*The two referees disagree on the overall tone; the second referee likes the paper style but the first doesn’t. We keep the basic style but add details to answer second referee*

Review 1 - Comment to Author:

The paper appears to piece together several ideas without a clear objective.  It is not well written – sentence structures are often sloppy and mathematical terms are not well-defined.  
First, the abstract states that  
i)               the paper presents a “suite of robust parallel algorithms running on both clouds and HPC systems”;  
ii)              the paper applies the suite to varying data;  
iii)            the paper “introduces improvements to known algorithms”;  
iv)             the paper discusses visualization for high-dimensional data;  
v)              the paper “exploits” deterministic annealing;  
vi)             the paper applies the methods to “several life sciences applications”;  
  
It attempts to do these, but in a very cursory way.  It would be better if the author picked just one topic and did it well. *Given remark at start we left style of paper unchanged*  
Section 2. Deterministic Annealing:  
            Just before equation (1), the author states that is introducing the Gibbs distribution at temperature T.  It is not clear if this is a standard technique or one of the improvements mentioned in the abstract. *This concept is not new and I explicitly comment on new ideas in paper and add*

*The formulation above shows deterministic annealing to be generally applicable and gives a method first described by Hofmann and Buhmann for the general problem [5] as a generalization of the original theory derived by Rose, Fox and collaborators [6, 7]. The discussion here includes published [4, 8] and unpublished improvements, and the latter are noted explicitly. Further I do not think the many published ideas have ever been integrated as efficient parallel functional libraries and that is another important contribution of this paper.*

            In the last paragraph of Section 2, the author discusses the difficulty in formulating the objective function (Hamiltonian) and in solving the algebra (higher computational complexity).  As a conclusion, he suggests that libraries should be build that “embody these more sophisticated algorithms”.  Does he mean the annealing methods or the K-means methods. *I clarified last sentence as*

*The resultant software complexity is at least an order of magnitude greater than simpler methods such as K means which reinforces the suggestion that one should build libraries that once and for all embody these most sophisticated powerful algorithms such as deterministic annealing.*  
Section 3.1. Basic Central or Metric Space Clustering  
            In this section, there inconsistent use of “x” and “i", causing the equations to be confusing, at best, and verging on inaccurate. *Corrected*  
    
Section 3.2  Fuzzy Clustering and K-means  
            In this section, the author is continuing to use the analogy to deterministic annealing by describing high temperature and lowering the temperature.  It is not clear how this works.  Perhaps, the author is alluding to the number of clusters (e.g., starting with a single cluster and increasing the number of clusters).  The major issue with fuzzy clustering and k-means is determining i) if clusters exist in data, and if so ii) how many exist.  The algorithms will always partition the data into the specified number of clusters.  How are the outcome deemed acceptable?  The author does not address these issues.

*I added*

*In deterministic annealing, one can stop in two different ways; one criterion is just reaching a particular cluster count as is essentially used in traditional K means. However the results of figure 3 come from a different approach stopping splitting at a particular value of temperature i.e. at a cluster size in target space. This emphasizes the value of concept that √T is related to distance in parameter space. In this application the cluster size is known as it corresponds to a measurement error that can be determined a priori.*  
Section 3.3 Multiscale (Hierarchical) and Continuous Clustering  
            This section appears to describe a contribution with the eigenvalue method, especially since it describes a way to divide large cluster into smaller clusters. *I added comment on contribution*

*It was introduced in metric case by Rose [4] without details and our extensions for the non-metric case and Sponge case are new.*  
            Again, the author is using the analogy to the deterministic annealing.  Does temperature have a physical meaning in the example?

*At the start of section 3.3 it explains*

*In many problems, decreasing temperature is a classic multiscale step with finer resolution being used as temperature T decreases. Note from equations (9) and (10) that we have factors like (X(x) - Y(k))2 / T and √T acts as a distance scale.*  
            Figure 2. is confusing. *I added explanation in text around this figure.*  
   
Section 4. Parallel Clustering  
            This section finally addresses issues with clustering on big data.  The author presents two basis approaches:  i) divide data evenly among the number of processors and each processor has a copy of all cluster centers, and ii) each processor would have a cluster center and the points closest to that center.  It is not clear if this second approach is the author’s own work.  If it is, that needs to be stated more clearly. *It is my work and I added text pointing this out*

Review 2 - Comment to Author:

This paper gives a clear and cogent description of principles underlying an important class of algorithms used to cluster large high dimensional datasets, describes approaches used to parallelize these algorithms along with methods to represent salient information in two or three dimensions to allow for visualization. The manuscript presents performance and clustering results from biomedical applications.   The paper is very well written and presents a very principled approach to clustering methods and to their parallelization.

*Thank you!*