iEMSs 2008:International Congress on Environmental Modelling and Software
Integrating Sciences and Information Technology for Environmental Assessment and Decision Making
4th Biennial Meeting of iEMSs, http://www.iemss.org/iemss2008/index.php?n=Main.Proceedings
M. Sànchez-Marrè, J. Béjar, J. Comas, A. Rizzoli and G. Guariso (Eds.)
International Environmental Modelling and Software Society (iEMSs), 2008

Multivariate Spatio-Temporal Clustering (MSTC) as a Data Mining Tool for Environmental Applications

Forrest M. Hoffman [a], **William W. Hargrove**[b], **Richard T. Mills**[a], **Salil Mahajan**[c], **David J. Erickson**[a], **and Robert J. Oglesby**[d]

[a]*Oak Ridge National Laboratory (ORNL), Computer Science & Mathematics Division, Oak Ridge, Tennessee, USA (forrest@climatemodeling.org, rmills@climatemodeling.org, ericksondj@ornl.gov)*

[b]*USDA Forest Service, Southern Research Station, Asheville, North Carolina, USA (hnw@geobabble.org)*

[c]*Texas A&M University, College Station, Texas, USA (salilmahajan@neo.tamu.edu)*

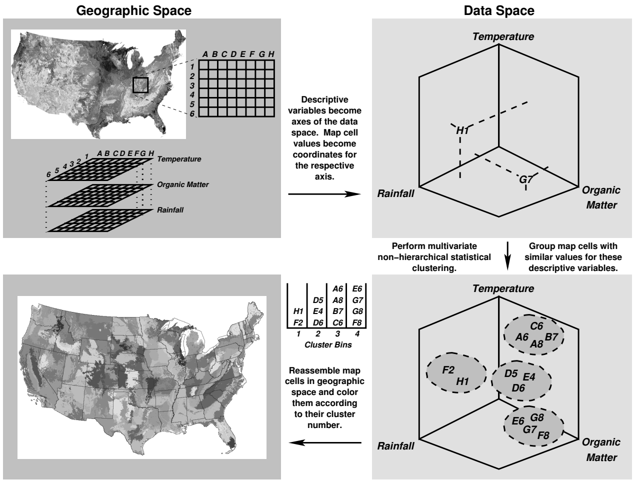
[d]*University of Nebraska, Lincoln, Nebraska, USA (roglesby2@unlnotes.unl.edu)*

Abstract: The authors have applied multivariate cluster analysis to a variety of environmental science domains, including ecological regionalization; environmental monitoring network design; analysis of satellite-, airborne-, and ground-based remote sensing, and climate model-model and model-measurement intercomparison. The clustering methodology employs a k-means statistical clustering algorithm that has been implemented in a highly scalable, parallel high performance computing (HPC) application. Because of its efficiency and use of HPC platforms, the clustering code may be applied as a data mining tool to analyze and compare very large data sets of high dimensionality, such as very long or high frequency/resolution time series measurements or model output. The method was originally applied across geographic space and called Multivariate Geographic Clustering (MGC). Now applied across space and through time, the environmental data mining method is called Multivariate Spatio-Temporal Clustering (MSTC). Described here are the clustering algorithm, recent code improvements that significantly reduce the time-to-solution, and a new parallel principal components analysis (PCA) tool that can analyze very large data sets. Finally, a sampling of the authors' applications of MGC and MSTC to problems in the environmental sciences are presented.

Keywords: cluster analysis; parallel computing; geospatial data; ecoregions; general circulation models

1 INTRODUCTION

A multivariate statistical cluster analysis technique based on an iterative k-means algorithm [Hartigan, 1975] has been applied by the authors to analyses in a wide variety of environmental science domains. The clustering methodology was first used by Hargrove and Hoffman to derive ecoregions in an objective and repeatable manner from map stacks of 9 and 25 synoptic geophysical characteristics for the conterminous United States. As shown in Figure 1, the Multivariate Geographic Clustering (MGC) procedure involves a transformation of map cells from geographic space (on the left) to points in the d-dimensional data space (on the right) formed by treating each of the factors (or maps) as an orthogonal axis. Every cell is represented by a single point, p, in data space, and points representing cells with similar characteristics will be near to each other in data space. The clustering task is to group nearby points together in an iterative fashion to create



Descriptive variables become axes of the data space. Map cell values become coordinates for the respective axis.

Perform multivariate non–hierarchical statistical clustering.

Group map cells with similar values for these descriptive variables.

Reassemble map cells in geographic space and color them according to their cluster number.

Figure 1: The Multivariate Geographic Clustering (MGC) procedure involves a transformation from geographic space (left) to data space (right), an iterative k-means cluster analysis, and a transformation back to geographic space.

the requested number of clusters, k, the centroids of which represent the mean conditions of the member cells. The cells are then projected back into geographic space and on a map are colored by their cluster assignment. Multivariate Spatio-Temporal Clustering (MSTC) employs the same approach except that multiple maps through time are combined, either as additional factors or axes in data space or as additional points in the same data space. That choice depends on whether the analysis is designed to group together points with similar histories or futures or to group points with similar conditions no matter when they occur to elucidate trajectories among clusters or states in the data or state space.

In order to cluster large ecological, climate, and remote sensing data sets, an efficient and scalable k-means algorithm that can make use of distributed memory parallel computers was created. Originally developed and implemented on a 128-node Beowulf-style parallel computer, called the Stone SouperComputer and constructed from surplus commodity desktop PCs [Hargrove et al., 2001], the high-performance parallel clustering algorithm [Hoffman and Hargrove, 1999] scales to thousands of processors on some of the largest supercomputers in the world. Described below are the clustering algorithm, recent code improvements that significantly reduce the time-to-solution, and a new parallel principal components analysis (PCA) tool that can analyze much larger data sets than any commercial package. Additionally, a sampling of the authors' applications of MGC and MSTC to problems in the environmental sciences are presented.

2 Cluster Analysis Algorithm

2.1 Overview

The clustering method consists of two parts: initial centroid determination, called seed finding, and iterative assignment of points to centroids until convergence is reached. Initial centroids are ordinarily established by sequentially examining each point in the data set and retaining a list of the k most widely separated points in data space, which become the seeds. This inherently serial process is difficult to parallelize; however, the data set can be divided equally among the m compute processes, each of which finds the best k candidate centroids, then a single node can find the "best of the best" k centroids from the $k \times m$ candidates. This parallel seed-finding scheme produces seeds that are not as widely dispersed as those produced by the serial scheme, but because the iterative portion of the clustering method runs quickly in parallel, starting with lower quality seeds that may require a few additional iterations does not significantly increase the time to solution. When the number of processes, m, is large, the cost (in time) for computing the "best of the best" seeds on a single process can be large. As a result, an alternative scheme was implemented in which the upper half of the active processes sends its k candidate seeds to the lower half of the active processes, the upper half become inactive, and the lower half find the best k candidate seeds from the $2k$ each now possesses. This "folding" procedure repeats until only a single process is left, and its best k candidates become the seeds.

In the iterative portion of the method, each point is assigned to the cluster centroid to which it is closest, by simple Euclidean distance, in data space. After all the points are assigned to a cluster, new positions are calculated for each centroid as the mean value along every axis of the points assigned to that centroid. This procedure of assigning points to centroids and recomputing the centroid locations repeats until the number of points that change cluster assignment drops below a convergence threshold. Once this threshold is met, the final cluster assignments and centroid locations are saved. This algorithm is implemented with a traditional master/slave parallel architecture using the Message Passing Interface (MPI). The algorithm is nearly perfectly parallelizable and produces the same result whether run in serial or in parallel. In each iteration, the master process distributes the current centroids to the slave processes, assigns blocks of points to slaves for classification (*i.e.*, assignment to nearest centroid), collects those classifications, and recomputes new centroid locations based on cluster membership. This procedure repeats until convergence is attained. The block size is specified by the user and can be set to optimize code performance on various parallel systems with different numbers of processors, amounts of memory, and I/O characteristics.

2.2 Cluster Algorithm Improvements

The need to analyze and compare increasingly large model and observational data sets has motivated a number of recent software engineering efforts aimed at improving the performance of the clustering code. These include 1) reorganization of the parallel clustering code to improve readability and simplify debugging, 2) implementation of an acceleration technique that decreases the number of comparisons that must be performed as progress toward convergence is made, and 3) implementation of a method for intelligently handling centroids that lose all cluster membership in an iteration.

Acceleration Technique. Two modifications to the standard k-means algorithm, described by Phillips [2002a, b], can significantly reduce the time-to-solution without changing the clustering results. The first eliminates unnecessary point-to-centroid distance computations and comparisons based on the previous cluster assignment and the new inter-centroid distances. Ordinarily, each point p is compared against each centroid c_i resulting in $O(nk)$ comparisons and a running time of $O(ndk)$ for each iteration. Phillips [2002a] showed, using the triangle inequality, that if the distance between the former centroid assignment c_i and the next candidate centroid c_j is greater than or equal to twice the distance between the point p and the former centroid c_i, evaluating the distance between the point p and the candidate centroid c_j is unnecessary. The triangle inequality states that $d(c_i, c_j) \leq d(p, c_i) + d(p, c_j)$, where the function $d(a, b)$ gives the Euclidean distance between points a and b. As a result, $d(p, c_j) \geq d(c_i, c_j) - d(p, c_i)$. Therefore, if it is known that $d(c_i, c_j) \geq 2d(p, c_i)$, one can conclude that $d(p, c_j) \geq d(p, c_i)$ without having to calculate $d(p, c_j)$. The candidate centroid, c_j, is eliminated without ever computing its distance to the point p because the former centroid, c_i, is equally close or closer. The second modification of Phillips [2002a] further reduces evaluations by sorting inter-centroid distances so that new candidate centroids c_j are evaluated in order of their distance from the former centroid c_i. Once the critical distance $2d(p, c_i)$ is surpassed, no additional evaluations need to be performed; the nearest centroid is known from a previous evaluation. When the number of points n is much larger than the number of clusters k, the cost of sorting inter-centroid distances is negligible.

In order to implement this acceleration technique in the parallel clustering program, the code was modified so that the master process retains the former cluster assignments into the next iteration and distributes them to the slave processes when they receive a new block of points to assign to cluster centroids. In addition, a parallel sorting scheme was implemented in which, at the end of each iteration and after the new centroid positions have been computed and distributed, the slave processes divide up the total number of centroids, k, and perform the necessary sorting of inter-centroid distances for their contiguous block of centroids. The k resulting ordered distance vectors are collected and distributed to all slave processes for use in the next iteration.

"Warping" Empty Clusters. Initial cluster centroids are chosen to be a subset of the points to avoid starting off the clustering process with centroids that have no members in the first iteration. This often prevents having centroids without any points assigned to them in all subsequent iterations, but experience has shown that centroids occasionally lose all cluster members in later iterations. We call such orphaned centroids empty clusters. A scheme has been developed to handle empty clusters by moving or "warping" them to the location in data space of the points that are the worst fit to their centroids.

This scheme is implemented by having all slave processes keep track of the farthest (or worst fitting) point for each cluster its subset of points, then these are passed to the master process. The master process keeps only the farthest point for each cluster it receives from all slave processes, called the "worst of the worst." At the end of an iteration, if any empty clusters are detected, the master process will sort the list of centroids by the distance to its farthest point. This list is read from the bottom since an ascending sort is performed, and each empty cluster is "warped" to the location of the point that has the next worst fit, called the "farthest of the far." When this occurs, the point is reassigned to the newly warped centroid and removed from the cluster to which it was assigned in the current iteration. Because some clusters may have only one member point, those points are not candidates for reassigning to empty clusters. In practice such single-member clusters will sort to the top of the list and are unlikely to be considered since the "farthest distance" will be zero for those clusters.

Theoretically, a maximum of $k/2$ empty clusters could be handled in this fashion. In practice, however, the presence of single-member clusters further reduces this number. If the list of candidate farthest points is exhausted, the remaining empty clusters are warped to the origin, $(0, 0, \ldots)$. In any iteration in which empty clusters are detected, the master process sets a "warp flag" that is broadcast to slave processes. This warp flag ensures that convergence is not reached in any iteration containing empty clusters, *i.e.*, at least one more iteration will be performed.

2.3 Parallel PCA Tool

Many data sets requiring analysis via MSTC are of high dimension: those incorporating hyperspectral imagery, for example, may involve on the order of hundreds of descriptive variables. Much of this information is redundant, and it is often possible to effectively describe the variability in the data using a significantly lower-dimensional representation. Working in a space of relatively low-dimension allows the data set to be clustered in less time than would be required using the original, high-dimensional representation, or, alternatively, allows a much larger data set to be clustered in the same amount of time. Additionally, clustering in the reduced space can sometimes lead to higher quality clusters due to noise suppression and other properties of the reduction.

One of the most common methods for unsupervised data reduction is principal components analysis (PCA), originally described by Pearson [1901] and Hotelling [1933], independently. PCA seeks an orthogonal linear transformation that projects the original data into a new coordinate system in which the first dimension (or *principal component*) captures the largest possible amount of variance, the second dimension is orthogonal to the first while capturing the largest possible amount of remaining variance, and so on. A reduced dimension representation that captures an arbitrary amount of the total variance of the original data set can be constructed by projecting the original data onto the first d' principal components that account for this total variance.

A great number of statistical software packages can perform PCA computations. Available packages are all designed for sequential computers, however, and are incapable of handling the very large data sets we use with the MSTC tool, due to both wall-clock time constraints and memory constraints. We have recently developed a distributed memory, parallel tool for PCA computations. It has been built using the Parallel Linear Algebra Package (PLAPACK) [van de Geijn, 1997], an object-oriented framework for expressing parallel dense numerical linear algebra computations at a high level of abstraction. To calculate the PCA, we do not explicitly form the covariance matrix, opting instead for the more numerically robust approach of computing the singular value decomposition (SVD) of the scaled, standardized data matrix. We use the SVD routine provided by PLAPACK, which performs parallel matrix-vector operations to reduce the matrix to bidiagonal form. A sequential bottleneck exists because the singular values of the bidiagonal matrix are then computed sequentially; this portion of the calculation is small, however, and happens quickly. After the PCA is computed, the matrix-matrix multiplication required to project the original data onto the principal components space is performed in parallel. Our parallel tool has enabled us to calculate the PCA on far larger data sets than has previously been possible: We recently computed the PCA of a data set consisting of 47 841 280 points of 131 variables each (a roughly 48 GB file when stored as single precision floating-point values) in only 1 533 seconds using 40 compute nodes (using one compute core per node) on Jaguar, the Cray XT4 system at Oak Ridge National Laboratory.

3 Environmental Applications

Initial applications of the cluster analysis methodology to environmental problems were described by Hargrove and Hoffman [2004b]. A sampling of these, along with more recent follow-on analyses, are presented below.

3.1 Delineation of Ecoregions

Ecoregions are useful to ecologists as simplified generalizations of complex combinations of climatic, edaphic, and topographic conditions that may be associated with the assemblages of plant and animal species for which those conditions permit growth and reproduction. Historically, ecoregions were delineated based on human expertise, and such subjective regionalizations are not reproducible by others since the methods and data employed are not available. An intermediate step toward repeatability was taken by Holdridge [1947] and Köppen and Geiger [1928, 1930-1939], who explicitly formulated and codified rules for discriminating one region or zone from another. However, these rules were simple thresholds applied sequentially to a small number of characteristics and did not consider interactions among characteristics. Two of us (Hargrove and Hoffman) began experimenting with clustering methods for objectively and repeatably generating quantitative ecoregions, using many co-registered data layers from a Geographic Information System.

Qualitative ecoregions were originally developed for generic purposes. The same ecoregions used for predicting the spread of an invasive plant would also be used to stratify sampling for a threatened animal species. Even Holdridge- and Köppen-type ecoregions are based on environmental factors deemed to be broadly important, like temperature and precipitation. This "one size fits all" philosophy decreases the utility of the ecoregion concept. Quantitative ecoregions, on the other hand, can be delineated from a custom-created combination of environmental characteristics believed to be relevant to the species or phenomena under investigation. For example, Figure 2 shows a set of 90 ecoregions in the continental United States developed from 30 factors chosen to stratify sampling from eddy-covariance flux towers. These ecoregions indicate areas within which the characteristics of CO_2 flux are expected to be about the same. Factors used include diurnal temperature differences, heat and cold degree day accumulations, precipitation sums, and numbers of wet days, all independently calculated for the growing and non-growing seasons. Also included are soil depth, nitrogen, and organic content of soil, a topographic relative wetness index, and remote sensing-derived values: Enhanced Vegetation Index (EVI), Fraction of Photosynthetically Active Radiation (FPAR), Gross Primary Productivity (GPP), and a Respiration Index (RI) from MODIS (the Moderate Resolution Imaging Spectroradiometer) [Hargrove and Hoffman, 2004a]. A generalized set of ecoregions is unlikely to be useful for such specific applications.

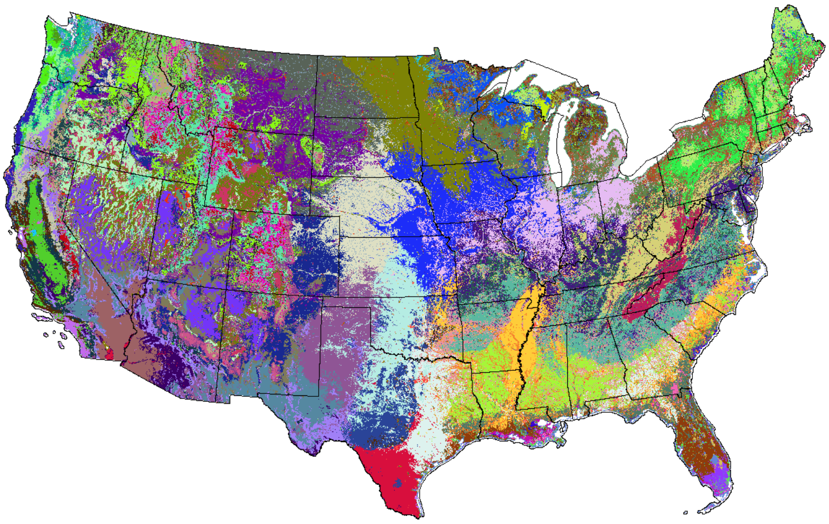


Figure 2: This map shows the 90 most-different ecoregions in the continental United States, colored randomly, custom developed for significance with respect to eddy-covariance flux tower sites.

3.2 Representativeness of Sampling Networks

The Euclidean distance from any region to the nearest training region indicates how well the mixture of conditions within that ecoregion represents the combination of conditions within the selected training area. The inverse of the same concept can be used to gage how well a network of sample locations represents the conditions found within a larger map that contains it. A network in this case can be a constellation of monitoring stations or locations where a set of samples are taken. By calculating the Euclidean distance from each ecoregion and the most similar of the sampling locations, we can map how well or poorly each ecoregion in the map is represented by the most similar station in the network. A map in which similar locations are colored white and dissimilar locations are colored black shows areas in the map that are well-represented and poorly represented (respectively) by this network. The sum of all Euclidean distances from each ecoregion to the most-similar network station inversely quantifies how well the network represents this area. Figure 3 shows the global representativeness of the international FluxNet network of eddy-covariance flux towers with respect to 14 characteristics that include climate and soil properties. Some incidental representativeness can be seen on continents not having any eddy-flux towers; however, some areas, like India and the Amazon, are very poorly represented by even the most similar station in the FluxNet network [Sundareshvar et al., 2007]. Such analyses can be used to design optimal sampling networks for any area before deployment [Schimel et al., 2007].

3.3 Model Diagnostics and Intercomparison

The MSTC technique has proved useful in evaluating the large time series output of global climate models. Such evaluations may be used to help diagnose model behavior, provide a basis for comparison of multiple ensemble simulations from one or more models, or extract patterns from model projections. MSTC was applied to the monthly time series output of temperature, precipitation, and soil moisture from five 99-year Business-As-Usual (BAU) transient simulations for years 2000–2098 from the Parallel Climate Model (PCM) [Hoffman et al., 2005]. In this case, the multivariate clusters form climate regimes in a climate phase space that vary seasonally and inter-annually. By treating these regimes as discrete climate states, the dynamic behavior of the climate system can be described as sequences of climate state occupancy. Cluster frequency plots showed that, over the course of these simulations, the conditions typical of Antarctic Winter (the coldest and driest climate state) shrank in land area while the conditions typical of desert regions (the hottest and driest climate state) grew significantly in land area.

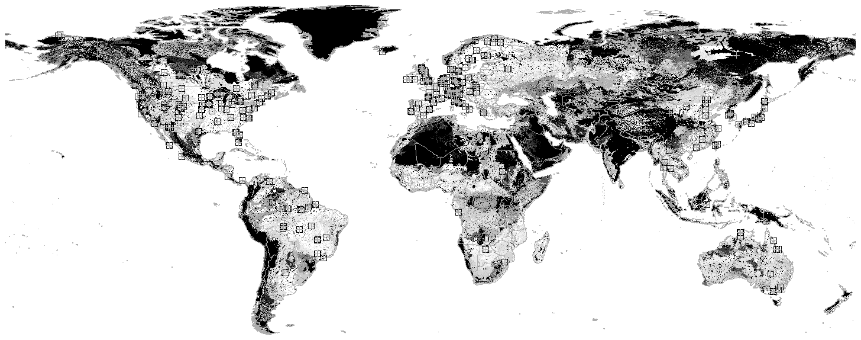


Figure 3: This map shows the global representativeness of the FluxNet network of eddy-covariance flux tower sites (boxes) in levels of gray. Dark areas are poorly represented or sampled by the existing network while light and white areas are well represented.

3.4 Model-Observation Comparison

The same cluster analysis approach has been used to compare high frequency atmospheric observations with global climate model results [Mahajan et al., 2007]. The observational data—consisting of temperature and moisture at 48 levels in the atmosphere, wind speed at 62 levels, and surface pressure—were collected by the U.S. Department of Energy's Atmospheric Radiation Measurement (ARM) program at its Southern Great Plains (SGP) site centered near Lamont, Oklahoma, US. The model results were obtained from an ensemble member of a Community Climate System Model (CCSM) general circulation model (GCM) run under the IPCC SRES A2 scenario that produced 6-hourly output. A three-way process was employed to compare ARM observations with GCM output, where 1) CCSM output was projected onto states derived from ARM observations, 2) ARM observations were projected onto states derived from CCSM output, and 3) both ARM observations and CCSM output were projected onto states derived from the combination of the two data sets. Comparisons of twelve atmospheric states derived from the combination of ARM observations and CCSM output indicate that distinct singular states exist in each data set. That is, the model exhibited an atmospheric state that never appeared in the observations, while the observations contained atmospheric states never captured by the model. Comparing the populations of state frequencies in this manner points out model deficiencies and provides quantitative information to model developers about how models can be improved.

4 Conclusions

The MGC and MSTC methodologies are powerful techniques for detecting patterns in large data sets and generating useful products for ecological and climate analyses. Such data mining techniques are often the only tools that can be used to analyze very large data sets, like climate model results, without "hiding" patterns of potential interest within other statistics. The improved parallel implementation of the accelerated k-means clustering method with empty cluster "warping," combined with the PCA tool, can now be applied to data sets of unprecedented size. Future work will focus on applying these tools to change detection in satellite remote sensing data as an early-warning system for detecting forest threats.

Acknowledgments

This research was partially sponsored by the Climate Change Research Division (CCRD) of the Office of Biological and Environmental Research (OBER) within the U.S. Department of Energy's Office of Science (SC). This research was partially sponsored by the U.S. Department of Agriculture (USDA) Forest Service. This research used resources of the National Center for Computational Science (NCCS) at Oak Ridge National Laboratory (ORNL), which is managed by UT-Battelle, LLC, for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

REFERENCES

Hargrove, W. W. and F. M. Hoffman. A Flux Atlas for Representativeness and Statistical Extrapolation of the AmeriFlux Network. Technical Memorandum ORNL/TM-2004/112, Oak Ridge National Laboratory, April 2004a.

Hargrove, W. W. and F. M. Hoffman. Potential of multivariate quantitative methods for delineation and visualization of ecoregions. *Environmental Management*, 34(5):s39–s60, 2004b. doi:10.1007/s00267-003-1084-0.

Hargrove, W. W., F. M. Hoffman, and T. Sterling. The Do-It-Yourself Supercomputer. *Scientific American*, 265(2):72–79, August 2001.

Hartigan, J. A. *Clustering Algorithms*. John Wiley & Sons, New York, 1975.

Hoffman, F. M. and W. W. Hargrove. Multivariate geographic clustering using a Beowulf-style parallel computer. In Arabnia, H. R., editor, *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '99)*, volume III, pages 1292–1298, Las Vegas, Nevada, June 1999. CSREA Press.

Hoffman, F. M., W. W. Hargrove, D. J. Erickson, and R. J. Oglesby. Using clustered climate regimes to analyze and compare predictions from fully coupled general circulation models. *Earth Interactions*, 9(10):1–27, August 2005. doi:10.1175/EI110.1.

Holdridge, L. R. Determination of world plant fomrations from simple climatic data. *Science*, 105:367–368, April 1947. doi:10.1126/science.105.2727.367.

Hotelling, H. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

Köppen, W. P. and R. Geiger. Klimakarte der Erde. *Gotha*, 1928.

Köppen, W. P. and R. Geiger. *Handbuch der Klimatologie*, volume 1–5. Berlin, 1930-1939.

Mahajan, S., F. M. Hoffman, W. W. Hargrove, S. W. Christensen, and R. T. Mills. A cluster analysis approach to comparing Atmospheric Radiation Measurement (ARM) observations with general circulation model (GCM) results. *Eos Trans. AGU*, 88(52), December 2007. Fall Meet. Suppl., Abstract A41A-0010.

Pearson, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

Phillips, S. J. Acceleration of k-means and related clustering algorithms. In Mount, D. M. and Stein, C., editors, *ALENEX '02: Revised Papers from the 4th International Workshop on Algorithm Engineering and Experiments*, pages 166–177, London, UK, 2002a. Springer-Verlag.

Phillips, S. J. Reducing the computation time of isodata and k-means unsupervised classification algorithms. In *Geoscience and Remote Sensing Symposium, 2002 (IGARSS'02)*, volume 3, pages 1627–1629, June 2002b. doi:10.1109/IGARSS.2002.1026202.

Schimel, D., W. Hargrove, F. Hoffman, and J. McMahon. NEON: A hierarchically designed national ecological network. *Frontiers in Ecology and the Environment*, 5(2):59, March 2007.

Sundareshvar, P. V., R. Murtugudde, G. Srinivasan, S. Singh, K. J. Ramesh, R. Ramesh, S. B. Verma, D. Agarwal, D. Baldocchi, C. K. Baru, K. K. Baruah, G. R. Chowdhury, V. K. Dadhwal, C. B. S. Dutt, J. Fuentes, P. K. Gupta, W. W. Hargrove, M. Howard, C. S. Jha, S. Lal, W. K. Michener, A. P. Mitra, J. T. Morris, R. R. Myneni, M. Naja, R. Nemani, R. Purvaja, S. Raha, S. K. S. Vanan, M. Sharma, A. Subramaniam, R. Sukumar, R. R. Twilley, and P. R. Zimmerman. Environmental monitoring network for India. *Science*, 316:204–205, April 2007.

van de Geijn, R. A. *Using PLAPACK: Parallel Linear Algebra Package*. MIT Press, Cambridge, MA, 1997.