# pKluster: A Tool for Scalable $k$-means Analysis of Geospatiotemporal Data Sets

Richard Tran Mills$^{\alpha}$, Forrest M. Hoffman$^{\beta}$, Jitendra Kumar$^{\beta}$, Sarat Sreepathi$^{\beta}$, Vamsi Sripathi$^{\gamma}$, William W. Hargrove$^{\delta}$

$^{\alpha}$Argonne National Laboratory, $^{\beta}$Oak Ridge National Laboratory, $^{\gamma}$Intel Corporation, $^{\delta}$USDA Forest Service Southern Research Station

## 1. Introduction

- The increasing availability of high-resolution geospatiotemporal data sets from sources such as observatory networks, remote sensing platforms, and computational Earth system models has opened new possibilities for knowledge discovery and mining of ecological data sets fused from disparate sources.
- Traditional algorithms and computing platforms are impractical for the analysis and synthesis of data sets of this size; however, new algorithmic approaches that can effectively utilize the complex memory hierarchies and the extremely high levels of available parallelism in state-of-the-art high-performance computing platforms can enable such analysis.
- We describe pKluster, an open-source tool we have developed for accelerated $k$-means clustering of geospatiotemporal data. pKluster supports distributed-memory parallelism and can effectively utilize state-of-the art multi- and manycore processors, such as the second-generation Intel Xeon Phi ("Knights Landing") processor, as well as GPGPUs.
- We examine some practical applications of pKluster to the climate, remotely-sensed vegetation phenology, and LiDAR data sets and speculate on some of the other applications that such scalable analysis methods may enable.

## 2. Scalable $k$-means Clustering with pKluster

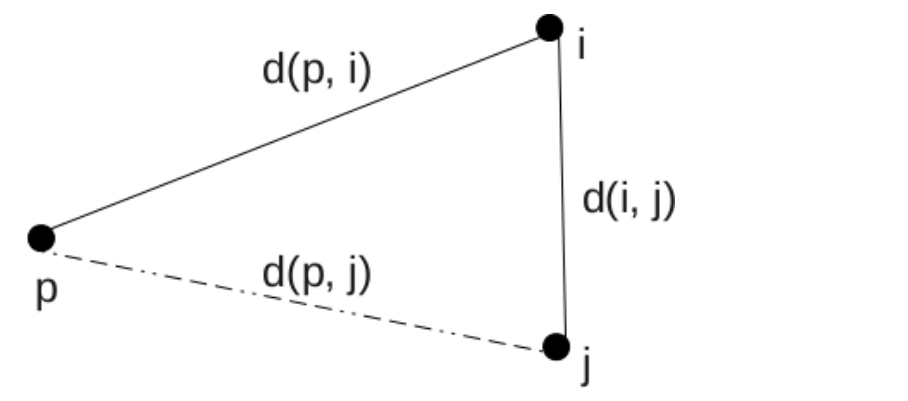
### 2.1 The pKluster distributed memory parallel $k$-means code

- Originally developed in 1996–1997 for use on the Stone Soupercomputer, a very early Beowulf-style cluster constructed entirely out of surplus parts (see "The Do-It-Yourself Supercomputer", *Scientific American*, 265 (2), pp. 72-79, 2001.)

- Because of extreme heterogeneity of the cluster, a master-slave parallel programming paradigm was used, as this provided excellent dynamic load-balancing.
- On modern, homogeneous machines, the master-slave paradigm may be less efficient than a fully-distributed, masterless approach.
  - We have explored the masterless approach in a prototype rewrite of the code.
  - We work with the master-slave version here, because some techniques described below introduce load imbalance even on homogeneous machines.
- When pKluster was initially written, on-node parallelism was virtually nonexistent on commodity PCs; the focus was purely on distributed-memory parallelism.
- Features:
  - Planned open-source release under the Apache License 2.0.
  - Runs on any machine (or cluster) with C89 (or higher) C compiler and an MPI implementation.
  - Option to improve cluster quality by moving or "warping" clusters that become empty to locations in data space where points that are farthest from their current cluster centroids reside.
  - Implements "accelerated" $k$-means algorithm.
  - Optimizations for manycore CPU and GPGPU systems.
  - Coming soon: Support for clustering observation vectors with many zero entries (e.g., species occurrence data).

### 2.2 "Accelerated" $k$-means Algorithm

- For very large datasets and/or cases when the number of clusters $k$ is large, straightforward implementation of $k$-means proves too expensive, even when using many compute nodes.
- We "accelerate" the k-means process using two techniques described by Phillips (doi:10.1109/IGARSS.2002.1026202):
  - Use the triangle inequality to eliminate unnecessary point-to-centroid distance computations based on the previous cluster assignments and the new inter-centroid distances.
  - Reduce evaluation overhead 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 are needed, as the nearest centroid is known from a previous evaluation.



$d(i, j) \leq d(p, i) + d(p, j)$
$d(i, j) - d(p, i) \leq d(p, j)$
if $d(i, j) \geq 2d(p, i)$ :
$\quad d(p, j) \geq d(p, i)$
$\quad$ without calculating the distance $d(p, j)$

**Figure 2:** *The triangle inequality is used to eliminate unnecessary distance calculations.*

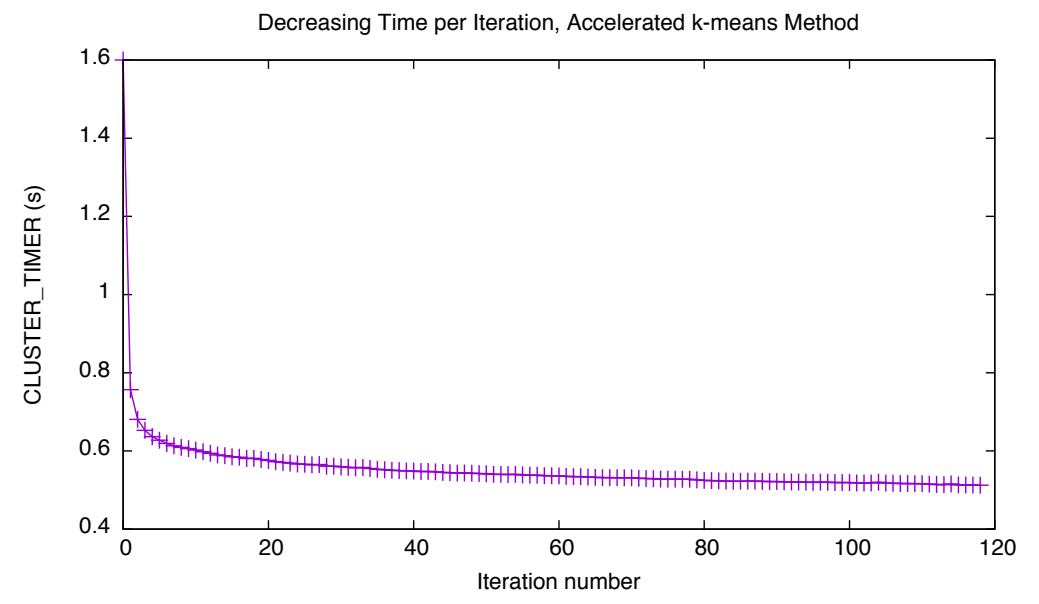


**Figure 3:** *Clustering the GSMNP LiDAR dataset from section 4.2 for $k = 2000$ with the accelerated $k$-means algorithm on the* `BDW` *system. Time for each iteration decreases as the accelerated algorithm is able to avoid many distance comparisons.*

- We also improve cluster quality by moving or "warping" clusters that become empty to locations in data space where points that are farthest from their current cluster centroids reside.

## 3. Optimizations for Novel Computing Architectures

**Computer test platforms used**

| Name | Description |
|---|---|
| `BDW` | Intel Xeon E5-2697 v4 ("Broadwell") node<br>(2.3 GHz core freq; 2 sockets; 18 cores per socket; TURBO off) |
| `KNL` | Intel Xeon Phi 7250 ("Knights Landing") node<br>(68 cores, 272 threads; 1400 MHz core freq; TURBO off) |
| `Titan` | One AMD Opteron 6274 ("Interlagos") 16-core CPU and<br>one NVIDIA Kepler GPU per node; 18,688 total nodes |

- Parallelism within compute nodes has been greatly increasing:
  - GPGPUs from AMD and NVIDIA can execute thousands of simultaneous threads.
  - The second-generation Intel Xeon Phi processor has up to 72 cores (4 hyperthreads per core); each core has two 512-bit vector processing units.
- We have made several adaptations to pKluster to support these architectures.

### 3.1 Improving Computational Intensity Using Level 2/3 BLAS

- We recently realized that it is possible to achieve greater computational intensity of the observation–centroid distance calculations by expressing the calculation in matrix form:
  - For observation vector $x_i$ and centroid vector $z_j$, the squared distance between them is $D_{ij} = \|x_i - z_j\|^2$.
  - Via binomial expansion, $D_{ij} = \|x_i\|^2 + \|z_j\|^2 - 2x_i \cdot z_j$
  - The matrix of squared distances can thus be expressed as $D = \overline{x}\mathbf{1}^{\intercal} + \mathbf{1}\overline{z}^{\intercal} - 2X^{\intercal}Z$, where $X$ and $Z$ are matrices of observations and centroids, respectively, stored in columns, $\overline{x}$ and $\overline{z}$ are vectors of the sum of squares of the columns of $X$ and $Z$, and $\mathbf{1}$ is a vector of all 1s.
- The above expression can be calculated using level 2 and 3 Basic Linear Algebra Subprograms (BLAS) operations, which admit very computationally efficient implementations.
- We have used the highly optimized BLAS implementations from Intel's MKL and NVIDIA cuBLAS to speed up distance calculations on Xeon Phi and GPGPUs, respectively.
- Distance calculations using the above formulation are dramatically faster than the straightforward loop over vector distance calculations when many distance comparisons must be made.

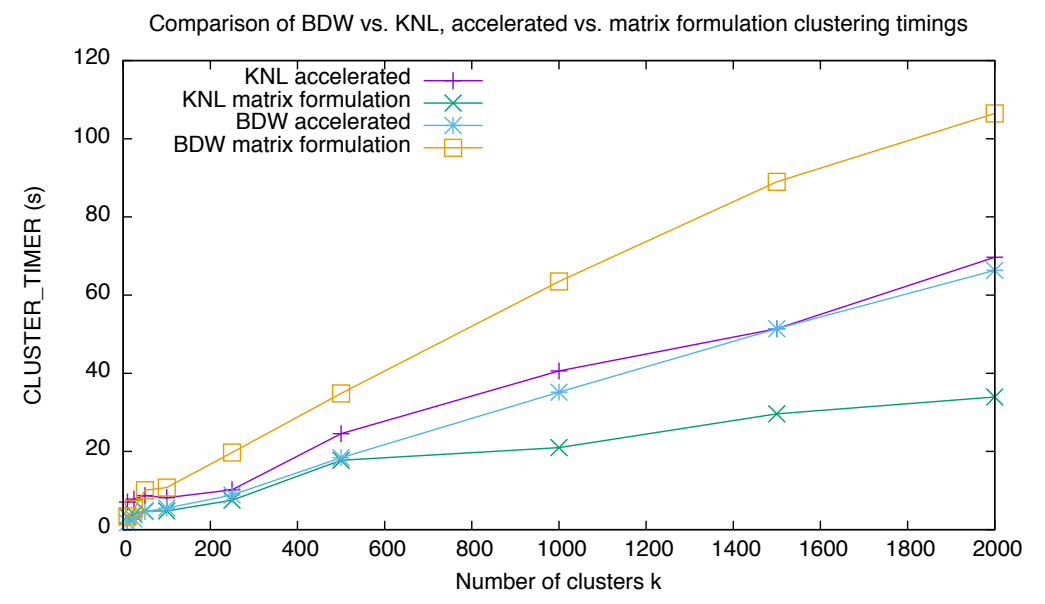


**Figure 4:** *Clustering the GSMNP LiDAR dataset from section 4.2 for different values of $k$ on* `BDW` *and* `KNL` *using accelerated $k$-means and the matrix formulation that uses level-2/3 BLAS calls. Although requiring many more distance calculations, the efficiency of the calculations on KNL is so high that it outperforms the acceleration scheme in all of our tests. On BDW, the matrix formulation only speeds up initial iterations (when many distance comparisons are required); after that, the accelerated approach results in much faster iterations.*

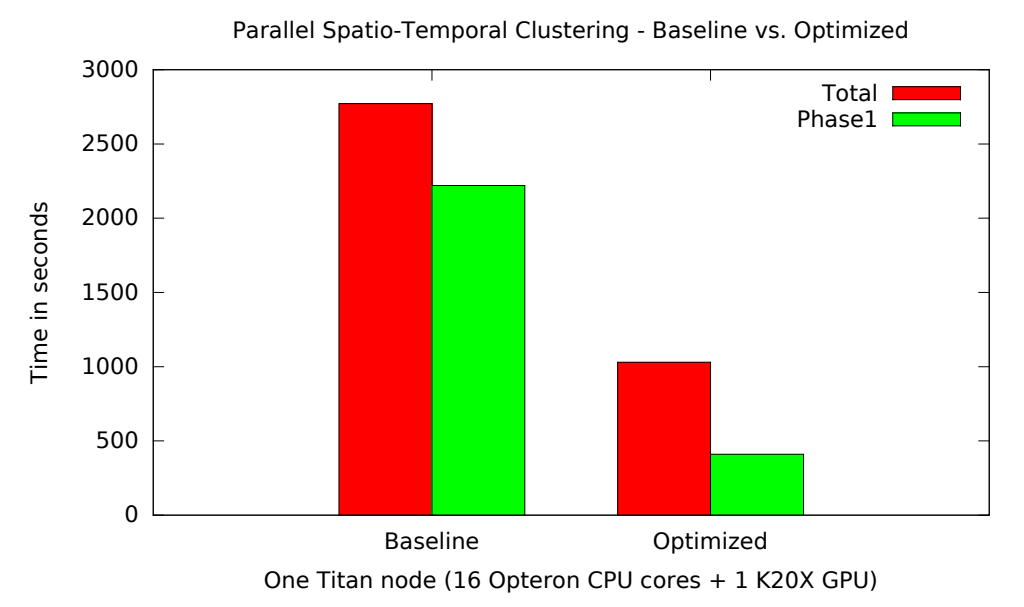


**Figure 5:** *Performance comparison of the baseline and optimized (using cuBLAS on the GPU) code versions for finding 8,000 clusters using the Global Climate Regimes (dimension 123,471,198 × 17) data set on one node of* `Titan`.

## 4. Climate and Ecology Applications

### 4.1 MODIS-based Phenoregionalization and Change Detection

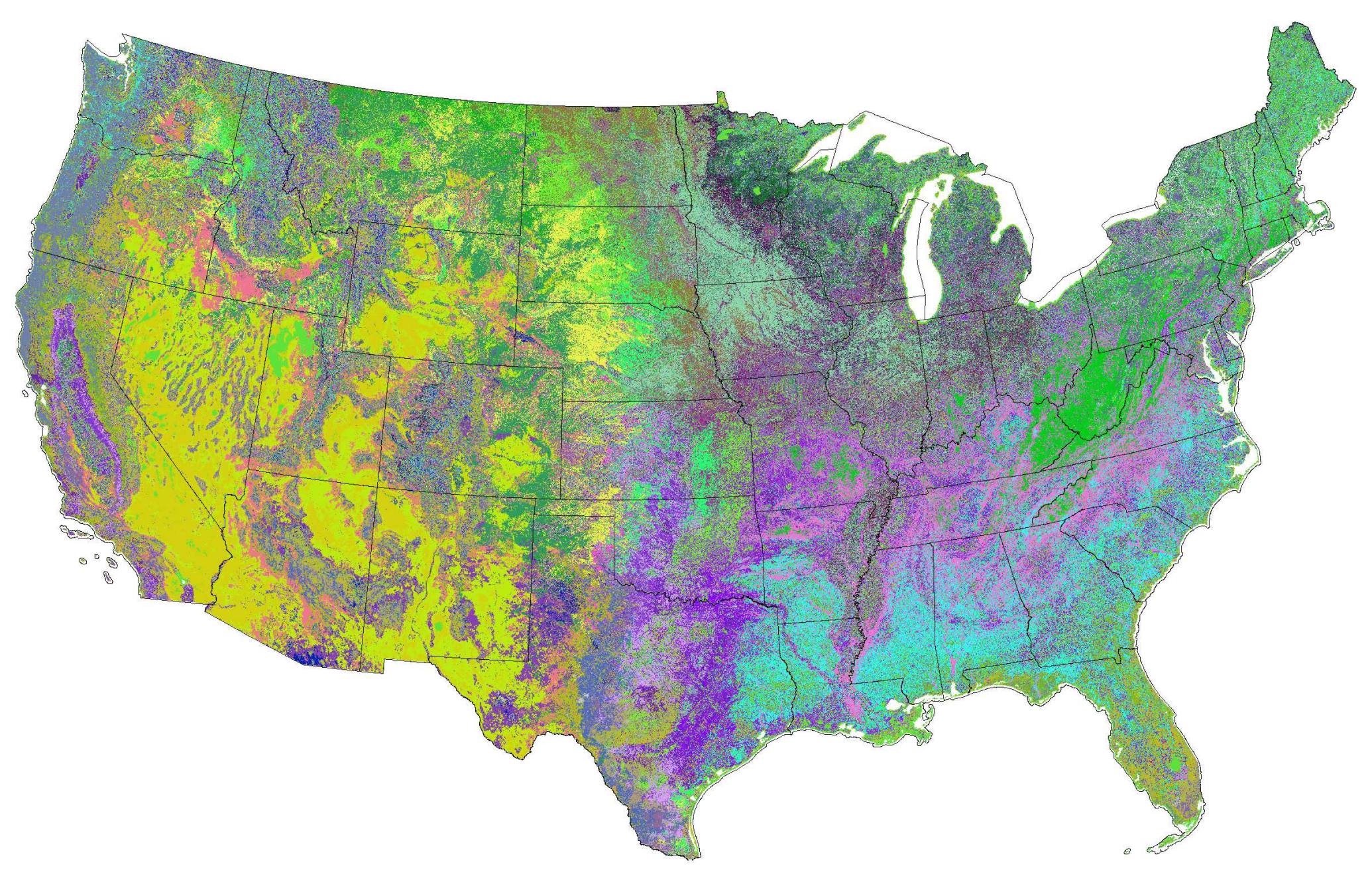


**Figure 6:** *A map of "phenoregion" assignments for the year 2012, based on k-means analysis with $k = 50$ of the entire MODIS-derived ForWarn NDVI product for years 2000–2012. The body of observation vectors being clustered consists of the year-long MODIS NDVI time series for every map pixel, for each year. The map indicates cluster membership (in random colors) for the phenology observed in 2012 at each map pixel.*

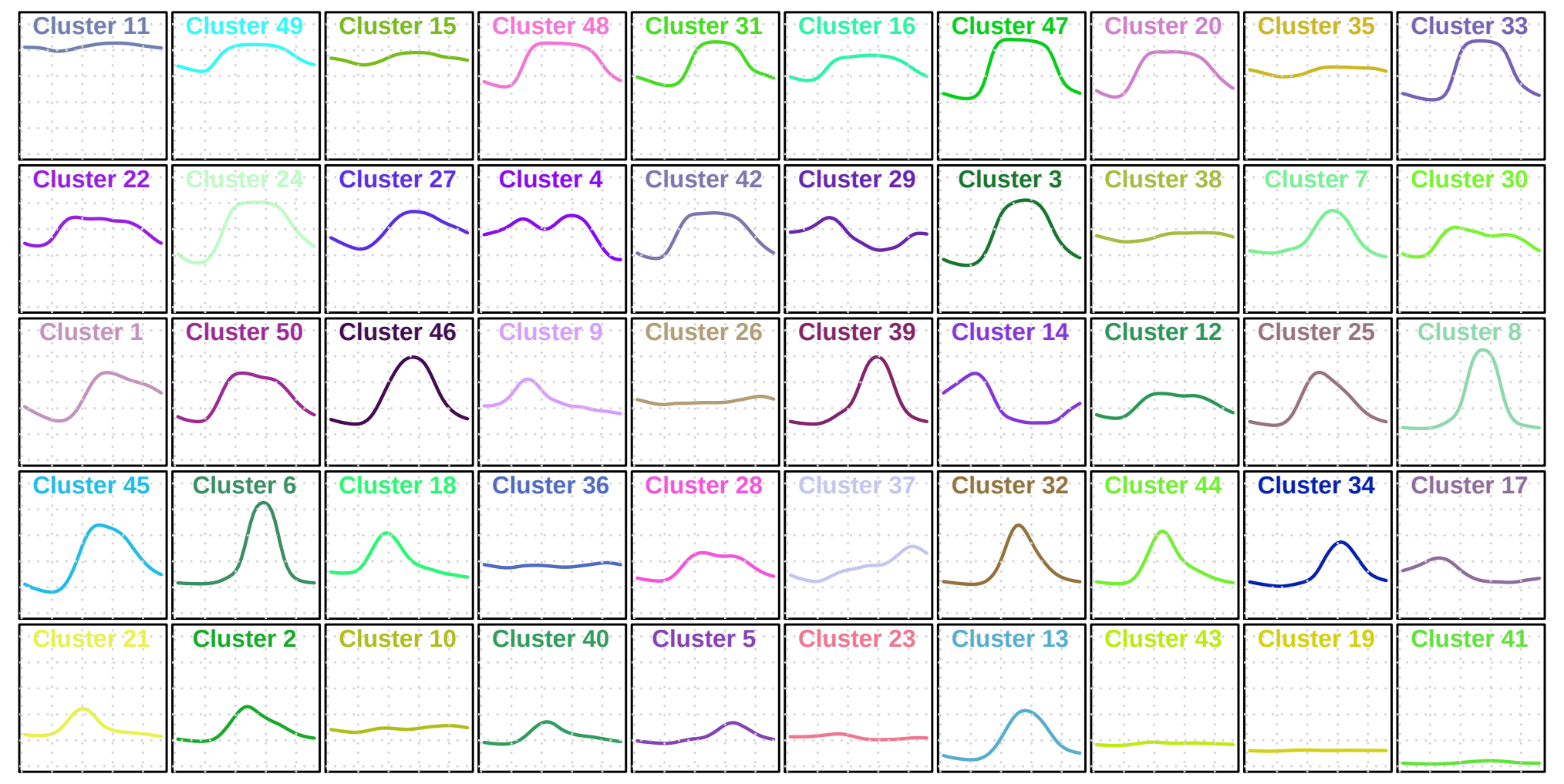
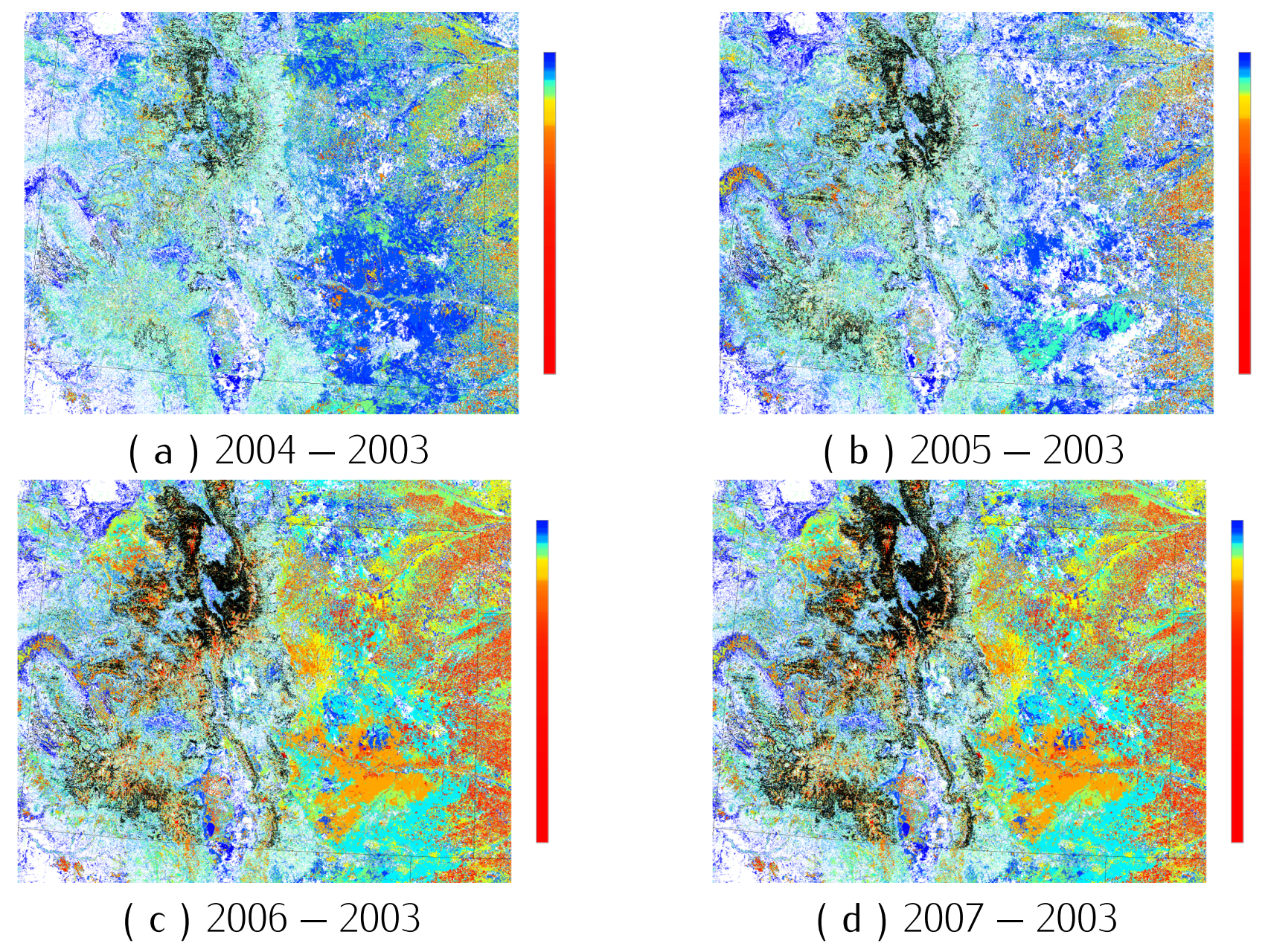


**Figure 7:** *The fifty centroids (corresponding to "phenoregion" prototypes) used for the membership assignments in the map in Figure 6. The colors the centroid plot correspond to the map colors.*



( a ) 2004 – 2003

( b ) 2005 – 2003

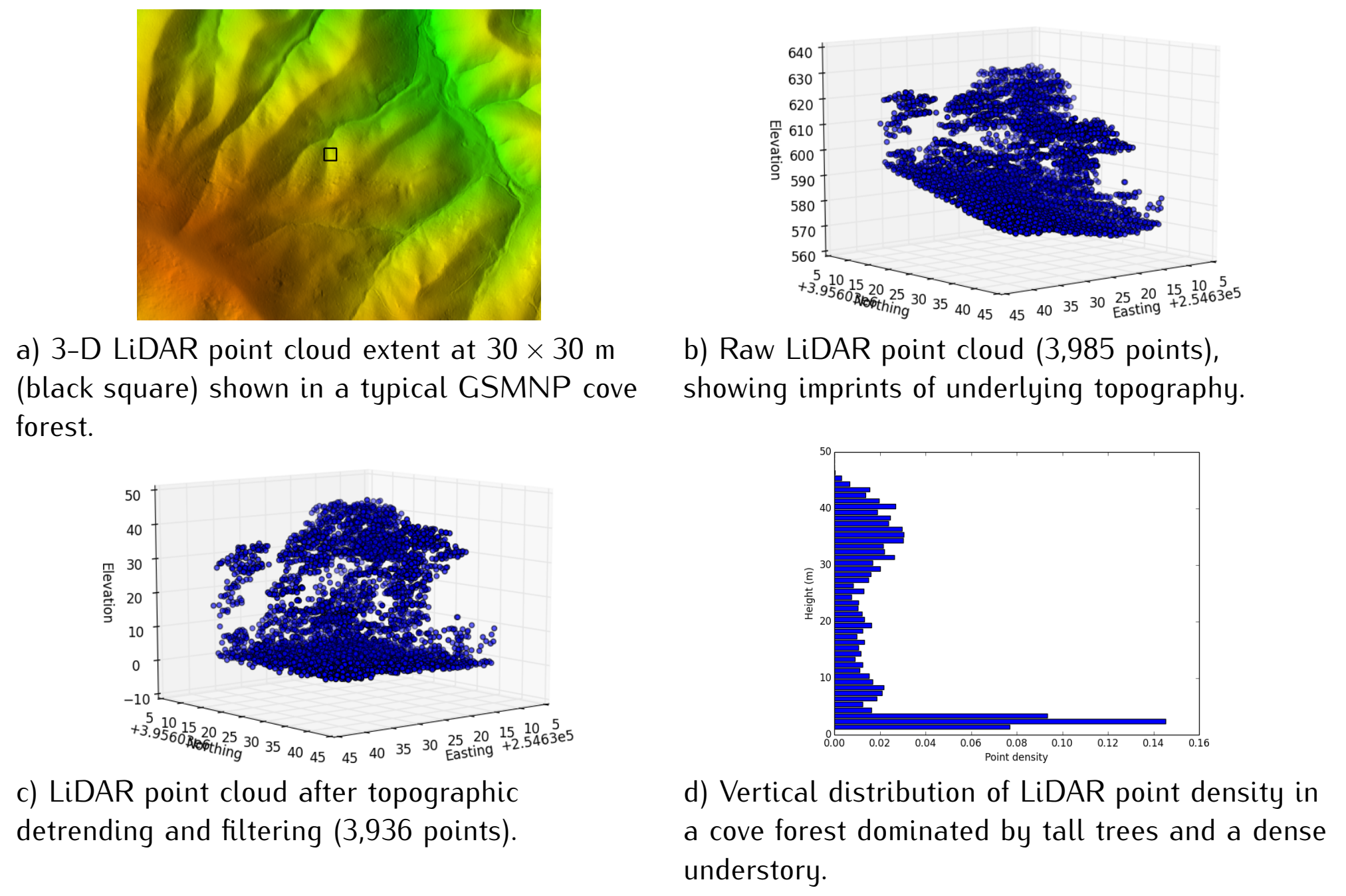
( c ) 2006 – 2003

( d ) 2007 – 2003

**Figure 8:** *Maps showing the relative state space transition distances (how different phenoregion assignments are for given years) between years in Colorado and southern Wyoming. Pine beetle mortality correlates strongly with high transition distances. Black-outlined polygons are disturbed areas indicated on aerial sketch maps.*

### 4.2 Classification of Vegetation Canopy Structure using LiDAR

- Airborne Light Detection and Ranging (LiDAR) enables large scale remote sensing of topography, built infrastructure, and vegetation structure.
- Multiple laser "returns" produce "point clouds" used to map the ground surface, buildings, roads, and utility infrastructure, and to reconstruct the structure of vegetation canopies.
- Large data volumes (current data set has dimension 3,186,679 × 74) pose significant computational challenges to employing LiDAR to monitor and manage forests and animal habitats.



a) 3-D LiDAR point cloud extent at 30 × 30 m (black square) shown in a typical GSMNP cove forest.

b) Raw LiDAR point cloud (3,985 points), showing imprints of underlying topography.

c) LiDAR point cloud after topographic detrending and filtering (3,936 points).

d) Vertical distribution of LiDAR point density in a cove forest dominated by tall trees and a dense understory.

**Figure 9:** *Shown here are the steps involved in converting a LiDAR point cloud into a vertical vegetation canopy distribution for subsequent cluster analysis.*

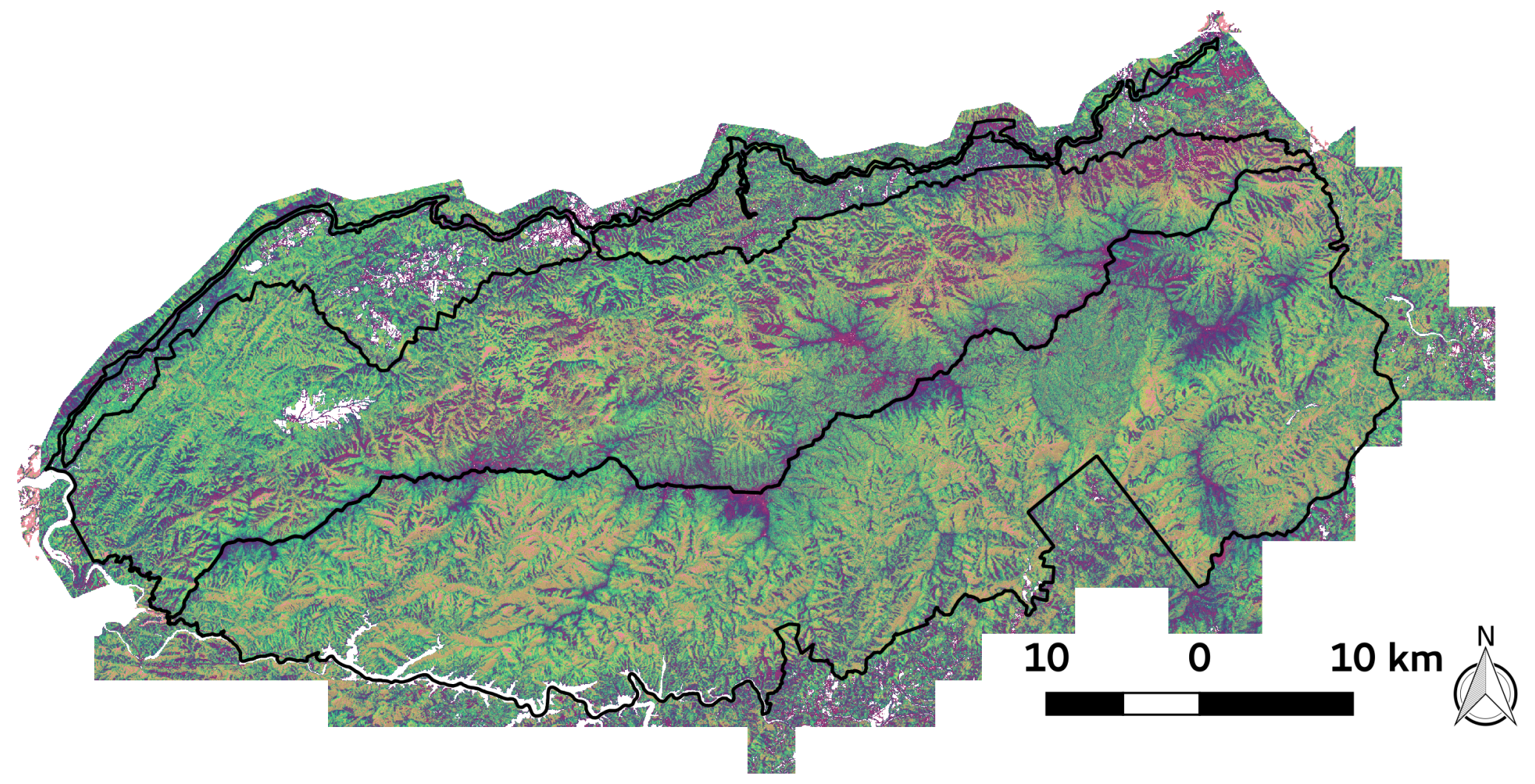


**Figure 10:** *This map shows the 30 most-different classes of vegetation canopy structure, as identified by $k$-means clustering for the Great Smoky Mountains National Park.*

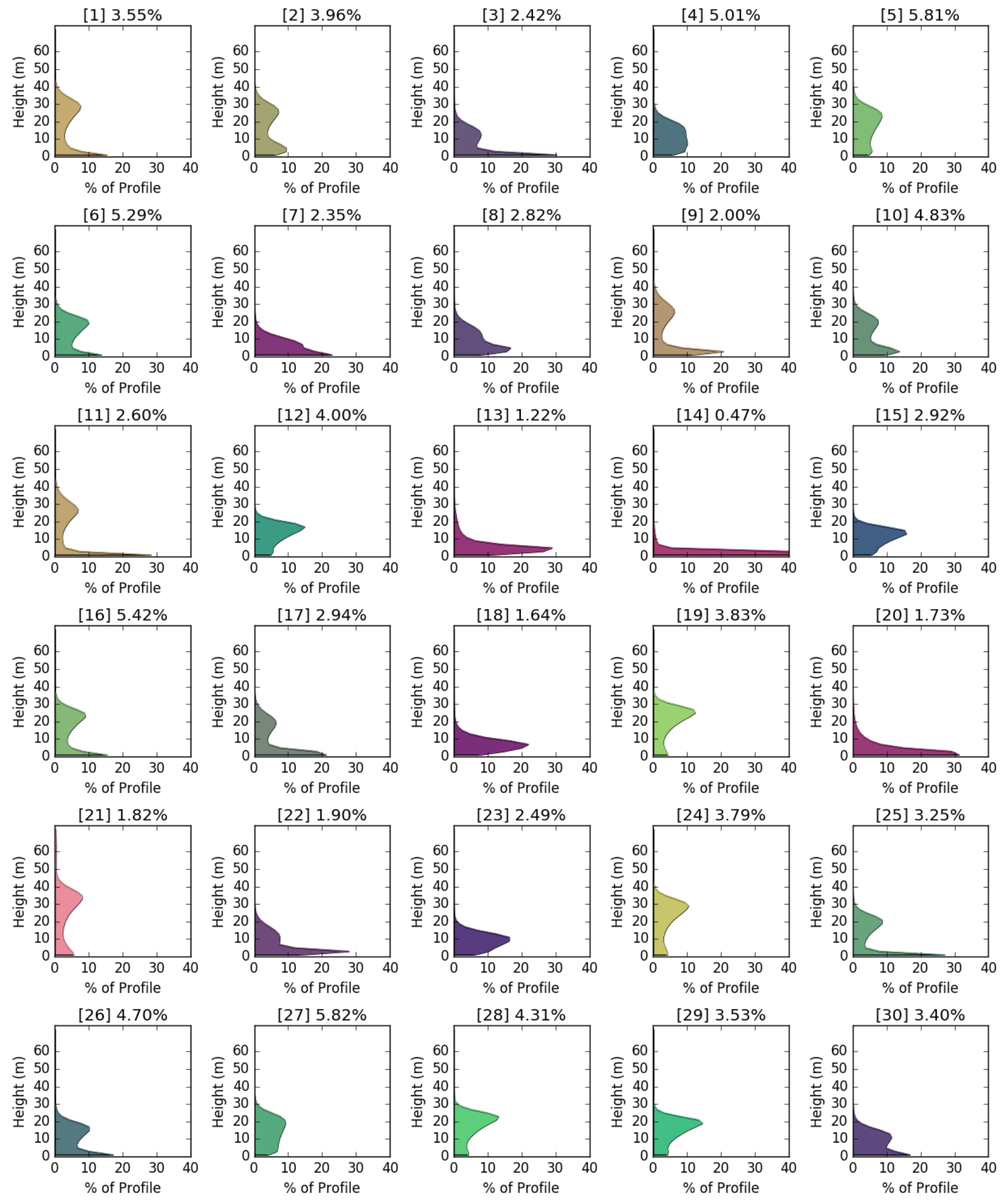


**Figure 11:** *The 30 centroids represent vegetation canopy structure prototypes.*

### 4.3 Analysis of Global Climate Regimes

- We can compute climate-based "ecoregions" using $k$-means analysis of bioclimatic plus ancillary variables.
- Comparing the maps produced for present day and for simulated future conditions facilitates quantitative study of the effects of projected climate change on ecoregion distribution.

**Table 1:** *Variables used for delineation of global climate regimes. Data drawn from Hijmans et al. 2005 [doi:10.1002/joc.1276], Saxon et al. 2005 [doi:10.1111/j.1461-0248.2004.00694.x], Baker et al. 2010 [10.1007/s10584-009-9622-2]*

| Variable Description | Units |
|---|---|
| **Bioclimatic Variables** | |
| Precipitation during the hottest quarter | mm |
| Precipitation during the coldest quarter | mm |
| Precipitation during the driest quarter | mm |
| Precipitation during the wettest quarter | mm |
| Ratio of precipitation to potential evapotranspiration | – |
| Temperature during the coldest quarter | °C |
| Temperature during the hottest quarter | °C |
| Day/night diurnal temperature difference | °C |
| Sum of monthly $T_{\text{avg}}$ where $T_{\text{avg}} \geq 5$°C | °C |
| Integer number of consecutive months where $T_{\text{avg}} \geq 5$°C | – |
| **Edaphic Variables** | |
| Available water holding capacity of soil | mm |
| Bulk density of soil | g/cm$^3$ |
| Carbon content of soil | g/cm$^2$ |
| Nitrogen content of soil | g/cm$^2$ |
| **Topographic Variables** | |
| Compound topographic index (relative wetness) | – |
| Solar interception | (kW/m$^2$) |
| Elevation | m |

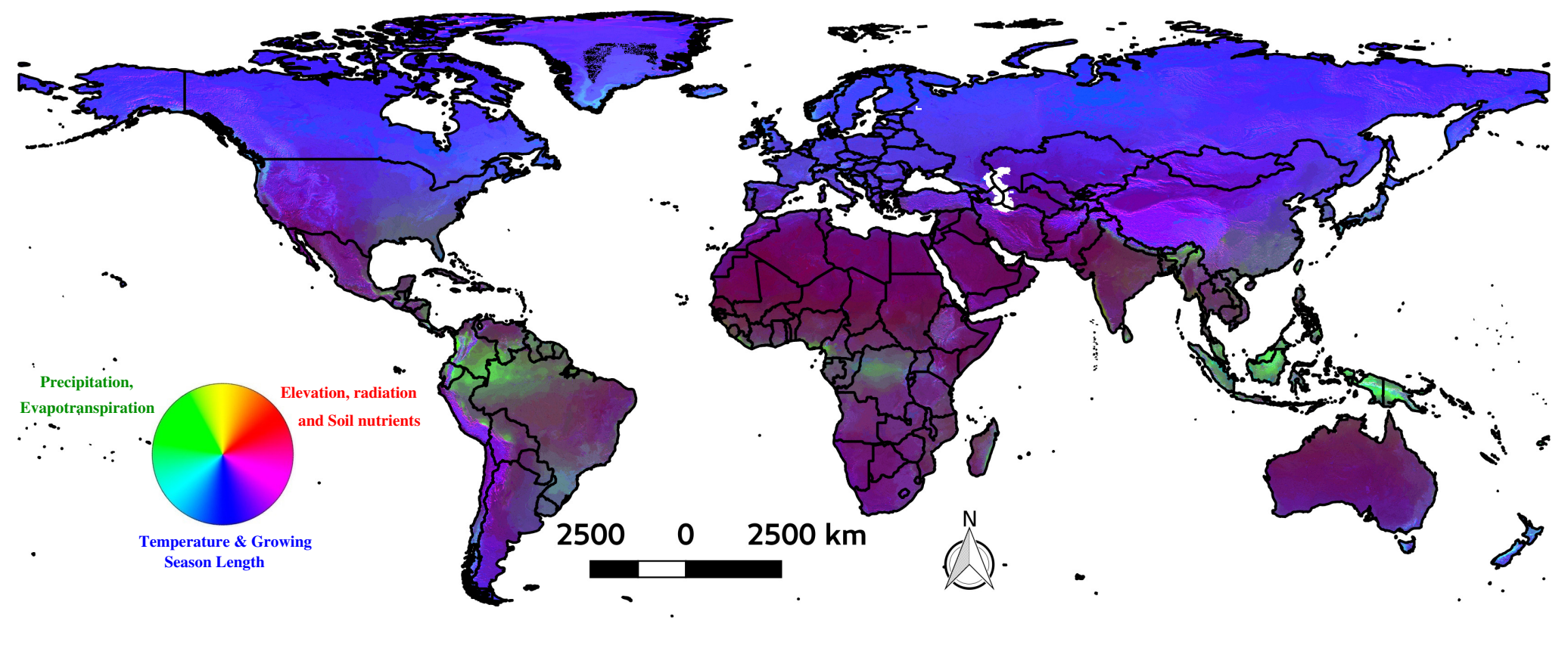

Precipitation, Evapotranspiration
Elevation, radiation and Soil nutrients
Temperature & Growing Season Length
2500
0
2500 km
N

**Figure 12:** *1000 Global climate regimes generated by the $k$-means clustering algorithm for contemporary time period. Clusters are colored according to a* similarity color *scheme using the top three components from principal components analysis. The red color channel largely reflects topography and soil properties; the green channel, precipitation variables and evapotranspiration; and the blue channel, temperature variables and growing season length.*

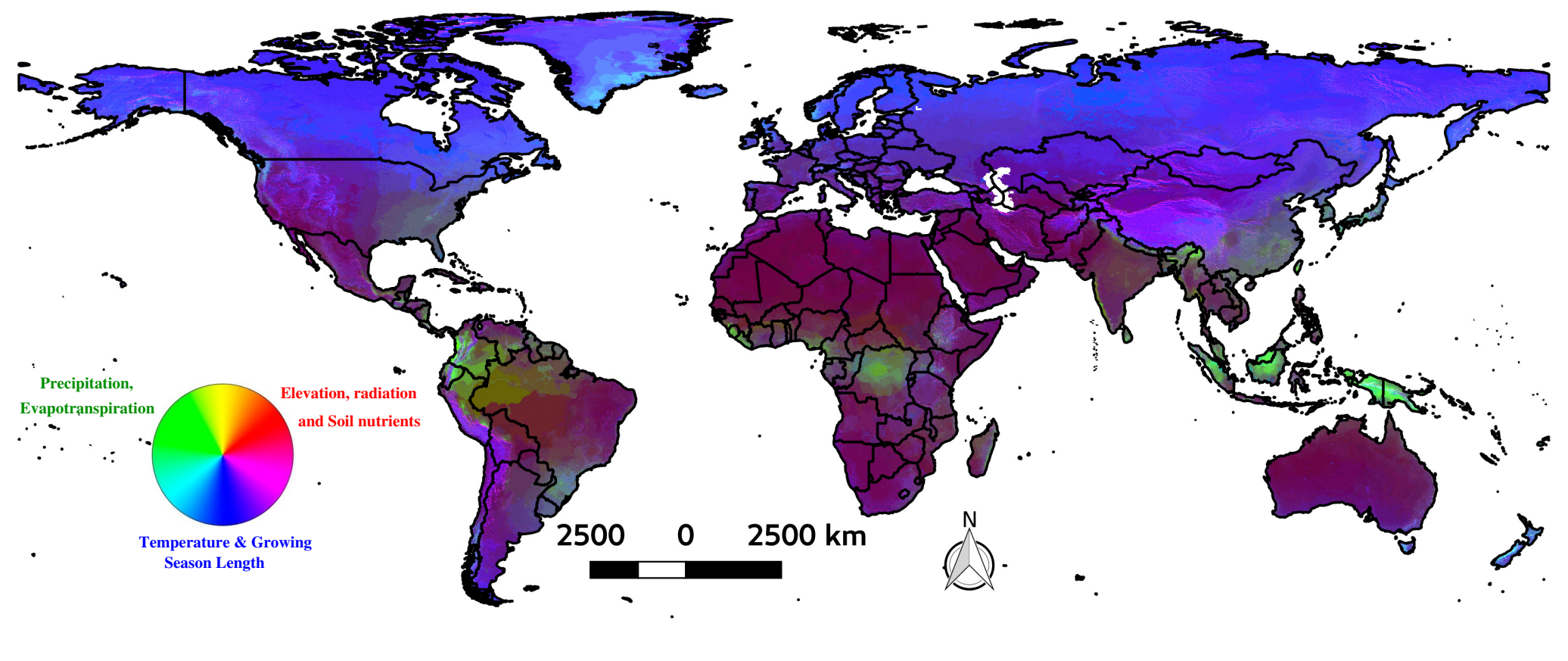

Precipitation, Evapotranspiration
Elevation, radiation and Soil nutrients
Temperature & Growing Season Length
2500
0
2500 km
N

**Figure 13:** *1000 Global climate regimes generated by the $k$-means clustering algorithm for predicted future 2100 by HadCM3 climate model under A1FI emissions scenario. Clusters are colored via similarity colors as in Figure 12.*

## 5. Future Directions

### 5.1 Further Improvements to pKluster

- Re-implement a fully distributed, masterless approach in the current version of the code to handle cases in which master-slave overhead is high (e.g., many cases on `KNL`).
- Add support for emerging high-capacity, non-volatile memory technologies.
- Investigate hybrid approach combining accelerated $k$-means method and matrix formulation within the same iteration.

### 5.2 Possible Science Goals

- Potential questions of interest:
  - How are global plant distributions affect by climate change?
  - What are the implications for global carbon budgets and feedbacks to climate?
  - What changes do we expect to key events like onset of growing season?
  - What changes do we expect to suitable growing ranges for crops?
  - Are there policy implications for agriculture and ensuring the food supply?
- Could combine analysis to all of the MODIS vegetative phenology record with global fine-scale meteorological reanalysis and possibly other ancillary data layers.
  - Enables attribution of vegetation changes to climate or other events.
  - Study directly observed vegetation responses to extreme events.
- Could analyze high-resolution and/or multi-model ensemble Earth system model simulations:
  - Project changes to distribution of eco-phenoregions (identified by the historical analysis) for different climate change scenarios.
  - Combine with crop physiology models to project changes in yields.
  - Combine with urban growth models or population models to assess resource planning, policy scenarios, and crop futures.
- Potential collaborators: Beta users of pKluster are welcome! What is **your** scientific question?