**Appendix S3: A slow and fast randomization method to quantify the likelihood that a signal is due to chance or not.**

*Conventional (slow) randomization*

In our approach (Fig. 3 main text) we propose randomization as a tool to determine the likelihood a signal is real. The basic idea of randomization techniques is that one can compare some model statistic determined on the observed data to the distribution of that same model statistic in many different randomized dataset. We have chosen as a model statistic the degree of model support compared to the baseline model without a weather effect (ΔAICc.) Ideally, one chooses a very large number of randomizations for this (e.g. 1,000-5,000), and the percentile of ΔAICcrand that is at least as low as the ΔAICcobs then gives a likelihood that such a low ΔAICcobs occurred by chance (PΔAICc, see Fig 4d). This method is well established in the statistical literature, but as the analysis of a single dataset can take anytime in between minutes to days, sometimes this conventional method of analysing thousands of randomized dataset may be too time consuming.

*Limited (fast) randomization*

To get a rough idea about the likelihood that a signal is true, one could perform a smaller, but still substantial number of randomizations (~100), and use the above approach to quantify a PΔAICc, which should give an unbiased, but more imprecise estimate. For situations where this is still too time consuming, we empirically derived an alternative method to determine the likelihood that a signal is real or not. This method requires only a limited number of randomizations (~5-10), making it useful for datasets and models that take hours or days to run.

To empirically derive an alternative likelihood metric, we analysed simulated data where the occurrence of a true signal was known (see section ‘*Performance of our approach and sample size considerations*’ in main text for simulation details). For each simulated dataset we categorized whether it contained a true signal (coded as 1) or not (coded as 0). We subsequently ran a complete model set to determine the best supported model, and stored several model statistics for each dataset. Specifically, we stored the sample size, the ΔAICcof the best supported model, the proportion C of models that are in the 95% model confidence set (i.e. the cumulative top 95% of Akaike weights; Box 2). It should be noted that when there is a true signal, the peak in the ΔAICc landscape is typically quite localized (see Fig. 4a and Figure of model weights in Appendix B) resulting in a low C value close to zero, while in false signals a clear peak in the ΔAICc landscape is typically lacking and many signals are equally well supported, resulting in C values closer to one.

We analyzed 2,000 datasets for each combination of sample size (N=10, 20, 30, 40, 47) and effect size (R2=0.80, R2=0.40, R2=0.20; dataset without a signal), and these dataset were analysed using either no cross-validation or cross-validation (with K=10 folds). This generated a total of 2,000\*5\*4\*2=80,000 datasets. Each of these 80,000 datasets were also randomized 5 times to quantify the median of the same above-mentioned metrics over the 5 randomisations. From this we calculated two new metrics for each dataset: ΔD= ΔAICcobserved- median(ΔAICcrandomizations) and ΔC= Cobserved- median(Crandomizations).

In the next step we divided the overall dataset in two halves. One half served as a training dataset to derive a test statisticthat can distinguish true from false signals (a discriminant function PC). The second half served as a test dataset to independently test the performance of PC and quantify the rate of misclassification (false positives and false negatives; Fig. 5a). For the aim of deriving a test statistic, it turns out that both ΔD and ΔC effectively predict the presence of a real signal, but are also highly correlated (pearson’s r>0.5 for all values of N and R2 tried), in the sense that both a high ΔD and ΔC were strongly indicative of a signal being real. However, the exact value of ΔD and ΔC that best separated true from false signals also depends on whether or not K-fold cross-validation is used, as cross-validation typically leads to much lower values of ΔAICc and C. Furthermore, a value of ΔD of say -10 units may imply a high likelihood a signal is real for a dataset with sample size of 40, but the likelihood of ΔD=-10 should be much lower for datasets with smaller sample size.

We therefore fitted the following two discriminant functions to dataset fitted with and without cross-validation:

Model 1: Logit(SignalTrue)~ β0+ β1ΔD+ β2N+ β3ΔD\*N

Model 2: Logit(SignalTrue)~ β0+ β1ΔC+ β2N+ β3ΔC\*N

In both the case with and without cross-validation model 2 was much better supported than model 1 (both ΔAIC<-2500), suggesting the ΔC metric is a better discriminant than the ΔD metric. We think the superior performance of ΔC over ΔD is due to the fact that ΔC incorporates not only information on how good the best model is compared to the baseline model, but also on how much the best model is better than most other models considered in the model set.

Based on the parameters estimated in model 2, the likelihood a signal is real (PC) is given by:

PC=1/(1+exp(-1\*(-0.540324+1.947674\*ΔC+0.078708\* N+0.313567\*ΔC\*N)

for datasets analysed without the use of cross-validation, and is given by:

PC=1/(1+exp(-1\*(-0.621031 +11.563537 \*ΔC+0.058663\*N+6.882248\*ΔC\*N)

for datasets analysed with the use of 10-fold cross-validation. The metric PC varies between 0 and 1, with values below 0.5 more likely than not to be a true signal (i.e. less likely to be due to chance).

Subsequently, we tested the rate of misclassification by PC on the independent test dataset. Specifically, we assessed the rate of false negatives in datasets that contained a real signal (Fig. 5a-i to a-iii), and the rate of false positives in datasets that contained no real signal (Fig. 5a-iii in main text). We would like to note that to assess the rate of misclassification, we had to set an arbitrary cut-off point for our statistic (PC<0.5) to decide whether we considered the climate signal to be a true signal or not, while the actual value of PC gives additional information on how certain one can be about the climate signal being true (i.e. PC=0.1 and PC=0.4 are both likely a true signal, but the likelihood is higher for *PC* =0.1). Furthermore, users may use different cut-off values depending on whether they feel false positives or negatives are most problematic for their study. For example, if one considers false positives more problematic than false negatives, one could use PC<0.25 instead of PC<0.5. Finally, we note that our PC discriminant function was trained and tested on datasets with sample sizes ranging between 10 and 47 and should thus be used with care for smaller and larger dataset. Therefore when the function ‘pvalue’ returns a PC value, in situations where the sample size falls outside this tested range *climwin* will assume a sample size of 10 or 47, depending on whether sample sizes are respectively smaller or larger than those tested. Users will be warned that the PC estimate in these scenarios will be respectively anti-conservative and conservative in these scenarios.

*How randomization is done within climwin*

The randomization procedure in function ‘randwin()’ occurs by shuffling (resampling without replacement) the ‘bdate’ column, meaning that in randomized data the weather data from a random year or site is linked to each biological record, and thus any weather signal is typically removed in the randomized data. The advantage of randomizing the ‘bdate’ column instead of for example the ‘yvar’ or ‘xvar’/’cdate’ column is twofold, first we keep the biological response variable ‘yvar’ and any confounding factors that appear in the baseline model (e.g. and effect of sex, or individualID random intercept) together. Second, we do not randomize the weather data (‘cdate’) itself within years, such that any temporal structure in the weather data is retained (autocorrelation, intra-annual trends and variability).