Metagenomics is a fast growing field within computational biology that applies genomic tools and techniques to study and characterize the microbial communities living in a broad range of environmental habitats e.g. the human gut, ocean floor, soil. The field has an immense potential to advance the state of discovery in many grand challenge areas such as human health, bioenergy alternatives and environmental sustainability. Recent advances in sequencing technologies have spurred on a number of environmental sequencing projects resulting in a wealth of data – for instance, the Genome OnLine Database reports about 400 ongoing metagenomics studies, and the CAMERA repository contains over a billion samples. Graph-theoretic modeling is a popular way to represent the underlying computational biology problems because it provides an effective way to capture and specify the interplay among the different biomolecular entities that constitute a naturally built environmental microbial system.

In this study, we are focusing our efforts on building scalable clustering methods that can efficiently cluster and identify strongly knit communities within large biological graphs constructed out of high-throughput metagenomics sequence data. Clustering can serve multiple purposes – for reducing the redundancy within sequence repositories, for functionally annotating microbial communities through the identification and characterization of protein families, for discovering groups and linkages across multiple types of data, etc. Owing to the computational hardness of clustering under most theoretical formulations, efficient heuristics need to be used in practice. However, the execution of even these heuristic approaches on modern day data sets could require tens of thousands to millions of CPU hours. Despite this growing need for parallelization, most graph clustering heuristics are inherently sequential, and implementing these heuristics under a parallel setting becomes a daunting task due to a combination of factors including dependence of computational workload on the input and irregular data access and movement patterns. In this study, we present novel clustering heuristics that are better suited to overcome the above outlined scalability limitations while also significantly enhancing the quality of the output clustering.

The key contributions can be summarized as follows:

1. **Design and evaluation of clustering heuristics for large-scale biological graphs with or without edge weight information.**

In developing our clustering heuristics, we draw ideas from a randomized sampling heuristic called Shingling that was originally developed by Gibson et al. The strengths of this heuristic are that it is highly suited to detect densely connected subgraphs within large graphs and is also amenable to parallelization. As our preliminary work, we implemented and evaluated this heuristic for identifying protein clusters within real world metagenomics homology graphs. While the experiments showed a general effectiveness in clustering [Rytsareva et al., 2013, IJHPCN], we also observed a tendency of the heuristic to leave out a number of low degree vertices as singletons. Enhancing clustering sensitivity is difficult and yet important particularly in the context of metagenomics data where the sampling rates are low and the degree of community annotation is affected by the loss of information during cluster recruitment. To overcome this limitation, we designed a novel variant of the standard Shingling heuristic that has demonstrated significant qualitative improvements with minimal impact on performance. We also propose a way to extend this heuristic to handle *weighted* graph inputs. The main idea is to recalculate the edge weights of all edges to represent their relative importance to a given vertex prior to randomized sampling.

2. **Design and implementation of novel MapReduce algorithms for efficiently parallelizing the proposed clustering heuristic.**

We developed parallel algorithms using the MapReduce paradigm for executing the above clustering heuristics on large-scale distributed memory clusters running Hadoop and MPI. We chose to use the MapReduce paradigm because the fundamental operations required to implement the Shingling heuristic naturally lend themselves to a map-reduce structure.

3. **Application of the proposed algorithms for clustering real-world graphs in metagenomics and other scientific domains.**

As a concrete real world application, we applied our methods on a large-scale ocean metagenomics network containing 10.3M vertices and 640M edges and its subsets. Our initial results demonstrate a) significant improvement in quality over both the standard Shingling heuristic and other serial heuristics by creating clusters with very high modularity (>0.9) and intra-cluster density (~0.8) while drastically reducing the singletons from ~27% to under 1%; and b) significant scaling potential – for instance, the analysis of the entire test graph took less than a minute on 4,096 core of the NERSC Hopper supercomputer. In summary, our implementation has shown linear scaling on thousands of processors and multiple orders of magnitude improvement in both on time to solution and problem scaling.

Future research includes incorporating cluster composition information as part of qualitative study and an application on larger real world graphs. We are also exploring a way to make the graph kernels and related clustering modules accessible to biologists and other non-HPC users as a transparent scientific workflow.