Block Cyclic Distribution of Data in pbdR and its Effects on Computational Efficiency

UMBC REU Site: Interdisciplinary Program in High Performance Computing

Team members: Matthew Bachmann[1], Ashley Dyas[2], Shelby Kilmer[3], and Julian Sass[4]

Graduate assistant: Andrew Raim[4], Faculty mentor: Nagaraj K. Neerchal[4], Clients: George Ostrouchov[5] and Ian F. Thorpe[6]

[1]Northeast Lakeview College, [2]Contra Costa College, [3]Bucknell University, [4]Math. and Stat., UMBC, [5]Oak Ridge Nat. Lab., [6]Chemistry, UMBC

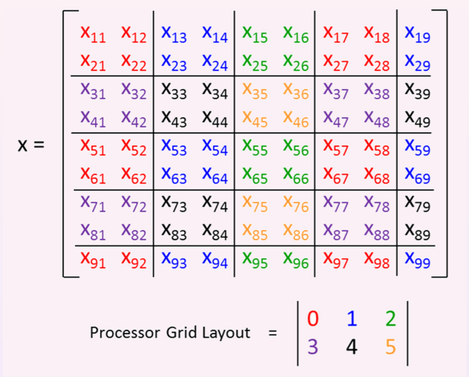
Summary

- pbdR is an R package used to implement high-performance statistical computing on very large data sets.
- Block cyclic arrangement is used in pbdR (through ScaLAPACK) for distributed operations on large, dense matrices amongst parallel processes.
- Selection of block size and grid processor layout greatly influence computational efficiency.

In this work, we explore block cyclic distribution by implementing the statistical method PCA. We illustrate a large-scale PCA with an application to protein movements, focusing on motional correlations.

Background

Block Cyclic Distribution



Here a 9×9 matrix is block cyclically divided with block size of 2×2 and processor grid layout of 2×3. Each colored 2×2 partition will be distributed to the process in the 2×3 grid with the corresponding color.

SNP and its Applications SNPs are variations that occur in a DNA sequence when a nucleotide differs between paired chromosomes, changing the amino acid sequence. These can be analyzed through motional correlations.

References and Acknowledgments

References at www.umbc.edu/hpcf > Publications:

- Full Technical Report: HPCF–2013–11
- A. M. Raim, pbdR Tutorial, HPCF–2013–2.

Acknowledgments:

- **REU Site:** www.umbc.edu/hpcreu
- NSF, NSA, UMBC, HPCF, CIRC

Methods

PCA

Principal component analysis (PCA) determines the directions of maximal variability in large dimensional data and is widely used for dimensional reduction. PCA is implemented on the sample covariance matrix:

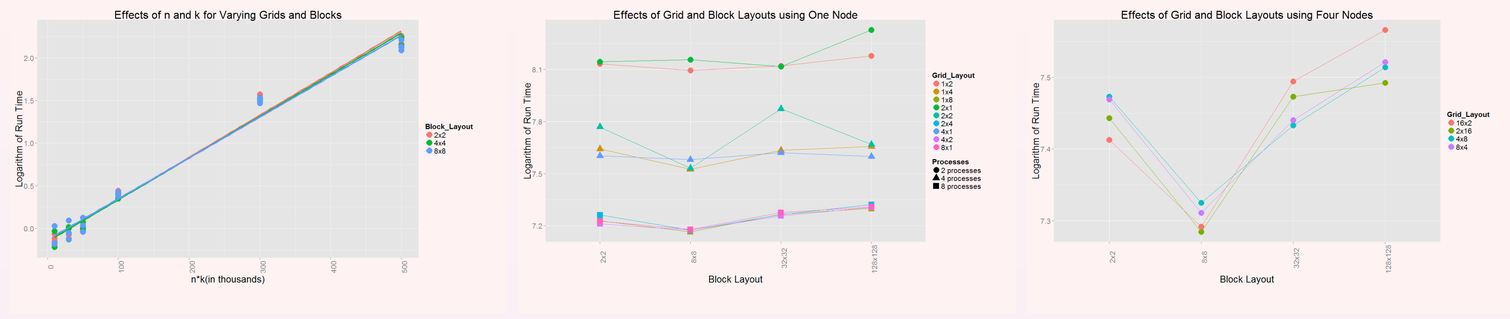
$$\mathbf{S} = \frac{1}{n-1}\mathbf{Y}'\left(\mathbf{I} - \frac{1}{n}\mathbf{J}\right)\mathbf{Y}, \quad \text{where } \mathbf{J} = \mathbf{1}\mathbf{1}' \text{ and } \mathbf{Y} = \begin{bmatrix} y_{11} & \cdots & y_{1k} \\ \vdots & \ddots & \vdots \\ y_{n1} & \cdots & y_{nk} \end{bmatrix}$$

$\mathbf{S}$ is positive semi-definite and therefore can be diagonalized: $\mathbf{\Lambda} = \mathbf{O}'\mathbf{S}\mathbf{O}$, where $\mathbf{O}$ is an orthogonal matrix. Each row of $\mathbf{O}$ is an eigenvector of $\mathbf{S}$. These eigenvectors are called principal components. $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues of $\mathbf{S}$. The number of significant eigenvalues determines the true underlying dimensionality of the data.

We analyze the effects of several factors of the PCA algorithm on computational speed:

- Vary the size of the $n \times k$ data matrix
- Vary block size and grid layout on a single node for constant n and k
- Extend study of block size and grid layout to multiple nodes
- Demonstrate real-world applicability

Results

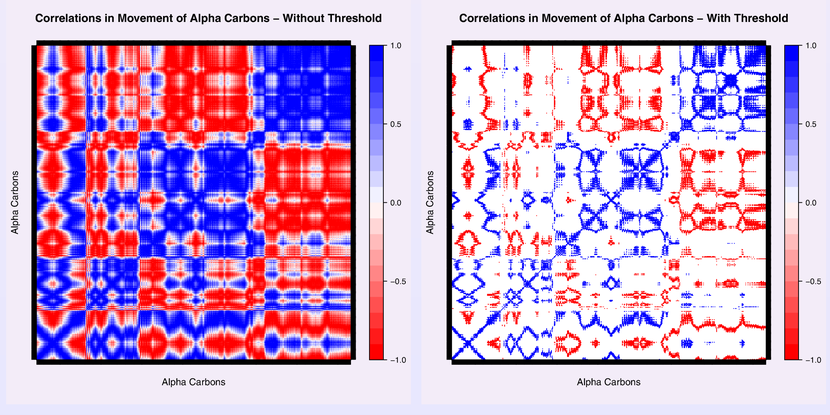


The left figure shows the predictible nature of the effects n and k have on efficiency. The center and right figures show the effects of grid layout and block size on computational speed for a constant n and k.

ANOVA Results:

Source	DF	One Node Study			Four Node Study		
		SS	MS	F	SS	MS	F
Grid	3	0.0008	0.0003	3.0000	0.0007	0.0002	0.2593
Block	3	0.0393	0.0131	131.0000	0.1040	0.0347	38.5556
Residuals	9	0.0011	0.0001		0.0080	0.0009	

These tables show ANOVA results for one node 8 processes on the left, and for four nodes 8 processes on the right. Results show that block size has a clear effect on computational speed. 8×8 block sizes are the most efficient for this n and k.



These figures show the level plot of Dr. Thorpe's data before and after PCA and greying out all statistically non-significant correlations. Note that relatively few correlations are statistically significant.