3	5	4	4	3	4

Table S1: List of hard (H) and soft (S) constraints. [1] excluding rain days. [2] excluding NETRAD < 0.

S2 Accounting for heteroscedasticity in the outlier score

We developed a non-parametric method to estimate the heteroscedastic behaviour of residuals that is capable of handling heterogenous patterns of heteroscedasticity found for different constraints and sites, and the typically very skewed data distribution of Y^{pred}.

(1) We first calculate the residuals R_i and sort them ascending according to Y^{pred}.

(2) We calculate the 25th and 75th percentile (P25, P75) of R included in a moving (non-overlapping) step window (see Fig. SI-2.1 bottom right panel). Default window size is 100 daily data points, which is reduced if necessary to yield at least 15 estimates for P25 and P75 if the time series is short. This yields vectors of $Y^{pred}{}_w$ (mean), $P25_w$, and $P75_w$ which are typically shorter than the original R and Y^{pred} vectors by a factor of 100. We observed that the estimated interquartile range ($iqr_w = P75_w - P25_w$) is occasionally (very close to or) zero, for example when flux partitioning results were truncated at 0 causing long consecutive periods of zero flux in winters for instance. In these rare cases of extremely small interquartile range of residuals the outlier score could become very large even for a tiny absolute residual due to dividing by a number close to zero. To counteract this we imposed a minimum iqr_w, i.e. we take the max(iqr_w, iqr_{min}) and adjust P75 and P25 accordingly if necessary. iqr_{min} was chosen to be 7% of the standard deviation of Y^{pred} – this heuristic choice is based on empirical trials and visual inspections.

(3) Since the step window approach does not provide an estimate for P25 and P75 for the smallest and largest values of Y^{pred}, linear regression is used to extrapolate $P25_w$ and $P75_w$ for the smallest and largest values in Y^{pred}. The linear regression uses the smallest or largest 10% of $Y^{pred}{}_w$ respectively, and corresponding $P25_w$, and $P75_w$ obtained from step (2) and at least 5 data points. The linear regression then estimates $P25_w$ and $P75_w$ for the smallest and largest value of Y^{pred} and those values are inserted in the respective vectors (see e.g. most right data point plotted in bottom right panel of Fig. SI-2.1).

(4) Since the empirical estimation of $P25_w$ and $P75_w$ is typically noisy we perform a lowess (locally weighted scatterplot smoothing, (Cleveland and Devlin, 1988)) filtering with a span of 20% of the length of the vector and by also specifying $Y^{pred}{}_w$ as the corresponding exogenous variable locations. This yields a smooth variation of how $P25_w$ and $P75_w$ vary with $Y^{pred}{}_w$ (see plotted lines in bottom right panel of Fig. S1).

(5) The smoothed versions of $P25_w$ and $P75_w$ at locations $Y^{pred}{}_w$ are linearly interpolated at the locations of the original, full, Y^{pred} vector to obtain the finally required $P25_i$ and $P75_i$.

The presence of outliers can inflate the estimated interquartile range of residuals and could result in false negatives, in particular if the distribution of outliers is in some way systematic with the magnitude. To counteract this, steps (2) to (5) are repeated several times and outliers detected in the current iteration (defined as exceeding nIQR=3) are masked out for the calculations of the next iteration. The iteration stops when a stable set of outliers are found.

The procedure described above has been developed to obtain a reasonable solution with tractable computational costs since the calculation of the heteroscedastic outlier score needs to be done for each site and constraint several times. Because the data adaptive and non-parametric estimation of percentiles also accounts for systematic changes of the median residual with predicted magnitude it effectively relaxes the linearity assumption of bivariate constraints. For some constraints this is an advantage (e.g. for LE vs GPP*sqrt(VPD)), while for others it is a conceptual disadvantage (e.g. for NETRAD vs LE+H).

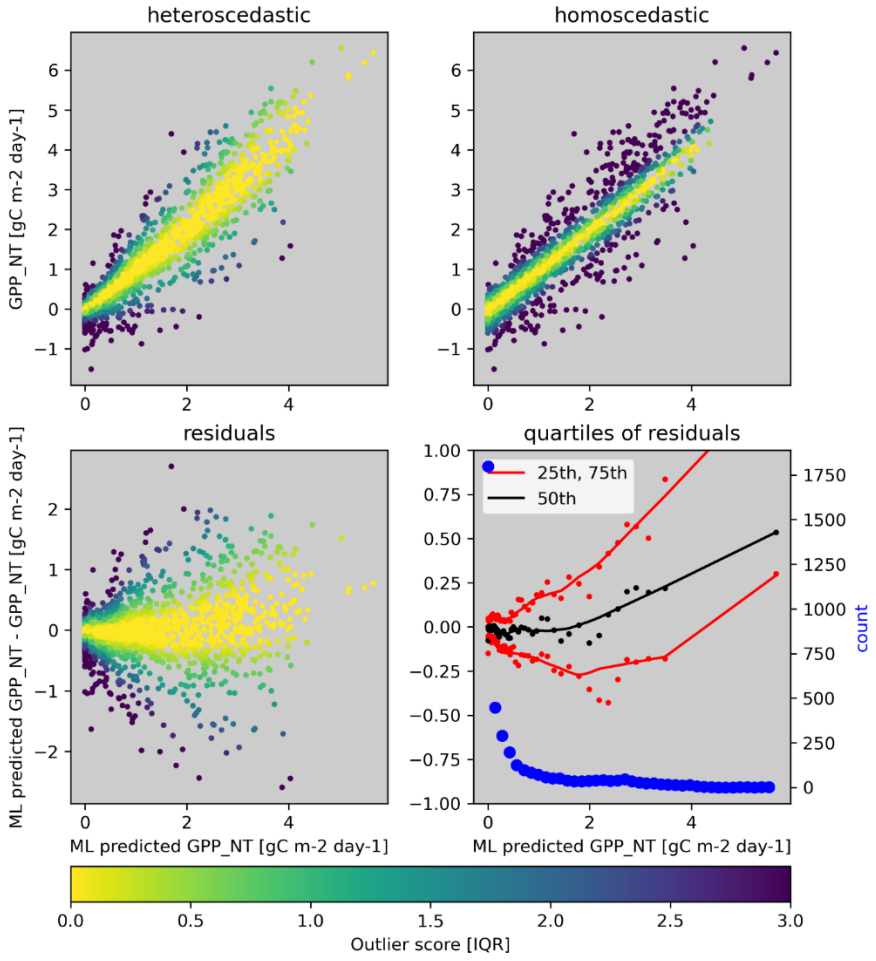

Fig. S1: Illustration of the behavior and calculation of the heteroscedastic outlier score for the machine learning constraint for GPP_NT for US-Wkg (see also Fig. 1). Top row shows the effect of accounting for heteroscedastic residuals for the outlier score (on colour). Bottom left illustrates how the variability of residuals vary with magnitude. Bottom right shows the empirically estimated quantiles by the step window (dots) and the smoothed and interpolated estimates for each data point (lines). The blue dots show the frequency distribution of data points. The example corresponds to nIQR=3.

S3 Implementation of machine learning constraints

Inputs are a set of specified and gap-filled predictor variables, and the target variable Y of the constraint (not gap-filled). The predictor sets (Table 3) were chosen to increase independence among different constraints, e.g. that variables used in bivariate constraints for the same target variable are excluded from the predictor set (e.g. NETRAD and PPFD_IN are not used to model SW_IN with machine learning). The outputs are 1) the outlier score for each data point i which is used in the within-site processing, and 2) robust estimates of correlation and root mean squared error later used for flagging site-variables (see section 2.2.5 and S7). The algorithm is:

(1) Run a cross-validation with a machine learning algorithm to obtain the cross-validated predictions Y^{pred}. We chose a 3-fold cross-validation with random partitioning, and the scikit-learn implementation of Random Forests (Breiman, 2001) with default parameters as machine learning model.
(2) Calculate the outlier score O^{ML} based on Y and Y^{pred} following equation 1.
(3) Calculate correlation and RMSE from Y and Y^{pred} with outliers removed (i.e. where $O^{ML} < 3IQR$).

Steps (1)-(2) are iterated a few times where outliers identified in the current iteration are masked out in the training data for the next iteration.

Missing values in the predictor variables of the training data set were imputed using missForest (Stekhoven and Bühlmann, 2011), which is an iterative gap-filling procedure for a set of variables based on Random Forests. This was done to maximize data availability and applicability. We calculated a set of water balance indicators from daily gap-filled precipitation (P) and evapotranspiration (E) to improve predictability for water-stressed conditions, especially because measured soil moisture contents are provided inconsistently. Those are added as predictors whenever measured soil moisture content was specified as predictor. These are cumulative water deficit (CWD), truncated cumulative water deficit (tCWD) with storage capacities of 15,50,100,150,200,250 mm, and detrended cumulative water balance (CWB):

$CWD_t = \min(CWD_{t-1}+P_t-E_t,0)$

$tCWD_t^C = \max(0,\min(tCWD_{t-1}+P_t-E_t,C))$

$CWB_t = CWB_{t-1}+P_t-E_t$ (detrended)

where C is the specified storage capacity.

S4 Implementation of bivariate constraints

Inputs are two variables, Y1 and Y2, and the outputs are 1) an outlier score for each data point i which is used in the within-site processing, and 2) robust estimates of correlation, root mean squared error, slope and intercept of a linear model used later for flagging site-variables (see section 2.2.5 and S7). The algorithm is:

(1) Compute a robust linear regression with X=Y1 and Y=Y2. Predict $Y2^{pred}$ as a function of Y1 accordingly. Calculate the outlier score $O^{Y2,Y2pred}$ following equation 1.
(2) Repeat step (1) with X and Y swapped and obtain $O^{Y1,Y1pred}$
(3) Take the maximum to obtain the final outlier score for the bivariate constraints: $O^B = \max(O^{Y1,Y1pred}, O^{Y2,Y2pred})$.
(4) Calculate an orthogonal regression between Y1 and Y2 but with outliers removed (i.e. where $O^B < 3IQR$) and report correlation, RMSE, slope and intercept.

As robust linear regression we used RANSAC (random sample consensus, (Fischler and Bolles, 1981)) implemented in scikit-learn which is based on finding consensus among many linear models fitted to different random subsets of the data. We chose a subset of random 50% of the data and a maximum iteration limit of 200.

S5 Implementation of u* uncertainty constraint for NEE

Inputs are the u* uncertainty calculated by Pastorello et al. (2020), and observed daily NEE. Outputs are the outlier score for each data point and the estimated median upper limit of tolerated u* uncertainty used for flagging site-variables (see section 2.2.5 and S7). The difference of the 84th and 16th percentiles (corresponding to 1 standard deviation for a normal distribution) of the estimated NEE distribution by Pastorello et al. was chosen as u* uncertainty metric U. The calculation of the outlier score follows as described in section 2.2.2 and S2 but with the modification that heteroscedasticity is modelled as a function of observed (instead of predicted) NEE and that only positive deviations are penalized since U is strictly positive (in contrast to residuals):

$$O_i^{u*} = \frac{U_i - P75_i}{2(P75_i - P50_i)} \quad (5)$$

The median of $P75_i$ is used for flagging site-variables as it can indicate sites where the u* uncertainty is unusual large.

S6 Attribution for flagging hard constraints

This step is relevant only if a) an outlier score from a hard constraint exceeds nIQR (e.g. an outlier in the SW_IN vs PPFD_IN constraint), and if b) none of the target variables assigned to this constraint (here SW_IN and PPFD_IN) were flagged for this data point. This can happen when all the inconsistency scores for this data point and the target variables are below 1 (e.g. $I^{SW\_IN} < 1$ and $I^{PPFD\_IN} < 1$), e.g. when all other constraints assigned to target variables give outlier scores < nIQR or when no other constraint was available for the data point. The simplest solution would be to flag both variables, while we want to avoid this to retain as much data as possible.

We use an iterative attribution scheme that aims at identifying and flagging which of the target variable is more likely to show an issue. In each iteration, we loop over the hard constraints and identify and handle only those data points that are outlier points in the current hard constraint but where none of the assigned target variables were flagged yet. Each iteration also executes steps (1) and (2) described in section 2.2.4, i.e. the propagation of flags to dependent variables and the consecutiveness constraint. Since each iteration causes flagging, the number of non-attributed outliers of hard constraints decreases with iteration.

In the first iteration, we force flagging for hard constraints that were assigned only to one variable. For example, outliers of the relationship between NEE_{NIGHT} and $RECO_{NIGHT}$_DT are attributed to RECO_DT only. The flag will be propagated to GPP_DT due to dependencies. Applying the constraint on consecutiveness will reject further data points in RECO_DT and GPP_DT. In the second iteration, we inspect if the inconsistency scores for the assigned target variables deviate by more than 0.5. If so, the target variable with largest inconsistency score is flagged. For the example on the SW_IN vs PPFD_IN constraint (Fig. S2), the second sample shows $I^{SW\_IN} = 0.8$ and $I^{PPFD\_IN}=0$ causing flagging for SW_IN. In the third iteration, we flag those variables where the inconsistency score exceeds 0.5 – this corresponds to relaxing the specified nIQR threshold to half (e.g. to 1.5 from 3). This step catches conditions where e.g. $I^{SW\_IN} = 0.7$ and $I^{PPFD\_IN}=0.3$ where no attribution was done in the previous

iteration because the difference of inconsistency scores was smaller than 0.5. In the last iteration, we flag the remaining outlier data points for all variables assigned to the hard constraint.

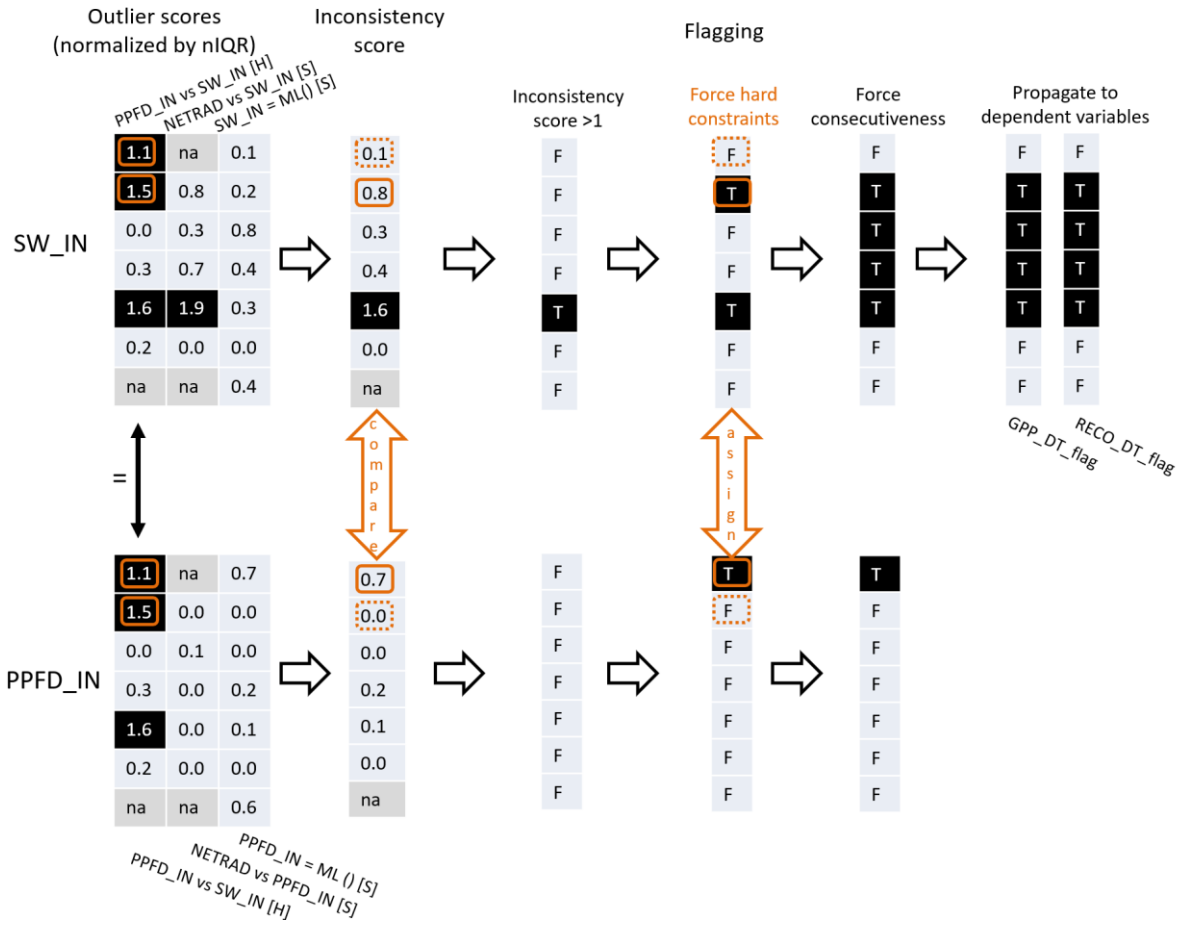

Fig. S2: Schematic flow chart for flagging data points for the example SW_IN. Data points with black background show values above the threshold nIQR or flagging (T=True, F=False). na indicates a missing value. The first two data points show outliers for the PPFD_IN vs SW_IN hard constraint, where PPFD is flagged for the first data point, while for the second data point, SW_IN is flagged in the process of forcing hard constraints (highlighted by orange colours) by considering the inconsistency scores for SW_IN and PPFD_IN. For the fifth data point, SW_IN is flagged because the inconsistency score exceeds nIQR. Forcing consecutiveness additionally rejects the third and fourth data point for SW_IN. Flagged SW_IN data points are propagated to the daytime flux partitioning variables due to dependence on SW_IN.

S7 Outlier scores for site-variables

The calculation of the outlier score for a constraint that is attributed to entire site-variables follows the principle of the box-plot rule modified for accounting of asymmetric distributions. Here we also need to distinguish constraints according to

whether outliers are in the upper or lower tail. For example, a data issue is indicated by an unusual low correlation (and not by a high one) and by an unusual high fraction of rejected values (and not by a low one).

$$O_s^{lower} = \frac{P25 - v_s}{2(P50 - P25)}$$

$$O_s^{upper} = \frac{v_s - P75}{2(P75 - P50)}$$

Where P25 and P50 are the 25th and 50th percentile of the distribution of values v and s is the index for site. Except for the correlations whose outlier score is only sensitive to the lower tail, all other metrics used for flagging site-variables are only sensitive to the upper tail.

The performance of a machine learning or bivariate constraint is measured by correlation, in a robust way since outliers were removed before calculation (see S3 and S4). Occasionally, low correlations are due to very low variance, e.g. in the absence of a seasonal cycle, which we account for by increasing the correlation value passed to the outlier score calculation under conditions of low variance (see below). This conditional upward adjustment of the correlation value is achieved by truncating at a specified minimum variance (VAR*), which is calculated from the median R^2 and the median MSE observed across sites for low variance conditions. We chose the 7th percentile of variances to identify low variance conditions. Please note that this adjustment of the correlation is only relevant for a very small percentage of sites, and that its effect is a decrease of the outlier score compared to no adjustment.

$$r_s^* = \sqrt{1 - \frac{MSE_S}{max\left(VAR_S\ , VAR^*\right)}}$$

$$VAR^* = \frac{MSE^*}{1 - median\left(r_s^2\right)}$$

Where MSE^* is the median MSE where $VAR_s < P7\left(VAR_s\right)$.

The outlier score for the regression lines of the bivariate constraints is based on slope and intercept of the linear orthogonal regression line calculated after removing outliers for robustness (see S4). Orthogonal regression minimizes errors perpendicular to the regression line, i.e. in both X and Y direction, such that slope and intercept are not sensitive to the choice for X or Y. The outlier score based on slope and intercept is based on first calculating the mean distance of a site to all other sites, and then the outlier scores sensitive to the upper tail is calculated using the distribution of distances. Because the values of slope and intercept are not comparable and have different units, the Euclidean distances are calculated in two dimensional space where the two dimensions is a pair of values that define the regression line: 1) Y at x=0 which is the intercept, and 2) Y at a typical value of $x=x_a$, i.e. for the pair (b_s, $x_a * m_s + b_s$). x_a is calculated based on the (robust) range of intercepts and the slopes for a given constraint:

$$x_a = median\left(\frac{P99(b_s) - P1(b_s)}{m_s}\right)$$

The final between-site outlier score for a bivariate constraint is then the maximum of the regression line based outlier score and the correlation based outlier score. The maximum corresponds to a logical OR operation, i.e. outliers appear if the bivariate constraint shows unusually weak performance or its regression line is unusual. Using both outlier scores individually as a

constraint instead of their maximum composite would cause problems due to violating the requirement of independence among constraints.

S8 Detection of temporal discontinuities

For a given site and target variable, we model the entire time series with two Random Forests: (1) using SW_IN, PPFD_IN, SW_IN_POT, TA, VPD, and water balance indicators (see S3) as predictors to account for variations in weather (the target variable was removed from the predictor set when relevant), and (2) using only SW_IN_POT and day of year as predictors to account for seasonality only. We use the cross-validated predictions of these models to calculate the residuals, and normalize those by the estimated IQR of the residuals varying with magnitude of the predictions (see Equation 1) respectively. The distance based test statistic (Szekely and Rizzo, 2004) given in equation (3) for the difference of two distributions X and Y (here from two temporal segments) is calculated based on the normalized residuals of the two models.

We did not calculate the significance for the test statistic, i.e. whether the difference in distributions is significant, because we found that it is not useful for our purpose: the significance test almost always returned significance while being computationally very expensive due to the necessity of recalculating the test statistic for many random permutations. To obtain a measure for the severity of a temporal discontinuity in a time series that is comparable across sites and variables we calculate the change in relative segment dispersion associated with each split. Dispersion is the sum of distances among data points (i.e. the sum of the distance matrix). Relative segment dispersion D is the sum of within segment dispersion D_l over all segments normalized by the initial dispersion D_0 among all data points (i.e. before segmentation):

$$D = \frac{\sum_{l=1}^{n_L} D_l}{D_0} \text{ with } D_l = \sum_{i=1}^{n_l} \sum_{j=1}^{n_l} \left| |X_i - X_j| \right|$$

At the beginning and before the first split, D=1. With every split z, D decreases and our break severity measure is the difference in D associated to the split compared to before the split ($\Delta D_z = D_{z-1} - D_z$). For example, consider that with the first split D decreased from 1 to 0.7, i.e. by 0.3, while for the second split it decreased from 0.7 to 0.68, i.e. only by 0.02 – the first split was obviously much more severe and relevant compared to the second.

After the break point detection was run for all sites and all target variables we estimate the outlier score (sensitive only to the upper tail) for every split based on the distribution of ΔD pooled for all sites and variables. This helps judging on how unusual a detected break point is given the context of the site network.