DETECTION OF RESIDUAL BLOCKS IN GRID-BASED DATA USING TREE SEGMENTATION

Karen Wolf [123] & Pierre Fernique [2] & Hans-Peter Piepho [3]

[1] *Limagrain Europe, Germany karen.wolf@limagrain.com*
[2] *Limagrain Europe, France pierre.fernique@limagrain.com*
[3] *University of Hohenheim, Germany hans-peter.piepho@uni-hohenheim.de*

Abstract. Applying spatial models to grid-based data relies on strong hypotheses, notably, stationarity. We propose a method to explore the hypothesis of stationarity and pinpoint parts of the grid-based data, where the hypothesis is violated.

We apply these methods to field trials in plant breeding, where grid-based data corresponds to measurements and coordinates of plant varieties in the field. The hypothesis of stationarity may be violated if spatial patterns exist in the residuals in certain areas of the field (i.e., residual blocks), due to environmental effects.

Our methods represent grid-based data in a tree graph using quad and binary tree. Residual blocks are then recovered using tree segmentation of the tree indexed data.

Keywords. Grid-based data, Spatial models, Gaussian Random Fields, Stationarity, Tree indexation, Tree segmentation, Plant breeding

1 Introduction

Plant Breeding To ensure food security despite global challenges such as climate change, steadily developing plant diseases, and increasing human population, the continuous improvement of plant varieties (i.e., cultivars) is crucial. Thereby, efficient plant breeding plays an essential role in achieving this goal. It aims at producing new cultivars with enhanced traits such as abiotic resistances (e.g., drought tolerance), biotic resistances (e.g., insects), and higher yield (Qaim, 2020).

Field Trials Plant breeders select cultivars based on their performance. This performance is measured in field trials, where the cultivar effect is computed to represent this performance. Field trials are in-vivo experiments. They are thus carried out under the influence of macro-environmental effects (e.g., rain or solar radiation affecting cultivar effect). Cultivars are therefore tested in and selected for specific macro-environments (i.e., geographic zones with homogeneous environmental conditions) where they can be cultivated by farmers.

The particularities of field trials in plant breeding are described by Mackay et al. (2019). Notably, plants cannot move. They are cultivated within plots which are usually small rectangular areas within the field, where they remain during the entire cultivation period. Therefore, in addition to macro-environmental effects, the cultivar effect of each plant is strongly affected by micro-environmental effects (e.g., variation in soil quality, shadowing of neighbouring plants). All these

undesirable micro-environmental effects are regrouped together into the generic field effect term. In the conduction of field trials, it is of high priority to estimate cultivar effects accurately and remove all micro-environmental effects from this estimation.

Experimental Designs One major problem when trying to conduct field trials to get an indication of the cultivar effect is that it may be confounded with the field effect. If this is the case, cultivar effect estimation is inaccurate and so are breeding decisions.

To overcome this problem, breeders use experimental designs. By specific spatial allocations of replications and randomisations of the cultivars, they can take into account *a priori* information relative to field conditions (e.g., soil uniformity, slopes, irrigations). For example, breeders can specify so-called blocks, which are areas in the field within which irrigation conditions are assumed to be homogeneous (Gomez and Gomez, 1984). The specifications of an experimental design can be represented in a design matrix $\boldsymbol{X}$ which encodes the cultivar effect of interest and several other effects such as various block effects to capture the field effect.

Trial Analysis During, or at the end, of the cultivation period, breeders collect data from each plot. In the case of metric traits such as yield, the data can be assumed to follow a Gaussian Random Field (GRF), denoted as follows,

$$\boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{\Sigma}),$$

where $\boldsymbol{X}$ is the design matrix, $\boldsymbol{\beta}$ is the vector of effects, and $\boldsymbol{\Sigma}$ is the variance-covariance matrix. $\boldsymbol{X}$ can be either given directly by the experimental design or altered to remove and add block effects *a posteriori*. Different types of GRFs exist, for example IID (i.e., $\boldsymbol{\Sigma} = \sigma^2\mathbb{I}$) or $AR1$ (e.g., $\boldsymbol{\Sigma} = \sigma^2 \cdot \left(\boldsymbol{\Sigma}_r \otimes \boldsymbol{\Sigma}_c\right)$, for more details see Butler et al. (2017)).

In general, the combinations of all possible design matrices $\boldsymbol{X}$ and all possible variance-covariance matrices $\boldsymbol{\Sigma}$ are used to set up a collection of models. And, among these models, the most appropriate modeling of the data can be found using model selection based on BIC, AIC, or cross-validation. Using the most appropriate model, we can restore the cultivar effect and use it for further analyses.

Residual Blocks Using the processes described hereabove, breeders assume several hypotheses, one of the strongest being stationarity. In this case, stationarity implies constant expectation (i.e., $\boldsymbol{\beta}$) and constant variance (i.e., $\boldsymbol{\Sigma}$) across the whole trial.

Stationarity can be assumed when all field conditions are known and included in the experimental design. However, not all these conditions are necessarily known and considered *a priori* in the experimental design. We here focus on the later case, that could lead to the violation of the assumption of stationarity. For example, if the field trial is very large, not all plots can be harvested on the same day. If this is not taken into account in the experimental design by including a block effect, cultivar effect estimation will be inaccurate. Furthermore, variance cannot always be assumed to be constant across the whole field trial, especially for large trials. Indeed, soil conditions may be locally homogeneous, but globally very heterogeneous, leading to large differences in terms of variance, notably for distant areas.

If the hypothesis of stationarity is violated, spatial patterns remain in the field trial. Residuals in different parts of the field follow a distinct GRF within each part. We will refer to the affected areas as *residual blocks*. Note that, if resources were unlimited, an appropriate number of replications could compensate for an imprecise experimental design. However, in practice, the possible number of replications is limited which leads to inaccurate cultivar effect estimation. Therefore, to avoid inaccurate breeding decisions, plant breeders would need to discard field trials where the hypothesis of stationarity is violated. However, if inaccuracy is induced by detectable residual blocks, plant breeders would not need to discard the whole trial. Instead, they could discard only those plots where the hypothesis of stationarity is violated.

In some cases, visualising the residuals of the estimated model in a heatmap may reveal those residual blocks. However, these manual investigations of the stationarity hypothesis are not always easy and methods able to screen a large number of trials to detect residual blocks could benefit plant breeding.

2 Detection of Residual Blocks

To detect residual blocks, we use tree indexation to get a multi-scale representation of grid-based data (i.e., matrix) and tree segmentation to find the best representative scale of this matrix.

2.1 Definition of a Tree Graph [1]

In graph theory, a graph G is defined by two sets, vertices $V \subset \mathbb{N}$ and edges E, with

$$\emptyset \subseteq E \subseteq \left\{(u, v) \in V^2 \mid u \neq v\right\}.$$

Some definitions are needed to apply the concept of tree graphs to the detection of residual blocks:

child set of vertex v, denoted as $ch(v)$, is defined as follows:
$\forall v \in V, ch(v) = \{u \in V \mid (v, u) \in E\}$.

descendant set of vertex v, denoted as $de(v)$, is defined as follows:
$\forall v \in V, de(v) = \left\{\cup_{u \in ch(v)} de(u)\right\} \cup ch(v)$.

parent set of vertex v, denoted as $pa(v)$, is defined as follows:
$\forall v \in V, pa(v) = \{u \in V \mid (u, v) \in E\}$.

leaf set of vertex v denoted as $le(v)$, or a set of vertices A, is defined as follows:
$\forall v \in V, le(v) = \left\{u \in de(v) \mid ch(u) = \emptyset\right\}$,
$\forall A \subseteq V, le(A) = \cup_{v \in A} le(v)$.

Since we use tree graphs to index grid-based spatial data, we associate a set of coordinates to each vertex in V. Therefore, instead of indices, we use coordinates to represent vertices, hence,

$$V \subset P\left(\{(i, j) \in [0\,..\,R[\times [0\,..\,C[\}\right),$$

[1] The definitions used are mostly based on Lauritzen (1996).

where R is the number of rows and C is the number of columns of the matrix and $\mathcal{P}(\cdot)$ denotes the power set. To facilitate manipulation of coordinates, let I_v (resp. J_v) be the set of rows (resp. columns) of a vertex v:

$$\forall v \in V, \qquad I_v = \{i \in [0\,..\,R[\mid \exists j \in [0\,..\,C[\wedge (i,j) \in v\}\,,$$
$$J_v = \{j \in [0\,..\,C[\mid \exists i \in [0\,..\,R[\wedge (i,j) \in v\}\,.$$

2.2 Tree indexation

Using the terms of Hunter and Steiglitz (1979), a two-dimensional array of information (i.e., matrix) can be represented by a tree graph. This can be done using tree indexation. To cover the full information of the matrix, the tree graph must have as many leaves as the matrix has elements.

The leaves are the greatest depth of the tree and the least compact representation of the matrix. All smaller depths represent the matrix more compactly, with the smallest depth, consisting of one single vertex, as its most compact representation. Thus, the tree can be regarded as a multi-scale representation of the matrix with the greatest depth being the finest scale of the matrix, and the smallest depth being the coarsest scale of the matrix.

All vertices are associated with a subset of the matrix. In our case, children are a nested split of the matrix of their parent. The connection of parents to their children (i.e., matrices and their nested split) is represented by the edges of the tree.

Applied to the example of field trials, a vertex is a block of the field. Its children are one possible division of this block and leaves are plots.

To construct a tree indexation of a matrix, we use quad and binary tree algorithms. In every step $t \in \mathbb{N}$ of the algorithm, the tree is updated. At $t = 0$, the tree T_0 consists of a single vertex v_0, which covers the information of the whole matrix:

$$T_0 = (V_0, E_0) \text{ with } V_0 = \{v_0\}, \qquad E_0 = \emptyset,$$
$$v_0 = \{(i,j) \in [0\,..\,R[\times [0\,..\,C[\,\}\,.$$

Deterministic quad tree For quad trees, vertices are either leaves or have four children. In our case, the quad tree divides a vertex into its children in a deterministic manner. Thus, the matrix is subdivided into four equal submatrices along the rows or along the columns:

$$\forall t \in \mathbb{N}, \qquad V_{t+1} = V_t \underset{v \in le(V_t)}{\cup} ch(v),$$
$$ch(v) = \left\{v_{\boxed{1}}, v_{\boxed{2}}, v_{\boxed{3}}, v_{\boxed{4}}\right\},$$

with $v_{\boxed{1}} = \left\{(i,j) \in v \mid i < \lceil m(I_v)\rceil, j < \lceil m(J_v)\rceil\right\}$, $v_{\boxed{2}} = \left\{(i,j) \in v \mid i < \lceil m(I_v)\rceil, j \geq \lceil m(J_v)\rceil\right\}$,
$v_{\boxed{3}} = \left\{(i,j) \in v \mid i \geq \lceil m(I_v)\rceil, j < \lceil m(J_v)\rceil\right\}$, $v_{\boxed{4}} = \left\{(i,j) \in v \mid i \geq \lceil m(I_v)\rceil, j \geq \lceil m(J_v)\rceil\right\}$,

where $m(\bullet) = \frac{max(\bullet)-min(\bullet)}{2} + min(\bullet)$.
Furthermore, the tree is updated by the corresponding directed edges,

$$\forall t \in \mathbb{N}, \qquad E_{t+1} = E_t \underset{v \in le(V_t)}{\cup} \left\{(v, v_{\square}), (v, v_{\square}), (v, v_{\square}), (v, v_{\square})\right\}.$$

ML binary tree For binary trees, vertices are either leaves or have two children. We use a binary tree that divides a vertex into its two children (i.e., a matrix into two submatrices, along the rows or along the columns) based on Maximum Likelihood (ML). We define

$$\begin{aligned} \forall t \in \mathbb{N}, \forall v \in le(V_t), \qquad \forall x \in I_v, \qquad & v^x_{\blacksquare} = \{(i,j) \in v \mid j < x\}, \\ & v^x_{\blacksquare} = \{(i,j) \in v \mid j \geq x\} = \overline{v^x_{\blacksquare}}, \\ \forall y \in J_v, \qquad & v^y_{\blacksquare} = \{(i,j) \in v \mid i < y\}, \\ & v^y_{\blacksquare} = \{(i,j) \in v \mid i \geq y\} = \overline{v^y_{\blacksquare}}. \end{aligned}$$

For a possible division of a vertex v into its children $\{v^x_{\blacksquare}, \overline{v^x_{\blacksquare}}\}$ (or $\{v^y_{\blacksquare}, \overline{v^y_{\blacksquare}}\}$) we can fit a separate spatial model to each of the induced submatrices, as described in Section 1 for the modeling of the data of field trials. For each possible division, we can therefore compute the log-likelihood and among all possible divisions of each leaf at step $t \in \mathbb{N}$, we select the one which maximises the log-likelihood. We will refer to the best possible division as $\{v^*, \overline{v^*}\}$ in the following.

The vertices $\{v^*, \overline{v^*}\}$ and their corresponding edges $\{(v, v^*), (v, \overline{v^*})\}$ are used to update the set of vertices V_t and edges E_t of the tree T_t:

$$\begin{aligned} \forall t \in \mathbb{N}, \qquad & V_t \underset{v \in le(V_t)}{\cup} \left\{(v^*, \overline{v^*})\right\}, \\ & E_t \underset{v \in le(V_t)}{\cup} \left\{(v, v^*), (v, \overline{v^*})\right\}. \end{aligned}$$

For both, quad tree and binary tree, tree indexation stops at t_{max}, where the tree consists of as many leaves as the matrix has elements. This implies that there is no leaf v of $T_{t_{max}}$ that still has children. $T_{t_{max}} = \left(V_{t_{max}}, E_{t_{max}}\right)$ will be referred to as $T = (V, E)$ in the following.

2.3 Tree Segmentation

As described in the previous section, a multi-scale representation of a matrix can be obtained by tree indexation, resulting in the tree $T = (V, E)$. All vertices of the tree are associated with a subset of the matrix. The set of vertices of the greatest depth (i.e., the leaves) represents the finest scale of the matrix, whereas the set of vertices of the smallest depth (i.e., v_0) represents the coarsest scale of the matrix.

If residuals can be separated into different parts, each following a distinct GRF, residual blocks correspond to a representation of the matrix in between the extremes of the finest and the coarsest scale. They can thus be represented by a subset of vertices of the tree, referred to as change-point set. Note that, vertices of the change-point set may be part of a different depth.

To estimate the change-point set (i.e., the best scale of the matrix) we use tree segmentation.

Let $\mathcal{P}(V)$ be the powerset of V. We define $\mathcal{P}'(V)$ as a subset of $P(V)$ such as

$$\mathcal{P}'(V) = \{P \in \mathcal{P}(V) \setminus \emptyset \mid le(P) = le(V)\}. \tag{1}$$

In the following, we will work with $\mathcal{P}''(V) \subset \mathcal{P}'(V)$, where additionally

$$\mathcal{P}''(V) = \left\{P \in \mathcal{P}'(V) \mid \forall (p,q) \in P^2,\ le(p) \cap le(q) = \emptyset\right\}. \tag{2}$$

The leaves of each vertex v of a possible change-point set, $P \in \mathcal{P}''(V)$, cover a fraction of the matrix. We can fit a separate spatial model to each of the induced submatrices of the division of a vertex to its children. Thus, for each possible change-point set $P \in \mathcal{P}''(V)$, we can calculate the log-likelihood.

Maximising the log-likelihood over all possible change-point sets will lead to overfitting since it results in choosing the most complex model, where each leaf of the tree represents one residual block (i.e., $P = le(T)$). Thus, we are using a score to account for the trade-off between bias (i.e., too few residual blocks) and overfitting (i.e., too many residual blocks).

Estimating the optimal change-point set within all possible change-point sets (i.e. $P \in P''(V)$), is a combinatory problem. Therefore, we use a heuristic approach, where we recover the residual blocks covered by P iteratively based on tree segmentation using two algorithms, Top-Down and Bottom-Up.

The Bottom-Up algorithm iteratively recovers residual blocks from the greatest depth of the tree (i.e., leaves) to the smallest depth (i.e., v_0). Based on a score, it decides if merges of vertices to their parents are accepted. The Top-Down algorithm, iteratively recovering residual blocks from v_0 to the leaves, decides if a split of a vertex to its children is accepted. As a score, we use BIC.

3 Results and Discussion

Material and Data To test the performance of the tree segmentation algorithms, we conduct a simulation study. We simulate field trials based on the geometry (i.e., the number of rows and columns and the coordinates for missing data) of already conducted field trials from historical Limagrain data. This ensures a realistic geometry of our simulations.

The different steps of the simulation are described in Figure 1. They rely on the simulation of blocks using Lloyd algorithm (Lloyd, 1982). Each parameter θ of a GRF is simulated using a zero-inflated model (see Eggers (2015)). The parameters are either 0, with a probability of 0.5, or drawn from different distributions:

- ρ_r, ρ_c, and β are drawn from a Beta distribution, with $\mathcal{B}(\alpha, \beta)$, with $\alpha = 3$ and $\beta = 2$.
- σ is drawn from a Gamma distribution, $\mathcal{G}(k, \theta)$, with $k = 3$ and $\theta = 1/3$.
- ϕ is drawn from an Exponential distribution, $\exp(1/\lambda)$, with $\lambda = 0.25$.

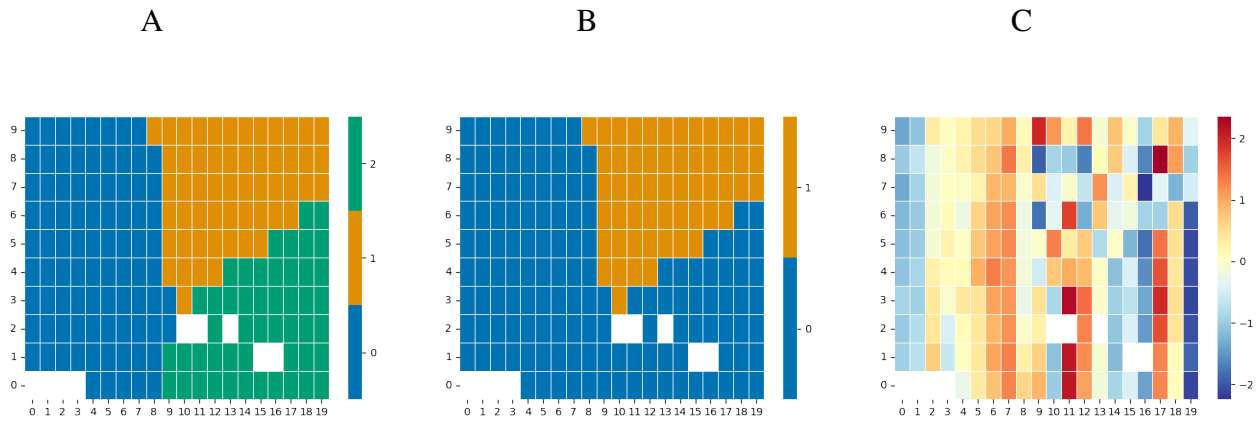

Figure 1: Illustration of the simulation of residual blocks. This simulation is based on the geometry of a field trial with 10 rows and 20 columns and missing data. Three blocks are simulated using Lloyd algorithm based on euclidean distance (A). Two simulated blocks are grouped into one resulting in two blocks (B), leading to more convex and concave shapes of the blocks than a direct simulation of two blocks. Residuals within each block follow a distinct simulated GRF (C).

Performance of tree segmentation We compare the recovered residual blocks of the tree segmentation algorithms with the true residual blocks of the simulations. To get an indication of how well recovered and true residual blocks coincide, we calculate a measure of similarity (Figure 2). The measure of similarity is based on the Hungarian algorithm (see Kuhn (1955), Denœud and Guénoche (2006)), either without clustering or by assuming perfect clustering (oracle clustering). The ML binary tree is a data-driven construction from top to bottom. We therefore expect meaningful residual blocks at small tree depth. Conversely, the quad tree is a deterministic construction. We therefore expect that segmentation is meaningful at great tree depths and that only clustering of this segmentation could lead to meaningful residual blocks.

For quad tree, Bottom-Up recovers more residual blocks than Top-Down (see Figure 2, E). The Bottom-Up algorithm tends to recover more residual blocks (i.e., vertices of P being part of greater depths of the tree) than the Top-Down algorithm since it recovers residual blocks from the greatest depth of the tree to the smallest depth. A larger number of recovered residual blocks has a higher chance of being clustered to the true residual blocks by oracle clustering. Therefore, with oracle clustering, Bottom-Up shows higher similarities than Top-Down (see Figure 2, A and B).

For binary tree, both algorithms recover a similar number of residual blocks (see Figure 2, E). No major difference in similarities can be found between the two algorithms (see Figure 2, C and D).

The Bottom-Up algorithm recovers more residual blocks for quad tree than for binary tree (see Figure 2, E). As mentioned earlier, separations of the binary tree at greater depths are less meaningful than for quad tree. Thus, Bottom-Up algorithm accepts fewer merges of vertices to their parent (equivalent to accepting more splits) for quad tree and stops at a greater depth of the tree (i.e., recovers more residual blocks).

The Top-Down algorithm recovers more residual blocks for binary tree than for quad tree (see

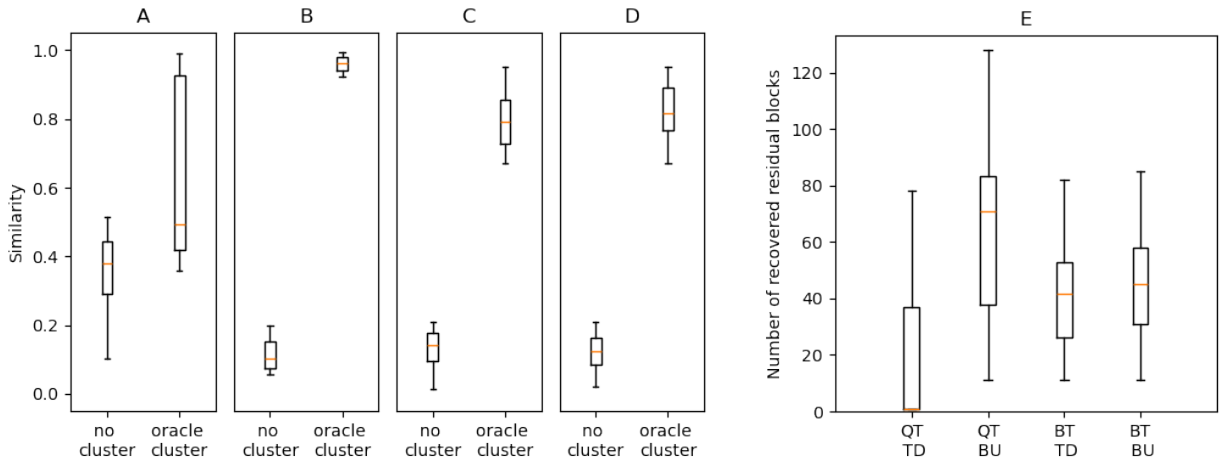

Figure 2: Boxplots without outliers for similarities between recovered and true residual blocks. A (resp. B, C, D) Similarity of quad tree with Top-Down algorithm (resp. quad tree with Bottom-Up algorithm, binary tree with Top-Down algorithm, binary tree with Bottom-Up algorithm). Similarities for recovered residual blocks without clustering are calculated using the Hungarian algorithm. Similarities for recovered residual blocks with oracle clustering are calculated using a recursive Hungarian algorithm.
(E) Number of recovered blocks of quad tree and binary tree with Top-Down and Bottom-Up algorithm.

Figure 2, E). Separations of a vertex into two (i.e., binary tree) have a higher chance of being accepted than separations into four (i.e., quad tree). Therefore, in comparison to quad tree, the Top-Down algorithm accepts splits of vertices until greater depths for binary tree (i.e., recovers more residual blocks).

For quad tree, where we aim at recovering more residual blocks, the Bottom-Up algorithm should be used. For binary tree, where we aim at recovering less residual blocks, the Top-Down algorithm is more adequate.

Constraints for tree segmentation Estimating P to recover residual blocks is a model selection problem where we are not looking for optimal modeling of only a fraction of the trial (i.e., one vertex) but of the whole trial (i.e., tree graph). For the estimation of P, we use constraint (2) to ensure that the information covered by each vertex of the change-point set, is not covered by any other vertex of the change-point set. Otherwise, the segmentation of one part of the tree would depend on other segmentations and more possible solutions for P would exist. This is the case in Fernique (2014), where the tree graph has a biological meaning since it represents a real tree. Thus, for segmentation, the whole structure of the tree graph (i.e., real tree) must be accounted and is of particular interest. Since we use the tree graph only as a tool to recover residual blocks, without biological meaning, the tree structure is of less interest and a less complex set of change-points can be considered, allowing for more computational efficiency.

Clustering of tree segmentation Constraint (2) also implies that residuals within several recovered residual blocks may follow the same GRF. As in Fernique (2014), a clustering algorithm must

account for the constraint that all elements of one recovered residual block are clustered with all elements of other recovered residual blocks. This can be done using a constrained Viterbi EM algorithm (Viterbi, 1967). Hereabove, we assume perfect clustering of recovered residual blocks to the true residual blocks (i.e., oracle clustering). Therefore, similarities are always higher with than without clustering (see Figure 2, A to D).

For quad tree, the Top-Down algorithm stops already at small depths of the tree (i.e., recovers low numbers of residual blocks) in comparison to the Bottom-Up algorithm with quad tree and in comparison to both algorithms with binary tree (see Figure 2, E). As already mentioned, this is due to the fact that separations into more parts (e.g., into four for quad tree) have a lower chance of being accepted than separations into fewer parts (e.g., into two for binary tree).

Therefore, especially for Top-Down algorithm with quad tree, synchronous segmentation-clustering might lead to higher similarities. The algorithm could test if splits of vertices into one, two, three, or four are accepted or not, leading to a higher chance of acceptance of splits in general and thus to more recovered residual blocks. Synchronous segmentation-clustering is less applicable for the Bottom-Up algorithm since it starts with a high number of residual blocks.

Scores for tree segmentation As score for tree segmentation, we use BIC. However, if tree segmentation leads to oversegmentation (i.e., recovers too many residual blocks), a subsequent clustering algorithm might not find the real residual blocks. Especially, the Bottom-Up algorithm with quad tree recovers a large number of residual blocks (see Figure 2, E). With oracle clustering it results in higher similarities than Top-Down with quad tree but without oracle clustering it results in lower similarities than Top-Down with quad tree. Thus, oversegmentation should be avoided (see Figure 2, A and B).

For segmentation, BIC was shown to tend to oversegmentation since its penalisation is too liberal (Zhang and Siegmund, 2007). Instead, other penalisations, such as slope heuristics could be used (Birgé and Massart, 2007).

Application of tree segmentation in plant breeding As Gilmour (2000) argues, '[post-blocking] based purely on statistical significance of arbitrary contrasts without a plausible explanation is not justified.' Our methods are meant to support plant breeders in screening large numbers of field trials to give them indications about areas in the trials that induce inaccuracy to their cultivar effect estimations. Based on their observations in the field during the cultivation period and data collection, plant breeders must decide if they want to discard those plots where the hypothesis of stationarity is violated.

References

Lucien Birgé and Pascal Massart. Minimal penalties for gaussian model selection. *Probability theory and related fields*, 138:33–73, 2007.

DG Butler, BR Cullis, AR Gilmour, BJ Gogel, and R Thompson. Asreml-r reference manual version 4. *VSN International Ltd, Hemel Hempstead, HP1 1ES, UK*, 2017.

Lucile Denœud and Alain Guénoche. Comparison of distance indices between partitions. In *Data Science and Classification*, pages 21–28. Springer, 2006.

Julia Eggers. On statistical methods for zero-inflated models, 2015.

Pierre Fernique. *A statistical modeling framework for analyzing tree-indexed data: Application to plant development on microscopic and macroscopic scales*. PhD thesis, Université Montpellier II-Sciences et Techniques du Languedoc, 2014.

Arthur R Gilmour. Post blocking gone too far! recovery of information and spatial analysis in field experiments. *Biometrics*, 56(3):944–945, 2000.

Kwanchai A Gomez and Arturo A Gomez. *Statistical procedures for agricultural research*. John Wiley & Sons, 1984.

Gregory M Hunter and Kenneth Steiglitz. Operations on images using quad trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):145–153, 1979.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

SL Lauritzen. Graphical models. volume 17 clarendon press, 1996.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Ian Mackay, Hans-Peter Piepho, and Antonio Augusto Franco Garcia. Statistical methods for plant breeding. In *Handbook of Statistical Genomics: Two Volume Set*, pages 501–520. Wiley Online Library, 2019.

Matin Qaim. Role of new plant breeding technologies for food security and sustainable agricultural development. *Applied Economic Perspectives and Policy*, 42(2):129–150, 2020.

Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.

Nancy R Zhang and David O Siegmund. A modified bayes information criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics*, 63(1):22–32, 2007.