Supporting Information File for

A Multiscale Spatio-Temporal Big Data Fusion Algorithm from Point to Satellite Footprint Scales

Dhruva Kathuria[1], Binayak P. Mohanty[1], and Matthias Katzfuss[2]

[1]Biological and Agricultural Engineering, Texas A&M University, College Station, Texas, USA

[2]Department of Statistics, Texas A&M University, College Station, Texas, USA.

S1. Illustration of different permutations for *Vecchia-Multiscale*

We illustrate the effect of different permutations (Figure S1 and S2) by applying the eight permutations to the hypothetical example in Figure 2 (a) comprising three datasets: areal datasets R_1 (64 green pixels) and R_2 (36 purple pixels), and point dataset P_1 (40 blue triangles), making the total number of observations $n = 140$. The numbers in columns (I) to (III) in Figure (S1) represent the ordering number in $\mathcal{A} = \{A_1, \ldots, A_{140}\}$ assigned to individual data in P_1 (I), R_1 (II) and R_2 (III) for the different permutations. Column (IV) denotes the subvector $\boldsymbol{A_{m_i}}$ (color-filled blue triangles, and color-filled green and purple pixels) for a randomly chosen pixel A_i (color-filled red) for $m = 20$.

The *Joint-Coordinate* permutation (Figure S1 (a)-(c)) sorts the data based on the sum of coordinate values resulting in the data from the three platforms getting ordered from the lower-left to the upper right along the diagonal. For any pixel A_i, this results in $\boldsymbol{A_{1:i-1}}$ located close to A_i. The subvector $\boldsymbol{A_{m_i}}$ (selected from elements of $\boldsymbol{A_{1:i-1}}$ closest to A_i in space) is thus located in the immediate neighborhood of A_i (Figure S1 (d)). *Middleout* ordering is based on the same heuristic as *Coordinate* ordering and orders the locations based on increasing distance from the mean location of the study domain (Guinness, 2018). Thus, it also has $\boldsymbol{A_{m_i}}$ located in the neighborhood of A_i (Figure S1 (h)).

The *Joint-Maxmin* ordering (Figure S1 (i)-(l)) selects the first pixel/point which is closest to the mean location of the study domain and then sequentially selects a successive pixel/point which maximizes the "minimum distance" to previously selected pixels/points (Guinness, 2018). This results in the pixels/points getting permuted such that for any A_i, $\boldsymbol{A_{1:i-1}}$ now consist of a good mix of both far and near pixels/points (Figure S1 (i)-(k)). The subvector $\boldsymbol{A_{m_i}}$ now consist of both far and near data surrounding

A_i (Figure S1 (l)). Though *Joint-Random* (Figure S1 (m)-(p)) is not based on any heuristic, it can give similar results to *Joint-Maxmin* (Guinness, 2018).

The corresponding "*Separate-*" orderings for the four "*Joint-*" orderings are given in Figure S2. The "*Separate-*" orderings separate the point and areal data, apply the permutations separately to each and then form the final permutation by sorting the permuted point data followed by the permuted areal data (Figure 4, main text). Though the "*Separate-*" orderings retain the heuristic of the corresponding "*Joint-*" permutations separately for point and areal data, the "*Separate-*" permutations introduce a constraint that the point data always lie in the beginning of the vector $\mathcal{A}$. For instance, in Figure S2 (Column I) since we have 40 point data, $\{A_1, \dots, A_{40}\}$ always represent point data in "Separate-" permutations. Now for any areal pixel A_i (which for "Separate-" permutations in this example represent $\{A_{41}, \dots, A_{140}\}$), $\boldsymbol{A_{1:i-1}}$ will always consist of point data. This often leads to the subvector $\boldsymbol{A_{m_i}}$ consist of point data which are near to A_i (Figure S2, Column IV).

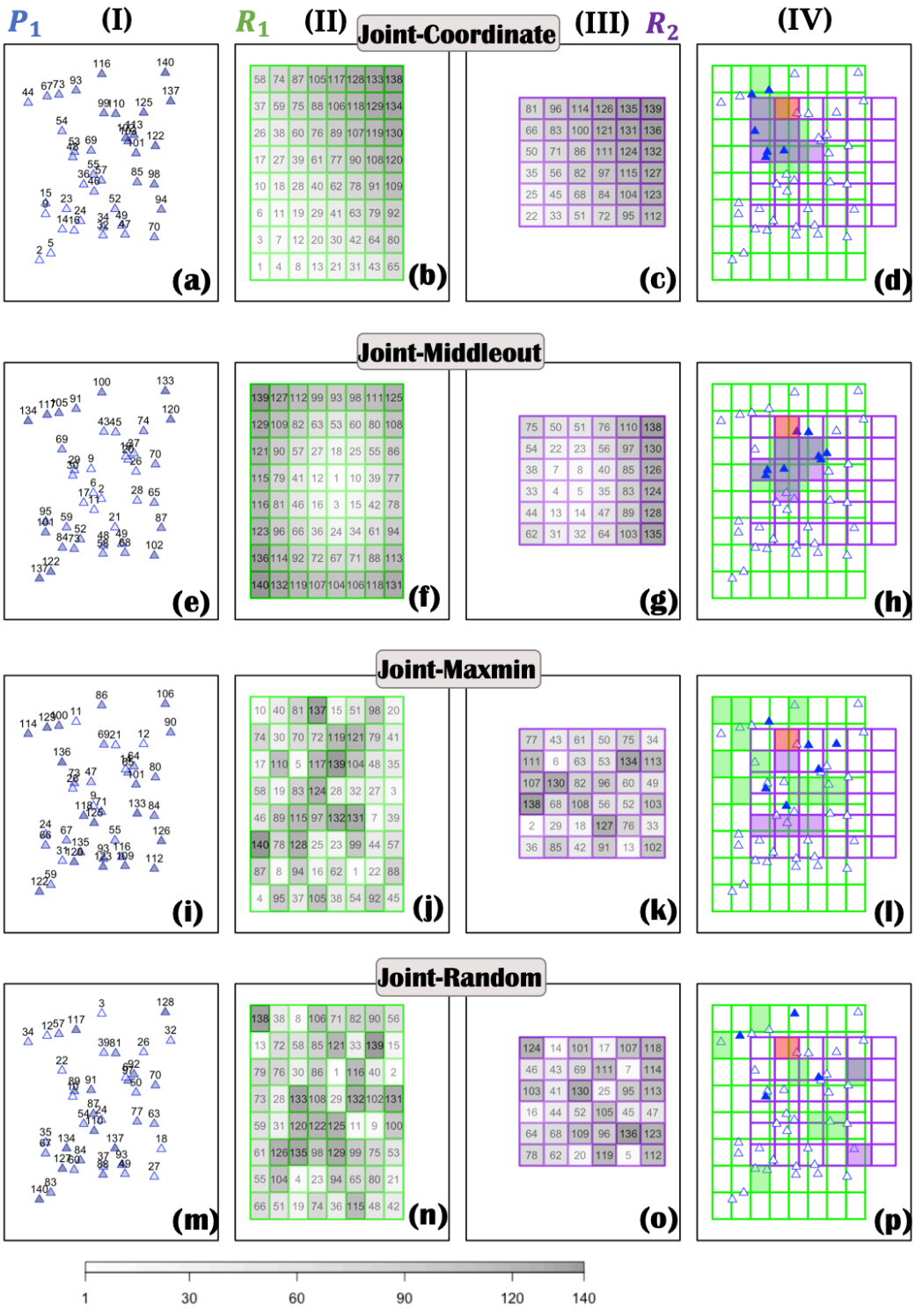

Figure S1. Illustration of the "Joint-" Permutations applied on the example from Figure 2 (a) in the main text consisting of 40 point data P_1 and 100 areal pixels in R_1 (64 pixels) and R_2 (36 pixels) . Numbers in columns (I) to (III) represent the ordering number in the vector $\mathcal{A} = \{A_1, \dots, A_{140}\}$ assigned to data in P_1 (I), R_1 (II) and R_2 (III) for the four different "Joint-" permutations. Column (d) denotes the subvector $\boldsymbol{A}_{\boldsymbol{m}_i}$ (equation 11, main text) comprising color-filled blue triangles, and color-filled green and purple pixels, for a randomly chosen pixel A_i (color-filled red) for $m = 20$.

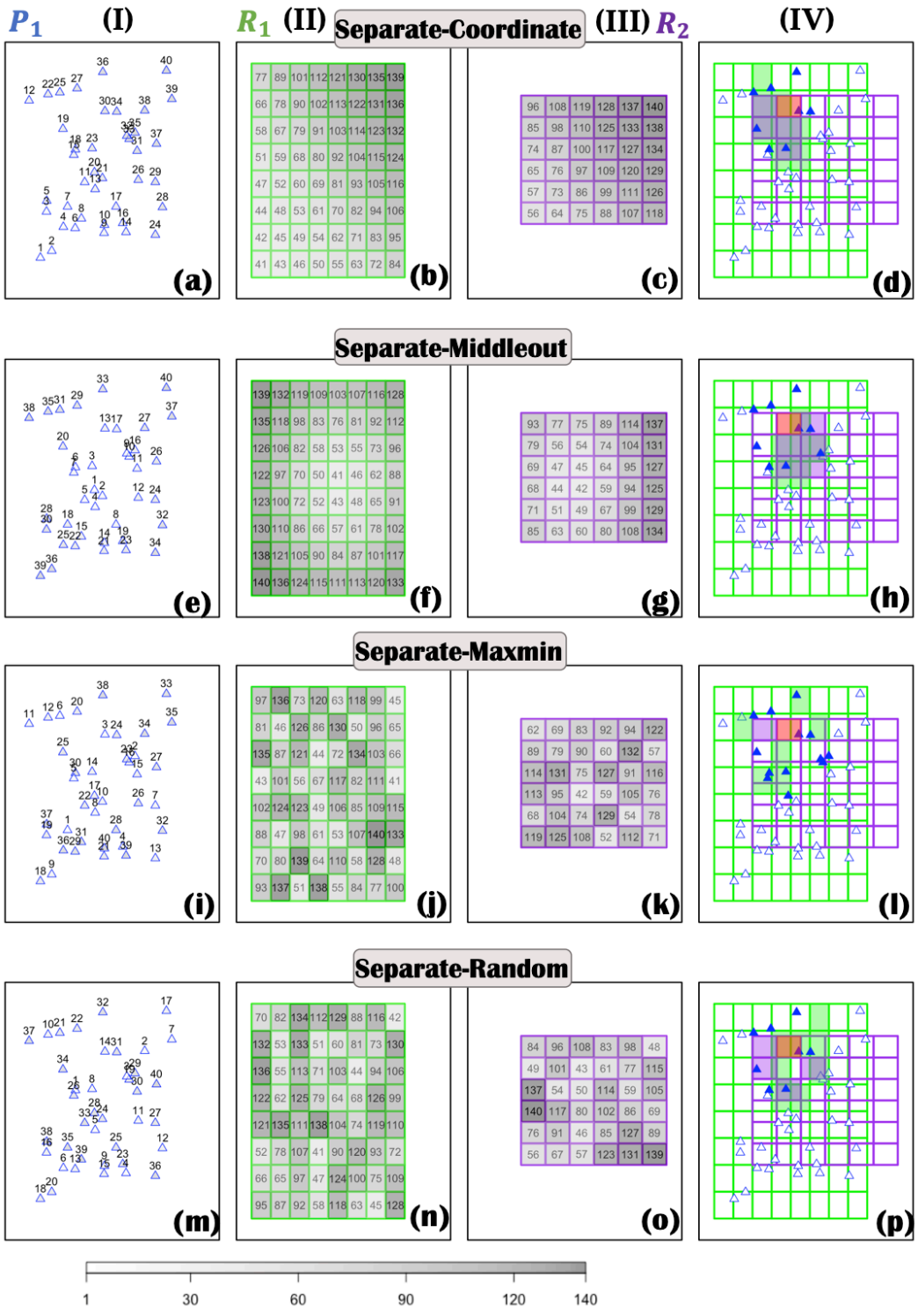

Figure S2. Illustration of the "Separate-" Permutations applied on the example from Figure 2 (a) in the main text consisting of 40 point data P_1 and 100 areal pixels in R_1 (64 pixels) and R_2 (36 pixels) . Numbers in columns (I) to (III) represent the ordering number in the vector $\mathcal{A} = \{A_1, \dots, A_{140}\}$ assigned to data in P_1 (I), R_1 (II) and R_2 (III) for the four different "Separate-" permutations. Column (d) denotes the subvector $\boldsymbol{A}_{\boldsymbol{m_i}}$ (equation 11, main text) comprising color-filled blue triangles, and color-filled green and purple pixels, for a randomly chosen pixel A_i (color-filled red) for $m = 20$.

S2. Simulation

We use simulations for two (e.g, a variable varying across latitude and longitude) and three (e.g., a variable varying across latitude, longitude and time) dimensions in space in a region $\mathcal{D} = [0,1] \times [0,1]$ and $[0,1] \times [0,1] \times [0,1]$ respectively. We fix each dimension between 0 and 1 for generality. The objective of the simulations is to investigate that for a given value of m, which approximation (equation 11) resulting out of the eight permutations better approximates the exact likelihood (equation 10). Similar to the hypothetical example in Figure 2 (a) in the main text, we assume three data sources for each setting—two aggregate datasets (R_1 and R_2) covering the entire region $\mathcal{D}$, and point dataset (P_1) in $\mathcal{D}$. The number of pixels in R_1 and R_2 along with their resolutions as well as the number of point data P_1 are given in Table S1. The number of point data are chosen as 1) 5% of the areal data to represent scenarios where the point data is sparse compared to areal data, and 2) 25% of the areal data to represent scenarios where point data are considerable in number compared to areal data. We assume an equidistant numerical grid $\mathcal{G}$ consisting of 11000 points for two dimensions and $1089 \times 11 = 11979$ points for three dimensions across $\mathcal{D}$.

As mentioned in the main text, evaluation of the exact likelihood requires quadratic complexity in the number of assumed grid points $n_{\mathcal{G}}$ and cubic complexity in the number of observations n. Therefore for the simulations, the number of observations of each platform and the size of the numerical grid are chosen so that the computation of actual likelihood $f(z(\mathcal{A})|\theta)$ is feasible.

We use a flexible class of covariance function called the Matern, with a range, smoothness and variance parameter, for simulating the covariance matrix. Other widely used covariance functions such as the Exponential and the Gaussian are special cases of the Matern. We do simulations for range $= \{0.2, 0.4, 0.6\}$, smoothness (nu) $= \{0.5, 1, 1.5\}$, variance $= 1$ and measurement error variance (in R_1 and R_2) $= \{0.05, 0.2\}$. This ensures that the simulations are carried out for a wide range of parameters resulting in a total of 72 simulations for each ordering. We perform 72 simulations for each of the eight orderings and take m = 5, 10, 20, 40, 60, 100, 120 and 180.

To control for simulation error, we use the Kullback-Leibler (KL) divergence, which measures how much information we lose using the approximation $\hat{f}(z(\mathcal{A})|\theta)$ (equation 11, main text) over the exact likelihood $f(z(\mathcal{A})|\theta)$ (equation 10, main text), both using the true value of the parameters. A lower KL-divergence between $\hat{f}(z(\mathcal{A})|\theta)$ and $f(z(\mathcal{A})|\theta)$ thus denotes a better approximation. Plots of eight representative simulations (out of 72) comparing the (log) KL-Divergence of the approximations over the true likelihood are given in Figure S3. For both 2D and 3D, in general, the *Separate-Maxmin* and *Separate-Random* perform the best while the *Coordinate-based* orderings perform the worst. There was no effect of measurement error on the relative performance of the orderings. Therefore, in general, we suggest adopting *Separate-Maxmin* or *Separate-Random* when using *Vecchia-multiscale*.

Table S1. Data setting for the simulations in Section S2.

Data	Resolution	Number of pixels/points	Grid points per pixel
	Two Dimensions		
R_1	0.09	$34 \times 34 = 1156$	9
R_2	0.06	$52 \times 52 = 2704$	4
P_1	-	$200(\approx 5\%)$ & $1000(\approx 25\%)$	-
Total	-	4060 & 4860	-
	Three Dimensions		
R_1	0.03	$11 \times 11 \times 11 = 1331$	9
R_2	0.02	$16 \times 16 \times 11 = 2816$	4
P_1		$20 \times 11 = 220(\approx 5\%)$ & $100 \times 11 = 1100(\approx 25\%)$	-
Total	-	4367 & 5247	-

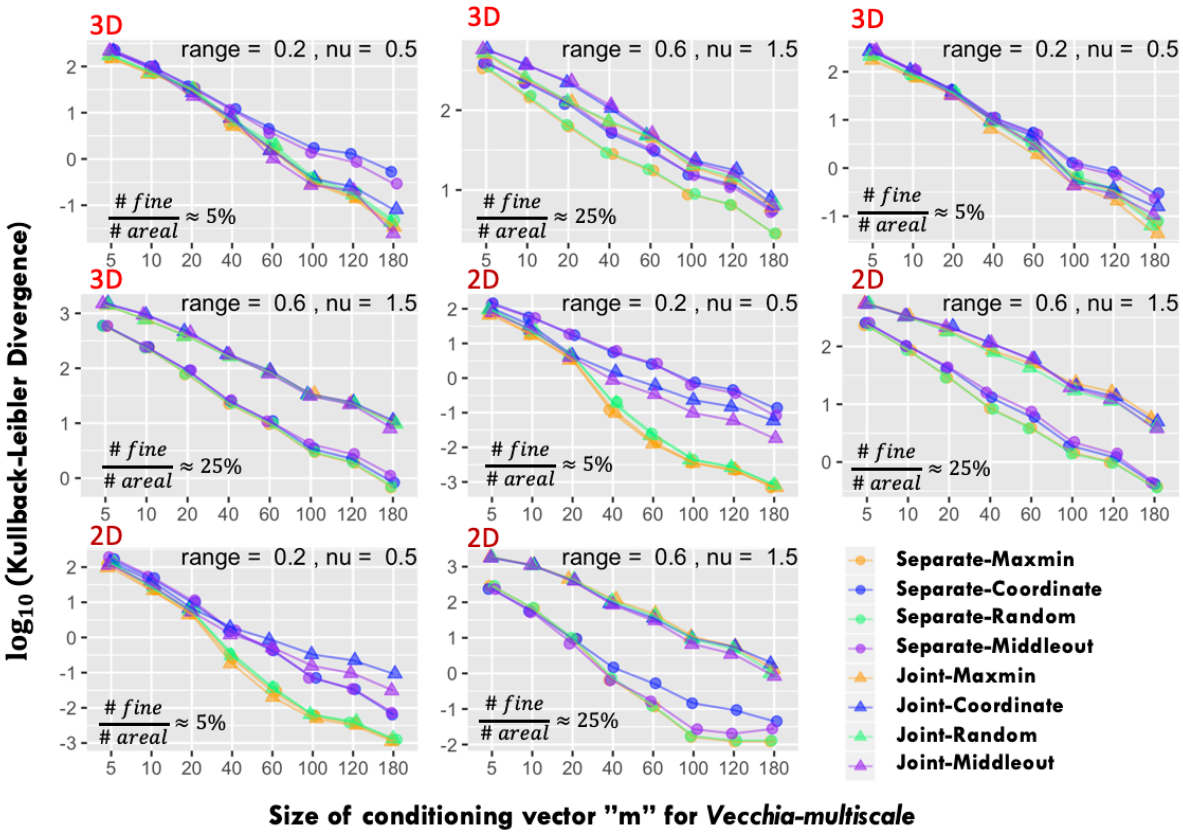

Figure S3 Representative simulations comparing the (log) KL-Divergence of the approximations over the true likelihood for measurement error variance equal to 0.05. A lower KL-Divergence denotes a better approximation. For the majority of the simulation settings, the Separate-Maxmin and the Separate-Random lead to better approximation of the exact likelihood.

S3 Supporting Information for Section 4 in the main text

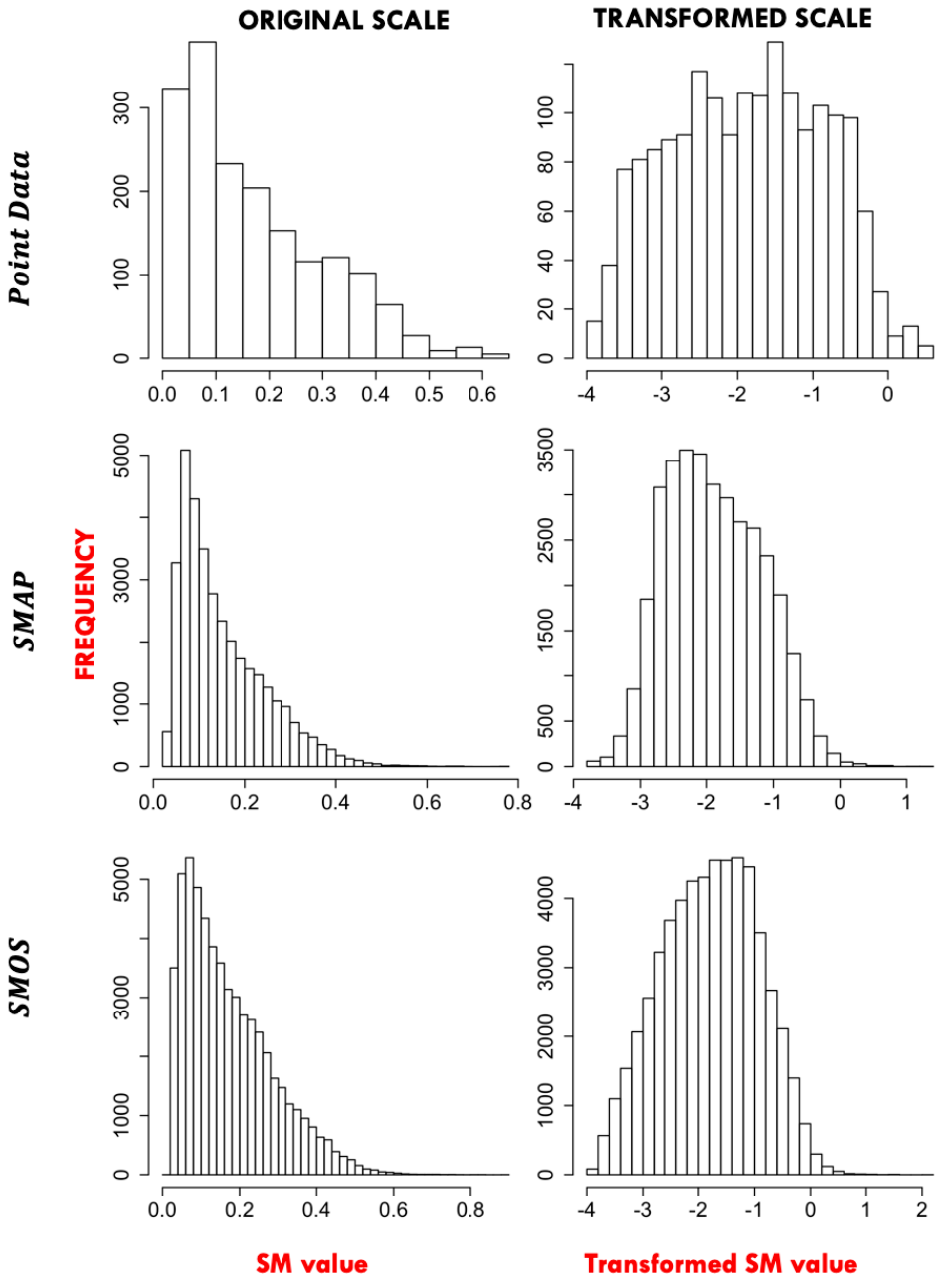

Figure S4 Histograms of point soil, SMAP and SMOS soil moisture data for July 06-20, 2017. On the original scale soil moisture exhibits considerable skewness but on the logit scale the soil moisture distribution becomes less skewed making the Gaussian assumption tenable.

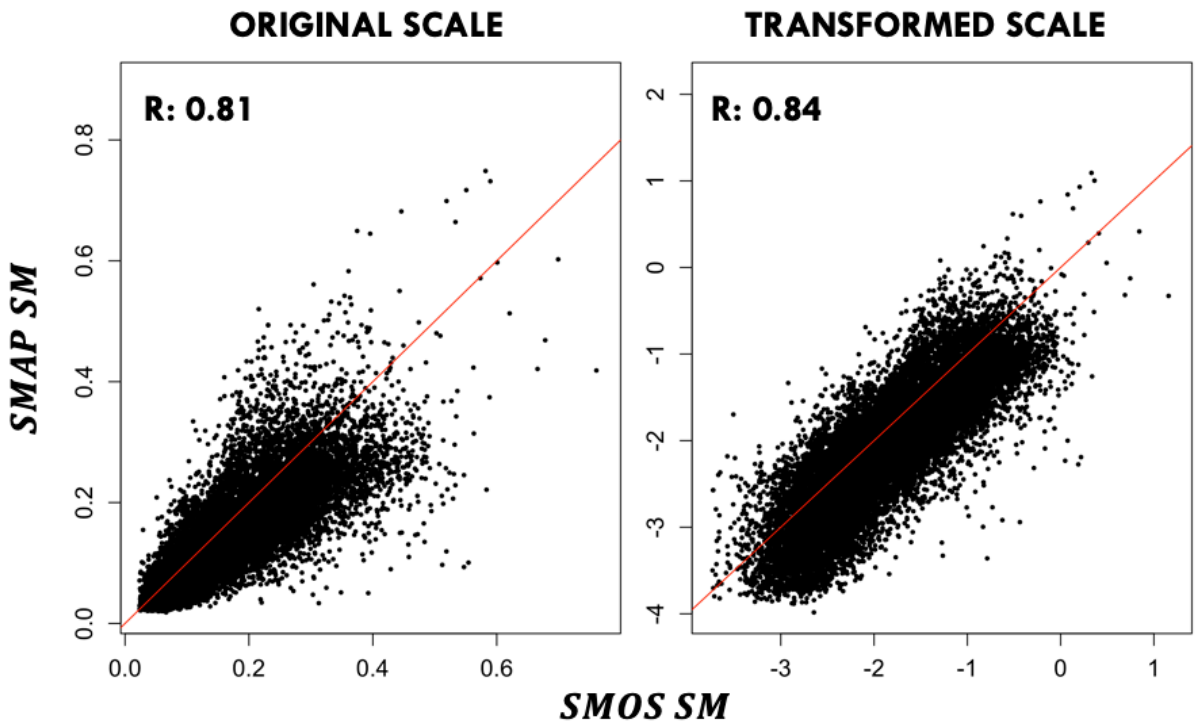

Figure S5 Overlapping SMOS and SMAP pixels for July 06-20, 2017. The SMOS pixels are bilinearly interpolated to the overlapping SMAP pixels for this exploratory analysis. The red line denotes the 1:1 line. The transformed scale results in a slightly better correlation (R) between the two datasets. On the transformed scale, it can also be seen that there is a bias between SMOS and SMAP datasets for the analyzed time period.

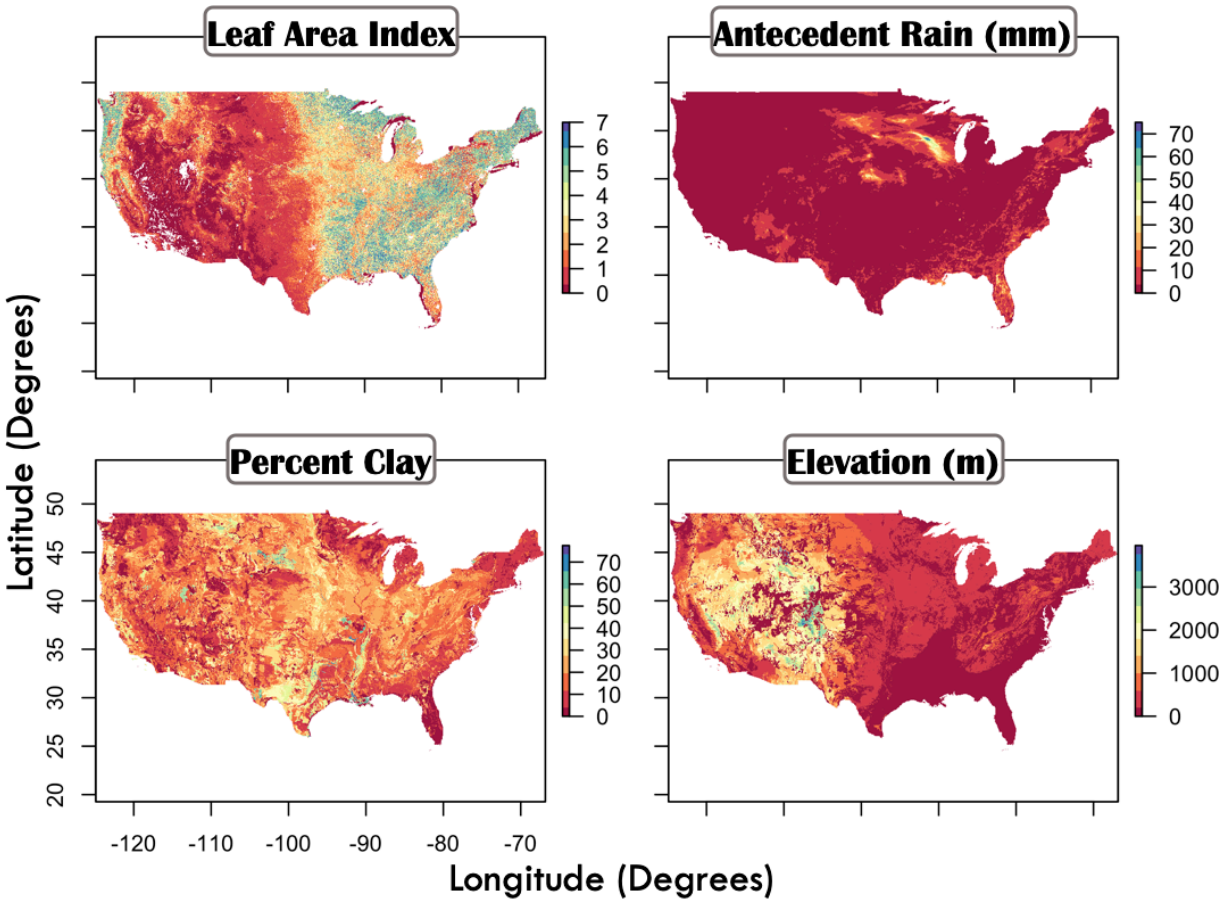

Figure S6 Covariate plots for July 06, 2020 for Contiguous US (CONUS). All the four covariates exhibit considerable heterogeneity across CONUS.