The proliferation and availability of medical data to researchers and health care professionals provide many opportunities for improving patient care, while also presenting difficult challenges arising from incomplete data, computational scalability, and proper application of algorithms. In this proposal we will investigate a system for assisting researchers in overcoming the listed difficulties, while also allowing them to scale with the ever increasing datasets available to health care professionals. In our proposal we will develop a novel system for large scale data clustering and demonstrate its applicability for the analysis of medical datasets. Health fields benefited from advances in large scale clustering include advances for medical resource allocation, temporal pattern mining of physiological metrics and biological processes, as well as more straightforward applications of clustering such as, image recognition, image segmentation, and many clustering problems in genomics.

In this proposal, we propose a clustering based tool to assist health professionals in using the increasingly large body of available medical information for making better informed decisions. Data clustering is the process of grouping like objects based on their similar attributes. Clustering is particularly useful in cases where complete data is unavailable, or experimentation is infeasible. Clustering does this by capturing latent unknown attributes that may be otherwise hidden among the complex models defining the underlining biological processes. The use of clustering in health care is not new; however due to a disconnect between the research goals found in computing and health care, newer more scalable clustering algorithms have been overlooked. Though theoretically sound, more common clustering algorithms do not take advantage of the recent advances in big data, parallel processing. One such area that we plan to focus on is scalability. Scalability has become a much larger issue in regards to health care recently, as new open data initiatives are vastly increasing the size of available data sets.

In this proposal we will develop a secure and scalable parallel system for data clustering using already deployed and available, cloud-based, distributed computing resources (e.g. Amazon EC2, Cloudera). In addition, we will demonstrate the efficacy, robustness, and speed of our system on the ill-posed problem of patient outcome prediction from temporal physiological data metrics [1] from the MIMICII [2] dataset.

**AIM 1. DEVELOP A SCALABLE PARALLEL CLUSTERING ALGORITHM BASED ON OUR NOVEL RANDOM PROJECTION HASH CLUSTERING (RPHash) ALGORITHM** 

RPHash is a novel method proposed for large scale parallel data clustering based on multi-probe random projection motifs of high dimensional data vectors hashed into the unique subset boundaries of the Leech Lattice. The system designed will focus heavily on communication scalability and will be compatible with commercial, cloud-based, distributed computing resources (e.g. Amazon EC2). The proposed system will provide a straightforward interface running on the popular MapReduce (MR) [3] parallel processing paradigm, for use by researchers and medical professionals. The clustering system proposed will also have wide applicability to a variety of biological data analysis problems. Furthermore, as the processing back-end system is commercial cloud-based, it will allow researchers to have access to always state of the art computing resources with the option to scale computation requirements to fit their particular data processing needs.

**AIM 2. TO PROVE THE ACCURACY AND EFFICACY OF OUR SYSTEM, WE WILL EMPLOY IT IN CLUSTERING TEMPORAL PHYSIOLOGICAL DATA COLLECT FROM ICU PATIENTS.**

In Aim 2 of the proposal, we will develop an application to predict patient outcomes, in particular mortality, using physiological metrics gathered from ICU patients. The initial task will follow a similar line of research as those presented in Marlin et al. [1]. We plan to improve upon the results of Marlin et al. using our scalable algorithm and the ICU physiological dataset in its entirety, a shortcoming of Marlin et al. that results from the computational complexity growth of the Expectation Maximization (EM) algorithm. As our algorithm is more closely related to fuzzy c-means clustering FCM [4], and thus missing some of the formal statistical analysis of EM algorithm, however has been shown to perform well on incomplete data [5] [6]. Due to a related, but not direct comparison with the EM algorithm used by Marlin et al., we will focus on more qualitative classification metrics such as precision-recall, ROC, and F-Measure.
AIM 3. APPLY TIME INVARIENT METHODS TO TEMPORAL PHYSIOLOGICAL DATA TO COUNTERACT SKEW AND LAG EFFECTS COMMON BETWEEN SIMILAR PATIENT’S DATA SEQUENCES. A shortcoming stated by Marlin et al.’s analysis of patient data is its dependence on a static physiological data. Under the assumptions of static data, trends in collected data are ignored. Furthermore, similarities between misaligned patients will not be discovered due to the effects of skew and individual lags in biological processes. To overcome the listed shortcomings, we will investigate time invariant sequence analysis in addition to raw instance metrics clustering. The addition of time invariant transformations to patient data will allow us to discover similarities between misaligned patients static data metrics as well as similarities in patient trends. AIM 3. serves as an exploratory investigation in improving the results of Marlin et al. while also demonstrating the data robustness of our system developed in AIM 1.

AIM 4. DEMONSTRATE THE DATA SECURITY OF OUR SYSTEM OVER DISTRIBUTED DATA RESOURCES. Privacy concerns over attacks on de-identification [7] [8], or de-anonymization of publicly available medical data such as whole genome sequence [9] data are making it difficult for researchers to acquire the data and results they need. Non-distributed data clustering requires the entire dataset to reside on the processing system, while distributed methods often require communication of full data records between nodes. In our approach, non-orthogonal random projection is a destructive function, that provides privacy of the data being processed and communicated. While centroid communication is by definition only transmitting the aggregate of data records, and in the case of patient data, is not any one’s information. This aims primary focus will be to show that original data vectors cannot be recovered after projection by a non-orthogonal random matrices.

SIGNIFICANCE
Clustering is one of the most commonly used methods for the analysis of unlabeled data. The relationship between physiological outcomes and commonly available metrics are often either too complex to apply directly to predicted outcomes or infeasible to obtain. Due to the complexities of such systems, health care providers often rely on models built upon their previous experiences to generate hypothesis. The experience based modeling process is the basis of cluster based prediction. The underlying complex biological processes remain unknown, however a posteriori conclusions can still be made using the similarity between unlabeled sets of data and known, labeled sets of data. Thus allowing health care professionals to predict conclusion despite not having a formal model of the underlying process [10]. Additionally other fields in health and biology are benefited by data clustering scalability. Such fields as Micro Array clustering, Protein-Protein interaction clustering, medical resource decision making, medical image processing, and clustering of epidemiological events all serve to benefit from larger dataset sizes.

In this proposal we plan to create a highly scalable, cloud based data clustering system for health care providers and researchers. This system will allow researchers to scale their clustering problems without the need for specialized equipment or computing resources. Our cloud processing solution will allow researchers to arbitrarily scale their processing needs using virtually limitless commercial processing resources.

According to the results of Marlin et al., The predictability of mortality outcomes based on collected physiological metrics on ICU patients, may be benefited by greatly by increasing the duration and sampling resolution of such samples. Prediction of mortality outcomes also suggests that we can predict other patient outcomes as well resulting in a useful tool for optimally allocating health care resources. As a similar goal, Aim 3 attempts to further increase the predictability of mortality and other outcomes through the application of time invariant data clustering. The techniques investigated will serve to augment and possibly replace clustering of static event metrics. The application of k-ordered outliers also serves as a natural transformation of feature data intrinsic to their clinical use which may be explored in other sequence based clustering methods.

The intrinsic random projection step in our proposed algorithm provides a destructive information step that allows for secure data sharing among institutions while also acting as a fundamental step in the
RPHash algorithm. Recent attacks on de-identification, the government’s response [9] make this an important attribute for any cloud based data analysis algorithm.

INNOVATIONS
AIM 1. proposes a novel clustering algorithm for scalable data clustering in the cloud. The algorithm developed combines a variety of approximate and randomized methods for providing solutions to real world data problems. Our focus on a non-deterministic clustering algorithm is somewhat uncommon in computing, but coincides well with the ill-posed problem of clustering. This follows naturally from a similar inversion regarding clustering and computing, in which real world problems tend to converge faster than theoretically posed problems. Furthermore, the addition of noise to theoretically posed problems often yield more favorable results than the theoretical problem alone. We believe our “computation as a service” model will, by leveraging commercial resources, result in an overall reduction of institutional costs associated with the maintenance and upgrading of computing systems, while also guaranteeing access to virtually limitless resources and perpetually state of the art equipment to researchers.

The goals of AIM 3. propose the use of not yet applied techniques from time invariant clustering to mortality prediction based on physiological metric data. The problem tailored approach of k-ordered outliers described in Aim 3. is also a simple yet novel method for temporal sequence feature extraction in the health care problem space.

APPROACH
AIM 1. DEVELOP A SCALABLE PARALLEL CLUSTERING ALGORITHM BASED ON OUR NOVEL Random Projection Hash (RPHash) ALGORITHM Clustering algorithms offer insight in a multitude of data analysis problems. Though clustering algorithms have known complexity scaling issues [11] [12], they are in general effective due to the nature of the data they are applied to. For parallel processing however, the communicated bottlenecks of clustering tend to dominate the processing complexity. Various parallel clustering algorithms have been proposed [13] [14] [15] to combat this and other issues with varying success on very specific types of data. Many of these algorithms however are not applicable to high dimensional, sparse data and instead exploit the structural of the expected clusters. In RPHash the approach from the ground up is intended for parallel processing architectures. The basic concept of RPHash has been successfully explored on sequential systems and for projections into one dimensional histograms [16]. Despite theoretical results showing that k-means has an exponential worse case complexity [17], many real world problem tend to fair much better under k-means and other similar algorithms. For this reason, clustering massive datasets, though suffering from unbounded complexity guarantees, is still a very active area in computing research. Approximate and randomized methods are common tools for overcoming complexity growth. In our algorithm, called Random Projection Hash (RPHash) algorithm, we utilize both approximate and randomized techniques to provide a scalable, approximate parallel system for massive dataset clustering. An outline of the steps in RPHash is given below, however we would also like to highlight some aspects of its function in regards to randomness and approximation. One way in which the RPHash algorithm achieves scalability is through the generative nature of its region assignment. Clustering region assignments are performed by decoding vector points into partitions of the Leech Lattice. The Leech Lattice is a unique lattice that provides optimal sphere packing density among 24 dimensional regular lattices [18]. Though optimal, due to the curse of dimensionality, the overall density is somewhat sparse, requiring that the algorithm apply shifts and rotations to the lattice to fully cover the \( \mathbb{R}^{24} \) subspace. Furthermore, in general vectors will be greater than 24 dimensions. We cite the Johnson-Lindenstrauss (JL) lemma to provide a solution to this problem (Figure ??). JL states that for an arbitrary set of \( n \) points in \( m \) dimensional space, a projection exists onto a \( d \)-dimensional subspace such that all points are linearly separable with \( \epsilon \)-distortion following \( d \propto \Omega\left(\frac{\log(n)}{\epsilon^2}\right) \). Though many options for projections exists, a simple and sufficient method for high dimensions is to form the projection matrix \( r_{ij} \in \mathbb{R} \) is \( m \times d \) as follows.
Random projection on a plane

\[ r_{ij} = \begin{cases} 
+1, & \text{with probability } \frac{1}{6} \\
0, & \text{with probability } \frac{2}{3} \\
-1, & \text{with probability } \frac{1}{6} 
\end{cases} \]

Approximate Random Projection Matrix [19]

Overview of the sequential algorithm

The basic intuition of RPHash is to combine random projection with discrete space quantization, and regard regions of high density as centroids. According to JL lemma, the sub-projections will conserve the pairwise distances in the projected space for points with \(\epsilon\)-distortion. In addition to compressing a dataset to a computationally more feasible subspace, random projection can also make eccentric cluster more spherical [20] [21].

The next piece of RPHash are discrete space quantizers. For our implementation of RPHash we will rely on the Leech lattice as our region quantizer. The Leech lattice unique lattice in 24 dimensions that is the densest lattice packing of hyperspheres in 24 dimensional space. The Leech lattice provides a useful space quantizer, as the 24 dimensional subspace it quantizes is a low enough dimensionality to see the benefits of random projection [19], while also having dimensionality large enough enough to represent very large datasets with low distortion [22]. Furthermore, the decoding of the Leech lattice is a well studied geometric object [23] [24] [18], with a worse case decoding complexity of 331 operations [25].

Space quantizers have hard margin boundaries and will only correctly decode points that are within the error correcting radius of its partitions. This is an an issue found in approximate nearest neighbor search [26] [27], and we choose to overcome it in a similar way a Panigrahy [26] by performing multiple random projection hashes(\(\Theta(\log(n))\)) of the same data point. Using many random projections of a vector allows the high dimensional vectors to be represented as 'fuzzy' regions probabilistically dependent on its higher dimensional counterpart. Figure ?? shows an example of this process. Though this step performs redundant computation, it is a requirement for compute node independence. For predominantly communication bottleneck algorithms this tradeoff is a desirable compromise. The end goal is to find lattice hash IDs that on average generate more collisions, than others. The the top \(k\) cardinality set of lattice hash ID vector subsets represent regions of high density. The centroids are computed from the means of the non-overlapping subsets of vectors for each high cardinality lattice hash ID. The algo-
Parallel Implementation

Due to the independence of random projection hashing of vectors, much of the \textit{RPHash} algorithm is naively parallel. The overall parallel portion of the algorithm is in fact no more complicated than the standard parallel sum algorithm. Parallel sum requires \(\Theta(n)\)-steps and \(\Theta(\log(c))\) communications for \(c\) computing nodes, making both the parallel scalability and more specific, complexity growth, asymptotically bounded by the number of required random projection hash probes per vector (overall \(\Theta(n\log(n))\)). The data transfer requirement of the directly applied parallel sum to \textit{RPHash} is on the order of the dimensionality of the vectors and the parameter \(k\) for each compute node. To minimize this, with no overall effect on the complexity order of Parallel \textit{RPHash}, we propose a two step approach. The steps will be summarized here, as well as in standard pseudo-code algorithmic form \((\ref{phase1})\) and \((\ref{phase2})\) respectively.

Phase 1 of Parallel \textit{RPHash}: Maximum Bucket Counting.

As \textit{RPHash} is a distributed algorithm, we will assume the data vectors reside on \(D\) independent computing nodes, connected through any non-specific, Hadoop \([28]\) compatible network hardware. The system architecture will utilize the Yarn MRv2 resource manager for task scheduling and network communication. With much of the parallel task scheduling details handled by Yarn MRv2, we focus our algorithm description on the per compute node task level. As the random projections can be performed independently, the compute node interdependency is very low. Due to the Leech lattice’s set dimensionality, conversion between vectors of \(\mathbb{R}^d\) and \(\mathbb{R}^{24}\) will be performed by a random projection matrix of random variables following a nearly Gaussian distribution as in Bingham \([19]\). Therefor our first parallel task is to distribute a random projection matrix to each compute node. To further minimize communication load, we will note the fact that the random projection matrix, though, stated as random, is actually the result of a pseudo-random number generation sequence that is uniquely defined by a single seed number. A trade-off exists between the communication cost of distributing a large \(\mathbb{P}_{m\rightarrow d}\) and the redundant calculation of such data per compute node. Parallel processing tuning is an effective solution to this problem, however in \textit{RPHash} we choose the generative method, due to the constant scalability barrier that results from network bandwidth saturation and empirical results from Chowdhury \([29]\) suggesting that the ratio of communication to computation increases with the number of MR nodes. With the initial data distribution requirements taken care of, we can return to focusing on per node computation. Each compute node processes its set of vectors independently; first projecting it to a random subspace, then decoding to its nearest Leech lattice centroid ID. The lattice region IDs are generative, and are thus identical across all compute nodes. In the ‘bucket counting’ phase, we are only concerned with the number of collisions each lattice ID receives. Once all data vectors have been processed and their collisions accumulated, we will use the standard parallel sum algorithm to accumulate the lattice ID counts across all compute nodes. The set of lattice ID counts is sorted, and the \(k\log(n)\)-largest Leech lattice ID subsets are broadcast to all compute nodes for Phase 2 processing.

Phase 2 of Parallel \textit{RPHash}: The two phased approach for clustering is motivated by the work of Panigrahy, Andoni, and Indyk \([26],[27]\) in c-approximate nearest neighbor search (cANN). In the first phase, the search database of high cardinality lattice hashes is constructed/While actual searching of the database is performed in the second phase. \textit{RPHash} is a slight inversion of cANN, in the second phase, all vectors of the database are searched, against the small subset of high cardinality lattice hash ID sets. The set of lattice hash ID’s corresponding to the \(k\log(n)\) largest lattice hash ID subsets are broadcast to

**Require:** \(X = \{x_1, \ldots, x_n\}, x_k \in \mathbb{R}^m\) - data vectors
\(D\) - set of available compute nodes
\(\mathbb{H}\) - is a \(d\) dimensional LSH function
\(\bar{X} \subseteq X\) - vectors per compute node
\(\mathbb{P}_{m\rightarrow d}\) - Gaussian projection matrix
\(C_s = \{\emptyset\}\) - set of bucket collision counts

**for all** \(x_k \in \bar{X}\) **do**
\(\bar{x}_k \leftarrow \sqrt{\frac{m}{d}}\mathbb{P}^\top x_k\)
\(t = \mathbb{H}(\bar{x}_k)\)
\(C_s[t] = C_s[t] + 1\)
**end for**
\(\text{sort}(\{C_s, C_s, \text{index}\})\)
**return** \(\{C_s, C_s, \text{index}\}[0:k\log(n)]\)

Phase 1 RP-Hash Clustering
all compute nodes through standard MapReduce broadcast methods. In the case of overlapping lattice hash set, clusters having similar centroids, an overlap factor of \(\log(n)\) is applied to the parameter \(k\), so overlapping subsets can be merged during the gather phase with little effect on computational complexity.

The per compute node processing of Phase 2 requires that we perform the projection and hashing process from Phase 1 for \(\log(n)\) random projections. While processing the projection and hashes, we must also store the sums of the original vectors for each \(k\log(n)\) cluster IDs. After processing the \(\log(n)\)-projections of \(n\) vectors, we use a similar parallel sum method for vectors, to accumulate the vector components and counts. The means for each vector sum are then computed using the set of lattice ID collision totals.

Require:
\[
X = \{x_1, \ldots, x_n\}, x_k \in \mathbb{R}^m \text{- data vectors} \\
D \text{- set of available compute nodes} \\
\{C_s, C_s.\text{index}\} \text{- set of } k\log n \text{ cluster IDs and counts} \\
\mathbb{H} \text{- is a } d\text{-dimensional LSH function} \\
\tilde{X} \subseteq X \text{- vectors per compute node} \\
p_{m \rightarrow d} \in \mathbb{P} \text{- Gaussian projection matrices} \\
C = \{\emptyset\} \text{- set of centroids} \\
\text{for all } x_k \in \tilde{X} \text{ do} \\
\text{for all } p_{m \rightarrow d} \in \tilde{\mathbb{P}} \text{ do} \\
\tilde{x}_k \leftarrow \sqrt{\frac{m}{d}} p^\top x_k \\
t = \mathbb{H}(\tilde{x}_k) \\
\text{if } t \in C_s, \text{index then} \\
C[t] = C[t] + x_k \\
\text{end if} \\
\text{end for} \\
\text{end for} \\
\text{return } C
\]

Phase2 RP-Hash Clustering

**Testing on synthetic data**

Though Aim 2. will provide a thorough test of the RPHash algorithm, we will also test parallel RPHash against the standard set of parallel data clustering algorithms found in the Apache Mahout project [30]. The included clustering algorithms are: parallel K-means and fuzzy K-means, Latent Dirichlet Analysis (LDA), and parallel Mean-Shift. Testing will be performed for processing time, scalability over varying computational units, and overall cluster quality with precision-recall and ROC tests.

**AIM 2. TO PROVE THE ACCURACY AND EFFICACY OF OUR SYSTEM ON HIGH DIMENSIONAL DATA, WE WILL EMPLOY RPHASH TO THE CLUSTERING OF TEMPORAL PHYSIOLOGICAL DATA COLLECT FROM ICU PATIENTS.**

**Patients for outcome prediction.** Clustering of physiological data metrics is a useful tool in the prediction of patient outcomes which can be used to better allocate physician and material resources. Though the primary reason for Aim 2 is to demonstrate the viability of our clustering algorithm on an often difficult to cluster dataset, the results obtained may be improve upon earlier results.

Physiological data metrics obtained in ICUs are often incomplete, and not consistent between hospitals or even patients. Because of this, ICU data is often very sparse and subject to noise from human and device recording errors. Marlin *et al.* suggest a probabilistic approach using expectation maximization [1] to help mitigate some of the malformed data issues. Expectation maximization is an iterative algorithm which attempts to find the set of model parameters that best define the data distribution clusters. Our algorithm is not an EM algorithm, and thus will require some coercing in order to perform the maximization tasks. Fortunately, Carreira provides a connection between EM and Gaussian kernel Mean-Shift [31], that we can use to provide the needed conversions. Carreira’s work gives a method for converting EM algorithms into ones solved by the mean-shift algorithm under a Gaussian kernel function. The convergence rate given in Carreira for mean-shift EM and EM is linear, thus an equivalence in run-time for the two algorithms is assumed up to a constant factor difference.

Due to our system’s ability to scale well with increasing dataset sizes, we are able to utilize a much larger training set for predictability, as well as a finer event resolu-
tions. To augment the training set, we will utilize data from the MIMIC II physiological waveforms and time series database [2]. Figure 2 shows results from Marlin et al.'s work showing a general increase for the recall prediction performance metric as duration information was increased. For this reason, we believe we can improve upon the results of Marlin et al.'s work. Similar steps to Marlin et al.'s work will still have to be performed to smooth out inconsistencies in data and remove outliers, however our principle objective in AIM 2. is to achieve similar results through more data and less a priori data cleansing.

For comparison to other similar parallel systems, we propose the implementation of the EM-like, Latent Dirichlet analysis on the Marlin et al. problem. Highly optimized implementations of parallel Latent Dirichlet Analysis are readily available, and deployable on various commercial distributed computing platforms [30]. Though there are qualitative differences between the algorithms, pLDA should allow for comparison of our system with those in production.

AIM 3. APPLY TIME INVARIANT METHODS TO TEMPORAL PHYSIOLOGICAL DATA TO COUNTERACT SKEW AND LAG EFFECTS COMMON BETWEEN SIMILAR PATIENT'S DATA SEQUENCES. Direct application of unsupervised clustering methods often requires data vectors to be independent discrete points contained in the same same subspace embedding. A difficulty with temporal patterns is that these restrictions are rarely observed. Patient event metrics, are one such example of temporal data. Patient event metrics are time and sequence dependent, that are often misaligned, and rarely occupying the same subspace. In full sequences, misalignment can be mitigated by aligning sequences to their minimum distance alignments with other vectors. This process is however computationally expensive, and works poorly on incomplete or missing data, which is common among patients admitted to ICUs. Marlin et al. cite this as a shortcoming in their analysis [1], while also providing a rudimentary solution by way of piecewise averaging. For short durations with small numbers of samples, piecewise averaging performs well [32] but often fail to capture all available data. In this proposal, we are looking at much longer durations, with higher resolution and more samples per sequence in which piecewise averaging is no longer a viable technique for clustering of time-series data.

Liao et al. states that clustering algorithms have predominantly dealt with static data due to the difficulty of encapsulating the interdependence between elements of a time sequences [33]. Various methods exist for converting temporal data into static data however such as the Discrete Fourier (DFT/FFT) and Wavelet transforms (DWT) [34] [35], as well as Autoregressive (AR) Coefficient Feature transforms, Hidden Markov Models(HMM) [36]. In addition a host of hybrid kernels combining a variety of sequence summery metrics have been proposed [37]. Many approaches attempt to find features on which to apply clustering that are applicable across a broad spectrum of time based sequences. In Aim 3. we propose an approached for physiological data that is tailored to its use in practice. The principle insight to this approach results from human physiological metrics having well known and established distributions. Wherein the means or average values of physiological metrics tend to imply that the correlated biological system is functioning properly, while outliers suggest something is wrong. The idea of outliers detection is of course not new [38], but its application to patient outcome prediction from physiological data has been overlooked.

Though many solutions have been explored for clustering time invariant signals [39], [40], [41], [37]. However, we feel the biggest problem with patient data is not global sequence interdependence, but rather only skew and outliers are important. Skew is considered in Sung [42] where signal shape transformation functions are used as feature vectors for a given signal such as the fourier transform and discrete wavelet transforms. In addition, Sung also uses the Earth Mover Distance (EMD), a distance
measure similar to edit distance, to compare signals. Though we plan to apply the above signal shape transformation to our signal feature vector clustering of physiological metrics, we suggest our own feature extraction method that naturally arises from the purpose of most medical recording devices.

In Aim 3, we propose a simple addition to standard time invariant data transforms, called top k sorted outliers transform. The transform is straightforward and consists of ranking physiological metric outliers’ distances from the patient’s and other accepted and demographically relevant averages. In addition to the outliers feature vectors, we will also incorporate Fourier, Wavelet, and AR coefficients to the feature vector. Similar to the research found in Yang [37], we will incorporated a weighted score to each feature set, that maximizes the covariance between the features and known training data outcomes. The steps of this algorithm consist of the following 3 preprocessing steps. First all data sequences are converted into their feature equivalent vectors for the 4 transforms. Then sequence vectors are cluster by their known patient outcomes. Weighting of the 4 transformed metrics is given by the loading matrix computed by the PLS regression algorithm as in Korn [43]. The weighted feature vectors, are then clustered directly by the RPHash algorithm. Patient prediction consists of feature vector conversion, then application of the weighting vectors. The nearest cluster to the weighted feature vector and its member’s outcomes are the set of likely outcomes for the given sequence. Evaluation of Aim 3. will be performed against the results of Aim 2. as well as the stated results of Marlin et al. Evaluation will be performed on a random 80-20 training set metric. In addition to our exploratory investigation in the improvement of patient outcome predictability, Aim 3. will also serve to demonstrate the data robustness of the system developed in AIM 1.

AIM 4. DEMONSTRATE THE DATA SECURITY OF OUR SYSTEM OVER DISTRIBUTED DATA RESOURCES. Recent United States government initiatives pushing for the large scale availability of data resources have made vast quantities of de-identified health information available to the public, however recent advances in attacks on de-identification of whole genome sequence data have been used to to associate anonymized medical records with specific individuals [8]. Similar de-anonymization attacks [7] [44] along with a presidential commission (privacy and progress in WGS) have prompted the need for better data security for medical records data. Our algorithm provides an intrinsic solution to this problem in both the distribution of data among servers as well as during the communication steps required by the algorithm.

RPHash is a distributed algorithm design to compute across internet spans. As such, data it uses to compute clusters can theoretically reside independently among different health care facilities with no requirement for any single location data storage architecture.

As a consequence of projecting the real data vectors to a random subspace via a near, but not completely orthogonal matrix, destructive data loss occurs providing a cryptographic ‘trap-door’ function. The data loss is an intrinsic part of the RPHash clustering algorithm that has no adverse effect on its function.

The only point at which fully qualified vectors are transmitted between compute nodes is during the Phase2 gather stage. Though the data represents a set of individuals, by definition of a centroid it is the average of only the k largest populations of patients. RPHash is a clustering algorithm for very large datasets, where the probability of identifying an individual from Phase2 vector information is proportional to the size of the smallest identified cluster.

TIMELINE.

Aim 1. will constitute the majority of our efforts. As it is the algorithm behind the tests in our subsequent aims, it must be completed and tested prior to use. The testing phase against Mahout clustering algorithms however need not be performed at this time. As the development and deployment on commercial cloud computing resources is by in large the longest duration task, we are allotting 6-8 months to complete it.

Aim 2. will be the shortest of sub-tasks and should only take 1-1.5 months to implement in its most
Aim 3. will be completed at the end of the proposal period. The development of new time invariant kernels and feature extraction methods for evaluating mortality rates in ICU patients is not essential to the other aims, and instead simply provides solutions to some shortcomings of other works, as well as another test of the algorithm defined in Aim 1. Aim 4. Is an ongoing demonstration of the security tolerance of the system from Aim 1. and used during Aim 2 and Aim 3.

References


REFERENCES


REFERENCES


