Clustering fMRI Time Series

Boon Thye Thomas Yeo ythomas@mit.edu   Wanmei Ou wanmei@mit.edu

December 2, 2004

Abstract

Clustering fMRI time series has emerged in recent years as a possible alternative to parametric modelling approaches [2]. In any clustering problem, a good solution depends on two components: the choice of the clustering metric and the clustering algorithm itself. In this paper, we consider a modified version of a common fMRI clustering metric obtained by the cross-correlation of the fMRI signal with the experimental protocol signal. To address a perceived deficiency of this signal-to-protocol metric, we devise a signal-to-signal metric by modifying the cross-correlation of two fMRI signals. In addition, we apply three different clustering algorithms (kmeans, spectral clustering and stochastic clustering) to the problem at hand. We shall demonstrate the superiority of our novel metric when applied to real fMRI data.

1 Introduction

Functional Magnetic Resonance Imaging (fMRI) based on Blood Oxygenation Level Dependent (BOLD) signal changes allows assessment of brain activity via local hemodynamic variations over time ([3],[9]). However, traditional methods, such as the 2-sample t-test or the general linear model (GLM) can only at most evaluate the level of activation of individual voxels[1].

Analyzing possible relations between different active voxels has the potential of elucidating biological relationships between the different functional areas of the brain and therefore motivates our investigation of fMRI clustering. By defining a similarity measure between different time series, we hope to group voxels with similar time series. The underlying assumption is that voxels with similar time series correspond to cells belonging to the same functional regions of the brain. We can then analyze inter- or intra- group relations after successful clustering.

In this paper, we modify a common metric used in fMRI clustering to improve the chance of physically-close voxels to be grouped together. This is justified by the biological fact that the brain is divided into active regions. Therefore physically-close voxels most likely correspond to cells within the same functional regions - and given our motivation in the previous paragraph - we would like to group them together. However, we have to be careful not to set too high a bias, or we risk voxels being grouped together simply because of their physical distance. In addition, we will define a new metric that addresses another perceived problem of the common metric (see 2.2) that is related to our intuition of similarity between time series. We shall also evaluate three clustering algorithms, Kmeans (standard and adapted), spectral clustering (2^n and K) and stochastic clustering on artificially generated fMRI signals.

2 Metric

A clustering algorithm can only be as good as the similarity metric used. The problem at hand is to meaningfully quantify the similarity between two fMRI time series. The first similarity metric we shall modify is the one commonly used by the fMRI clustering community. It is only “semi-model” because it makes some prior assumptions about the fMRI signals.

Before we define the similarity between two time series, we first have to do the following pre-processing:

1. We first apply a second-order low pass filter on the time series to improve its SNR. Typical fMRI SNR ranges from 0.2 to less than 1. Actual SNR depends on the fMRI machine, imaging procedures as well as voxel resolution.

2. We standardize each filtered time course by subtracting the mean and dividing it by its standard deviation. In the absence of standardization, cross-correlation between two signals could be dominated by similar signal variances rather than similar patterns [1].

More sophisticated pre-processing can be performed ([1], [2], [5], [7], [8]), but we will not indulge in them since they are not our focus.

[1] Voxel is the basic unit of composition of a 3D image.

2.1 Semi-model Based Metric

A commonly used semi-model based metric is the signal-to-protocol metric [2]. The time series of a voxel, $f[n]$ is cross-correlated with the experimental protocol $s[n]$ resulting in $z[n]$. $s[n]$ is a square wave which takes value of zero when the subject is resting and value of one when the subject is presented a stimulus. Hence $z[n] = \sum_m f[m]s[n+m]$. The peak value, P and its delay, L of $z[n]$ are taken as the features of the time series. The peak difference, P_{ij} and delay difference, L_{ij} of two voxels, i and j, are defined as $P_{ij} = |P_i - P_j|$ and $L_{ij} = |L_i - L_j|$. Unlike Goutte [2], we will include the euclidean distance between two voxels, E_{ij} in the similarity metric. Utilising all possible pairs of voxels, we then standardize the three measures, P_{ij}, L_{ij} and E_{ij} separately, resulting in $\hat{P}_{ij}$, $\hat{L}_{ij}$ and $\hat{E}_{ij}$. Note that after standardization, certain $\hat{P}_{ij}$, $\hat{L}_{ij}$ and $\hat{E}_{ij}$ will become negative. The distance and affinity between the time series of two voxels, are therefore respectively defined as:

$$D(i,j) = \hat{P}_{ij} + \hat{L}_{ij} + c\hat{E}_{ij} \;\; ; \;\; A(i,j) = e^{-D(i,j)} \tag{1}$$

where c is a positive constant. Including the euclidean distance in the similarity metric increases the chance of voxels in similar regions of the brain to be grouped together. If c is too big, we risk voxels being grouped together simply because they are physically close. From our experiments, we find that $c = 0.5$ provides the satisfactory trade-off.

2.2 Data-driven Metric

Consider three time series $f_1[n]$, $f_2[n] = -f_1[n]$ and $f_3[n] = 0$. Then according to the signal-to-protocol metric, $f_1[n]$ and $f_2[n]$ is more similar to $f_3[n]$ than with each other (if we ignore the delay), because $P_{12} = |P_1 - P_2| = 2|P_1|$ while $P_{13} = P_{23} = |P_1|$. This is counter-intuitive, since a positive signal should be considered more similar to a negative signal than to no signal. We are also uncomfortable with the use of the protocol, $s[n]$ because that is implicitly assuming the temporal extent of the hemodynamic response is at most the length of the stimulus. We therefore define our signal-to-signal metric:

1. Consider two time series $f_i[n]$ and $f_j[n]$. Then $z_{ij}[n] = \sum_m f_i[m]f_j[n+m]$.
2. Define $z'_{ij}[n]$, such that $z'_{ij}[n] = z_{ij}[n]$ when $z_{ij}[n] \geq 0$ and $z'_{ij}[n] = -k \times z_{ij}[n]$ when $z_{ij}[n] < 0$, where k is a positive constant, which we set to be 0.5 in our implementation
3. Hence, $P_{ij} = max(z'_{ij}[n])$ and $L_{ij} = arg\ max_n z'_{ij}[n]$. After standardization, we have:

$$D(i,j) = -\hat{P}_{ij} + \hat{L}_{ij} + c\hat{E}_{ij} \;\; ; \;\; A(i,j) = e^{-D(i,j)} \tag{2}$$

   where there is a minus sign in front of $\hat{P}_{ij}$ because in this case, the higher the value of $\hat{P}_{ij}$, the more similar voxels i and j are. Note that there is an intuitive meaning behind the value of k. By setting k to be 0.5, we are saying that we believe a negative peak is as good as a positive peak half its size.

3 Clustering Algorithms

In this section, we describe the clustering algorithms we investigate: Kmeans (normal and adapted), Spectral Clustering (2^n and K) and Stochastic Clustering.

3.1 Adapted Kmeans

K-means is an iterative method for locating clusters in a set of unlabeled data while minimizing the total within cluster variance. It is known that Kmeans approximates the Expected Maximization (EM) of a mixture of gaussians when the covariance matrices of the gaussians is close to a scaled constant of the identity matrix and the diagonals approach zero. At each iteration, a point is assigned to the cluster with the closest cluster center. We then update the center of each newly defined cluster by averaging the features of the points belonging to the cluster.

Unfortunately, our data-driven metric (see eq. 2) only provides us with a pairwise distance matrix, $D(i,j)$. There is no notion of the actual coordinates of each point. Setting the cluster center to be the average time series of the cluster voxels is wrong since the averaged time series might no longer belong to the space of fMRI signals. Therefore, in this case, Kmeans cannot be applied directly. To overcome this problem, we define the distance of a point, P_i to a cluster, C_x as the following:

$$D(P_i, C_x) = \frac{1}{|C_x|} \sum_{j \in C_x} D(i,j) \tag{3}$$

where $|C_x|$ is the cardinality of C_x from the last iteration. Like regular Kmeans, adapted Kmeans is initialized with K random centers, and the remaining points are assigned to the “closest” cluster centers. For subsequent iterations, data points are assigned to the nearest clusters according to (3).

3.2 Spectral Clustering Families

In spectral clustering, clustering is viewed as a graph partition problem. The graph nodes correspond to the data points, while the weights of the edges reflect similarity between the nodes. The NxN affinity matrix, A therefore completely characterises a N-nodes graph where A_{ij} is the weight (or affinity) between node i and node j.

Spectral clustering algorithms are based on the eigen-decomposition of the normalized affinity matrix, $N_A = Q^{-1/2}AQ^{-1/2}$ (Q is a diagonal matrix, $Q_{ii} = \sum_j A_{ij}$) and rely on the assumption that the subspace spanned by N_A's eigenvectors is more stable than that spanned by N_A's columns. Weiss [6] proved that the eigenvectors of N_A corresponding to the k largest eigenvalues are piecewise linear if the data giving rise to N_A comes from k well-separated clusters. However, if the clusters overlap or are close to one another, the subspace spanned by the first k eigenvectors become less stable according to matrix pertubation theory [11].

3.2.1 2^n Spectral Clustering

In 2^n spectral clustering, we recursively divide the cluster with the highest intra-cluster variance (defined to be the average intra-cluster weights) into two by performing an elementary spectral cluster split. We stop when the desired number of clusters is achieved. Each elementary spectral cluster split is performed by applying Kmeans (with 2 clusters) on the eigenvector corresponding to the second highest eigenvalue of the cluster's normalized affinity matrix, N_A.

2^n spectral clustering is effectively equivalent to Shi's recursive 2-way normalized cuts [10] as shown by Weiss [12]. But as Shi pointed out himself [10], recursive 2-way cuts are susceptible to prematurely breaking up a "should-be-coherent" cluster into two. He suggested using the recursive normalized cuts to oversegment the data into $k' > k$ clusters and then either greedily merge the k' clusters into k clusters, or treat the k' clusters as the meta-nodes of a new graph and repeat recursive normalized cuts. Note that in our implementation of 2^n spectral clustering, we stop the iterations once k clusters are reached. In light of this problem, a potentially better way of producing k clusters is K Spectral Clustering [6].

3.2.2 K Spectral Clustering

While 2^n spectral clustering recursively divides data into two groups based on the second eigenvector, K spectral clustering [6] simultaneously separates data into K groups using the eigenvectors corresponding to the K largest eigenvalues. We outline the algorithm below:

1. Construct $X = [x_1, x_2, ...x_k]$, x_i is the eigenvector corresponding to the i^{th} largest eigenvalue.
2. Normalize each row of X, resulting in Y, where $Y_{ij} = X_{ij}/(\sum_j X_{ij})^{1/2}$.
3. Treating each row of Y as a point in R^k, cluster them into k clusters via Kmeans.
4. Assign original point, P_i to cluster j if and only if row i of Y is assigned to cluster j.

Weiss [6] provides another perspective about K spectral clustering. He shows that if the data comes from k well-separated clusters, then there will be k mutually orthogonal points on the surface of the unit k-sphere around which Y's rows will cluster. Furthermore, these clusters correspond exactly to the true clustering of the original data. Unfortunately, the accuracy of K spectral clustering can suffer from its dependence on the initial random cluster centers of Kmeans (step 3).

3.3 Stochastic Clustering

We shall consider the stochastic clustering algorithm presented by Gdalyahu [14]. Like spectral clustering, stochastic clustering views clustering as a graph partition problem. If we consider the graph-partitioning problem as a search through the hypothesis space of feasible partitions, most clustering algorithms, including spectral clustering, simply returns a point from the hypothesis space. On the other hand, stochastic clustering attempts to induce a probability distribution over the hypothesis space and returns an average solution [14]. This is achieved by generating randomized-cuts in the graph and computing an "average" cut. Therefore, unlike most deterministic graph partitioning algorithms, it is more robust against accidental edges and small spurious outliers. A second advantage is that a foreknowledge of the number of clusters is not required. Instead, stochastic clustering returns a nested sequence of partitions, $G^r(V, E)$, whose significance is correlated with a defined measure, $\delta T(r)$. This is attractive because the usual aim of clustering is to discover hidden data structures using as little prior assumptions as possible. These two advantages of stochastic clustering will be illustrated in the synthetic fMRI data section (see section 4.2)

3.3.1 Algorithm Outline

Consider the graph $G(V, E)$. A partition of V into r disjoint clusters (V_1,...,V_r) is called a r-way cut. By combining all edges between two clusters (via adding their weights) such that there is at most 1 meta-edge between any two clusters, we can imagine the nodes of each cluster as a single meta-node. Hence, we can obtain a (r-1)-way cut from a r-way cut simply by $contracting$ a meta-edge between two meta-nodes and combining their external meta-edges.

We define a contraction sequence as taking the original graph, $G(V, E)$ (which is a $|V|$-way cut) and at each iteration, contract the graph by combining 2 meta-nodes, until a single massive cluster ($r = 1$) remains. The probability of combining two meta-nodes is directly proportional to the edge between them. We repeat this contraction sequence M times. We estimate p_{ij}^r, the probability of node i and node j being in the same cluster in a random r-way cut, to be the percentage of times out of M contraction sequences, that node i and node j are in the same cluster in a r-way cut. For clarity, we list the steps below.

1. Create a 3-D matrix S_{ij}^r, initializing the components to zero.
2. We perform the graph contraction sequence M times, each time contracting the graph $G(V, E)$ ($|V|$-way cut) until only one meta-node remains (1-way cut).
3. At each stage of the contraction, we take the r-way cut and update the S_{ij}^r matrix, setting $S_{ij}^r \leftarrow S_{ij}^r + 1$ if node i and node j belong to the same meta-node.
4. We then estimate p_{ij}^r to be S_{ij}^r/M.

It can be shown that if $M > 0.35\frac{\log_2|E| - log_2\delta+1}{\varepsilon^2}$, then $Pr(\exists i, j\ s.t.\ |\hat{p}_{ij} - p_{ij}| > \epsilon) < \delta$ [14]. In our implementation, we set both ϵ and δ to be 0.05. After estimating p_{ij}^r, we can now generate the nested sequence of partitions:

1. For each value of r, we generate a new graph $G^r(V, E)$ from $G(V, E)$ by removing edges between nodes i and j if $p_{ij}^r < 0.5$.
2. For each reduced graph, $G^r(V, E)$, we find its connected components (or clusters), (A_1,...,A_s). Note that the number of connected components, $s \leq r$ (from experience $s << r$) but increases monotonically with r. This is because p_{ij}^r decreases monotonically as r increases, hence $G^r(V, E)$ always has at least as many edges as $G^{r+1}(V, E)$.
3. For each r, we calculate $\delta T(r) = T(G^r(V, E)) - T(G^{r-1}(V, E))$, where

$$T(G^r(V, E)) = \frac{2}{N(N-1)}\sum_{i>j} N_i N_j \tag{4}$$

   $N_k = |A_k|$ denoting the number of nodes in the kth cluster. Hence, $T(G^r(V, E))$ measures the number of inter-cluster edges in the r-partition relative to the total number of edges in the complete graph.

4. Typically, the partitions $G^r(V, E)$ corresponding to consecutive values of r are very similar, and therefore $T(G^r(V, E))$ tends to remain relatively stable as r changes. However, abrupt changes in $T(G^r(V, E))$ do occur and these tend to correspond to significant changes in the structure of the graph partition. Hence we output the graph partition only when $\delta T(r) > threshold$. The algorithm is not very sensitive to the exact value of the threshold (see figures 4b and 7), and in our implementation, we arbitarily set the $threshold$ to be 0.01.

3.3.2 Implementation Issues

It turns out that implementing the stochastic clustering algorithm is not trivial, because of both space and runtime issues. Since we encountered many implementation problems, we shall include a short discussion here.

Consider the matrix S_{ij}^r. Suppose we are interested in clustering $N = 1000$ points, the space for storing the full S_{ij}^r matrix would be $10^9 \times sizeof(int)$ bytes of memory! We therefore make the assumption [14] that the average cut is not sensitive to the p_{ij}^r of an edge whose original weight, w_{ij} is zero. Note that even though w_{ij} is zero, p_{ij}^r will be non-zeros for some r's (assuming the random walk associated with the graph is ergodic) due to transitive relations via other nodes. The justification for the assumption is that since p_{ij}^r is completely due to transitive relations, we can rely on these very same transitive relations in the formation of connected components in the second stage of the algorithm. With this assumption, the space requirements for S_{ij}^r would drop drastically from $N^3 \times sizeof(int)$ bytes to $\sharp$ non-zero edges $\times\ sizeof(int)$ bytes.

From here, we depart from Gdalyahu's implementation. While he further approximated p_{ij}^r using histograms and speeded up the algorithm using trees, we implemented the complete version using matlab. Even with the heavy reduction of the size of S_{ij}^r, stochastic clustering is still really slow. Clustering 400 points (see section 4.1) takes more than 12 hours. A significant portion of the runtime is concentrated in the sampling stage of the algorithm. Because of the repeated sampling and contraction, there are for loops that cannot be avoided in matlab. For future work, implementing stochastic clustering in C++ would definitely be more rewarding.

4 Results

4.1 The Windmill of Gaussians

We test our clustering algorithms on artificially generated 2-dimensional mixture of 4 gaussians. All the algorithms are reasonably successful. As we pointed out in the previous section, the performance of K spectral clustering can be sensitive to the random initialization of Kmeans (fig. 1e). Note that 2^n spectral clustering can also become slightly unstable due to its use of Kmeans in the elementary spectral split. However, since the second eigenvector consists of 1-dimensional data points, we can easily use a deterministic algorithm to split the eigenvector into two instead of using Kmeans. For example, we could sort the eigenvector, and find the pivot element that minimizes the normalized-cut criterion. The stochastic clustering algorithm generates a series of partitions (fig. 1f), of which one is shown here. Notice how outliers are relabeled as separate clusters.

4.2 Result on Synthetic fMRI data

We generate synthetic fMRI data using hemodynamic response functions provided by ([4]). We create simulated hemodynamic signals for selected voxels in a 64×64 image. Each signal consists of 108 time points with "on" and "off" periods corresponding to that of a real fMRI experiment (see section 4.3). We then add gaussian noise to all 64×64 voxels. However, it appears from the affinity matrices of the synthetic fMRI data that the signal-to-protocol metric is better at handling lower SNR (note that this is not true with real fMRI data - see section 4.3). Hence depending on our metric, we generate data with different SNR.

4.2.1 Evaluating Adapted Kmeans and Spectral Clustering using Signal-to-Signal Metric

Consider figure 2a. The hemodynamic response within each colored block is constant, with SNR $= 1$. The signals from the blue pixels are similar to those from the yellow pixels while the signals from the red and cyan blocks are similar (fig. 2d). We select the 800 activated voxels and an extra 100 unactivated voxels (the brown pixels) for clustering. We see that adapted Kmeans is unable to separate the diagonal blocks even with the euclidean constraints (fig. 2c). Both types of spectral clustering are unable to separate the diagonal blocks without the euclidean constraint (fig. 2b), but are successful when the constraint is imposed (fig. 2e, only results from 2^n spectral clustering shown).

4.2.2 Evaluating Normal Kmeans and Spectral Clustering using Signal-to-Protocol Metric

We set SNR $= 0.3$ (fig. 3a, c). Like adapted Kmeans, normal Kmeans is unable to separate the diagonal blocks even with the euclidean constraints (figures not shown). This time, both 2^n and K spectral clustering are able to separate the diagonal blocks even without the euclidean constraints (fig 3b, d, only results from 2^n spectral clustering shown). However, we observe that without the euclidean constraint, there is some corruption across the diagonals, with some brown pixels labeled as yellow (and vice versa), as well as some cyan pixels labeled as red.

4.2.3 Stochastic Clustering

We tested stochastic clustering using the signal-to-signal metric, but with a smaller data set (300 activated and 40 unactivated pixels) to accommodate its slower runtime. Consider figure 4a. Like before, the hemodynamic response within each colored block is constant with SNR $= 2$. The SNR is set higher this time because we decide to use very similar signals in all three blocks. Without this higher SNR, the affinity matrix, A, shows that the blocks are indistinguishable.

As promised, stochastic clustering generates a series of partitions, of which the two corresponding to the largest values of $\delta T(r)$ are displayed (fig. 4c, d). Just before the first major $\delta T(r)$, all the activated pixels are grouped into a single cluster, with most of the unactivated pixels separated from this main group. At the first major peak, the top two clusters are separated from the bottom one, and at the second peak, the top two clusters are separated.

For visualization purposes, clusters that contain only 1 pixels are grouped together and displayed as a single color. Notice that these usually correspond to unactivated pixels (c.f. fig. 4a and d). Hence, stochastic clustering is clever at ignoring outliers, rather than forcefully clustering them. In practice, we could take these single pixels and either discard them or try to incorporate them back into the main clusters depending on our objective.

4.3 Result on real fMRI data

The fMRI scans were obtained during an auditory "two-back" word experiment. Each experiment consisted of five rest epochs and four task epochs, each epoch lasting 30 seconds. In the rest condition, the subjects were instructed to concentrate on the noise of the scanner and not move. In the task condition, the subjects were presented with a series of pre-recorded single-digit numbers, three seconds per number. The subjects were asked to tap their index finger on the thumb when hearing a number that was the same as the one spoken two numbers before. Experiments were repeated ten times for each subject.

The anatomical images, T1-weighted structural, were acquired for each subject on a 1.5 Tesla GE signa clinical MR scanner using a 3D-Spoiled Gradient Recalled sequence. EPI images were acquired on the same scanner with imaging parameters as follows: axial orientation, TR/TE=2500/50 msec, FA90, 64 by 64 acquisition matrix, 6mm slice-thickness, no gap, 24 slices. More details on this study can be found in [13]. In this section, we illustrate the detectors' performance on one of the subjects from the study [2]. For our clustering experiments, we pick the top 300 voxels using a two-sided t-test from slices 18 to 22 as input into our clustering algorithms. 300 voxels correspond to about 5% of the brain voxels in the 5 slices, meaning that probably a percentage of the 300 voxels picked are false positives.

4.3.1 Semi-model metric VS Data-driven metric

Although the synthetic fMRI data section (see 4.2) seems to imply that the signal-to-protocol metric is more resilient than the signal-to-signal metric in the presence of noise, this is definitely not true when we use real fMRI data.

As can be seen from the plots of the affinity matrix A of the signal-to-protocol metric (fig. 5a, b), the signal-to-protocol metric seems unable to discover any similarity between the time series of different voxels (with and without the inclusion of the euclidean distance in the similarity metric). The plots are almost completely blue (a deeper red corresponds to a higher affinity, while deeper blue corresponds to lower affinity). In fact, when we run the clustering algorithms using the signal-to-protocol metric, the clustering results were bad with many small spurious clusters whose elements are spatially scattered, even when the euclidean constraint was imposed (figures not shown). On the other hand, we can see the signal-to-signal metric (fig. 5c, d) is successful in revealing structures among the voxels. The imposition of the euclidean constraint reduces the similarity between voxels that are physically far away.

The reason is because the signal-to-protocol metric assumes the hemodynamic response is instantaneous with respect to the stimuli and the temporal extent of the response is at most the length of the stimulus. When that assumption is true (such as in the synethetic fMRI data), the signal-to-protocol metric performs better than the signal-to-signal metric, because this *correct* prior knowledge helps to overcome poor SNR. In the real human neuro-biological systems however, it is evident that the assumption is not wholly accurate, and we see that our signal-to-signal metric is superior.

4.3.2 Clustering Results

Because of Kmeans' abysmal results with synthetic fMRI data and its ubiquity in fMRI clustering literature, we shall exclude it from our experiments with real fMRI data. For brevity, we will only present in details the results from stochastic clustering (see figures 6 and 7). Note that each sequence of slices should be seen as a 3-d block of the brain. The five slices should be imagined as stacking on top of each other.

Figure 6a shows clustering results using the signal-to-signal metric without the euclidean constraint, corresponding to the maximal peak of $\delta T(r)$. As expected, at the peak, stochastic clustering gives us a single big cluster of voxels (colored purple in the figure). As r increases (not shown), the voxels eventually break up into different groups but the number of spurious pixels (clusters with single member) also increases significantly. Hence, we find that without euclidean constraints, stochastic clustering is unable to convincingly separate the different activated voxels into coherent groups. This is not surprising since a clustering algorithm is only as good as the metric used. From figure 5c, the large patches of orange show that most of the voxels are very similar to each other according to the clustering metric we use. We therefore should not expect a reasonable clustering algorithm to split them up.

Figure 6b, c show clustering results with euclidean constraints, corresponding to the two highest peaks of $\delta T(r)$. We see that at the first peak (fig. 6b, 7), the algorithm manages to separate a coherent group of voxels (shown as green) from the rest of the voxels (shown as purple). At the highest peak (fig. 6c, 7), we see that the purple cluster in turn splits up into two groups (shown as cyan and purple). Note that at higher r (not shown here), the cyan cluster breaks up into two: one cluster at the center of the brain and one at the side. However, more spurious clusters (containing single member) are also produced.

We also test both types of spectral clustering with the real fMRI data (figures not shown) using 10 clusters. The results are similar to stochastic clustering except that there are less spurious clusters (single-point clusters). However, we actually see this as a major advantage of stochastic clustering: it gracefully eliminates outliers rather than forcefully group them with otherwise coherent clusters. This prevents the outliers from altering the mean properties of these coherent clusters (refer to comments on stochastic clustering on synthetic fMRI data in section 4.2).

5 Conclusions and Further Work

Our investigation of various clustering methods and metrics facilitates the study of relatonships between different functional areas of the brain. While 2^n spectral clustering and K spectral clustering produce similar results to stochastic clustering, stochastic clustering is more appropriate for clustering fMRI time series because it gracefully excludes spurious non-active voxels which occur due to inaccuracies in the t-test. Since there is no ground truth about the actual number of functional

[2] The original study contains nine subjects, but for the purposes of this paper, we decide to present the results for one subject across all detectors

clusters in the brain, stochastic clustering relieves us from having to specify an arbitrarily-determined number of clusters which might lead to overfitting. Kmeans do not perform very well with our tests and should therefore be avoided. Our novel signal-to-signal metric outperforms the common signal-to-protocol metric because it does not assume instantaneous hemodynamic response, which is unrealistic in neurobiological systems. Potential future work might include:

- The study of intra- and inter-cluster relationships between different functional regions of the brain using graphical models, after clustering the voxels.
- Further evaluation of spectral and stochastic clustering methods in the presence of more inactive noise voxels.
- The handling of spurious clusters containing single point in stochastic clustering.
- Improvement of the metric, $\delta T(r)$ in stochastic clustering. Currently $\delta T(r)$ only takes into account the number of edges, but not their weights. Perhaps $\delta T(r)$ can be altered to be similar to the normalized cuts criterion.
- Improvement of the clustering metric by the incorporation of standard detection statistics such as t-test and GLM.

References

[1] Christine Baudelet and Bernard Gallez. Cluster analysis of bold fmri time series in tumors to study the heterogeneity of hemodynamic response to treatment. *Magnetic Resonance in Medicine*, pages 49:985–990, 2003.

[2] C. Sarver E. Rostrup L.K. Hansen Cyril Goutte, F.A. Nielsen. Space-time analysis of fmri by feature space clustering. *NeuroImage*, pages 7:4, part 2, S610, 1998.

[3] Christian Windischberger Kurt Hornik Evgenia Dimitriadou, Markus Barth and Ewald Moser. Quantitative comparison of functional mri cluster analysis. *preprint submitted Elsevier Science*, 2003.

[4] Feter Jezzard, Paul M. Matthews, and Stephen M. Smith. *Functional MIR – An Introduction to Methods*. OXFORD, 2002.

[5] Jason Tregellas Robert Freedman Jody Tanabe, David Miller and Francois G. Meyer. Comparison of detrending methods for optimal fmri preprocessing. *NeuroImage*, pages 902–907, 2002.

[6] Jordan M.I. Ng A.Y. and Weiss Y. On spectral clustering: Analysis and an algorithm. *NIPS*, 2001.

[7] Peter Lundberg Ola Friman, Magnus Borga and Hans Knutsson. Detection and detrending in fmri data analysis. *Manuscript, Linkoping University, Sweden*.

[8] Daniel B. Rowe and Steven W. Morgan. Computing fmri activations: Coefficients and t-statistics by detrending and multiple regression. *Technical Report, Division of Biostatistics, Medical College of Wisconsin*, 2002.

[9] D.W. Tank S.G. Him H. Merkle J.M. Ellermann S. Ogawa, R.S. Menon and K. Ugurbil. Functional brain mapping by blood oxygenation level-dependent constrast magnetic resonance imaging. *Biophys. J.*, pages 64:803–12, 1993.

[10] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *PAMI*, pages 888–905, 2000.

[11] G. W. Stewart. *Matrix Perturbation Theory*. Academic Press, 1999.

[12] Y. Weiss. Segmentation using eigenvectors: a unifying view. 1999.

[13] Wei X, Yoo S-S, Dickey CC, Zou KH, Guttmann CRG, and Panych LP. Functional mri of auditory verbal working memory: long-term reproducibility analysis. *NeuroImage*, page (in press), 2004.

[14] Daphna Weinshall Yoram Gdalyahu and Michael Werman. Self-organization in vision: Stochastic clustering for image segmentation, peceptual grouping, and image database organization. *PAMI*, pages 1053–1074, 2001.

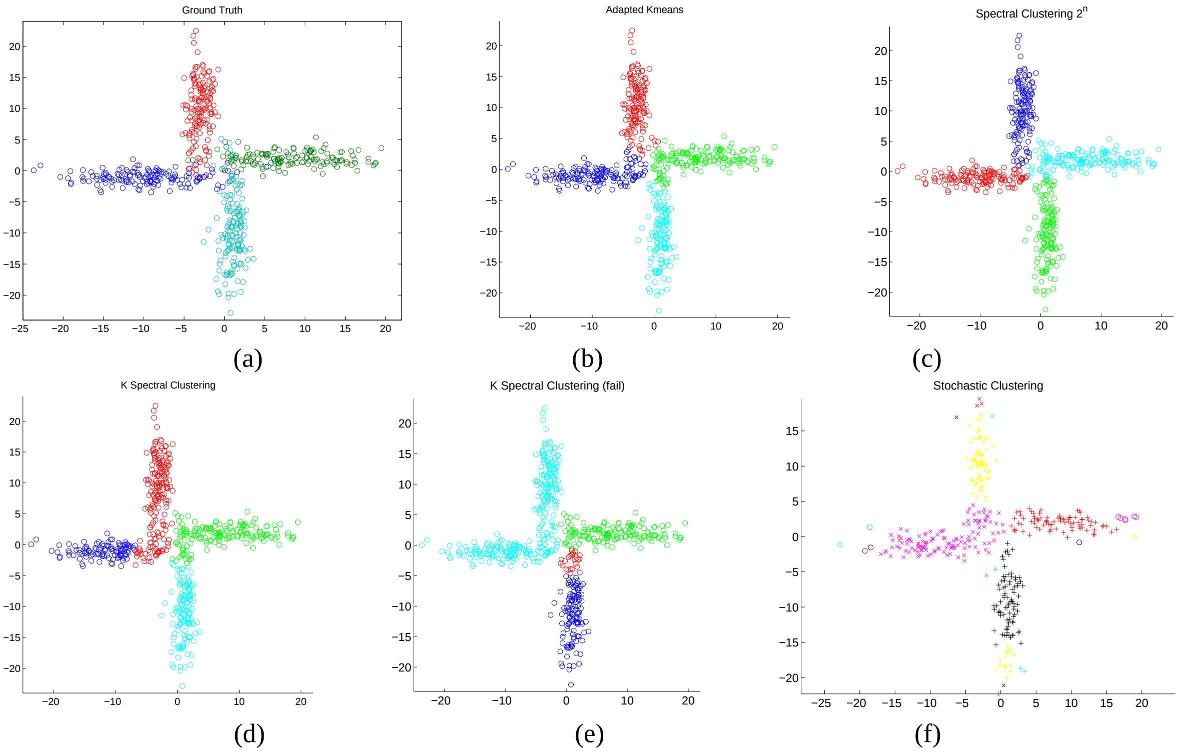


Figure 1: Clustering result for windmill mixture of Gaussian data. (a) Ground Truth (b) Adapted Kmeans (c) 2^n Spectral Clustering (d) K Spectral Clustering (e) misclassified result from K Spectral Clustering (f) Stochastic Clustering

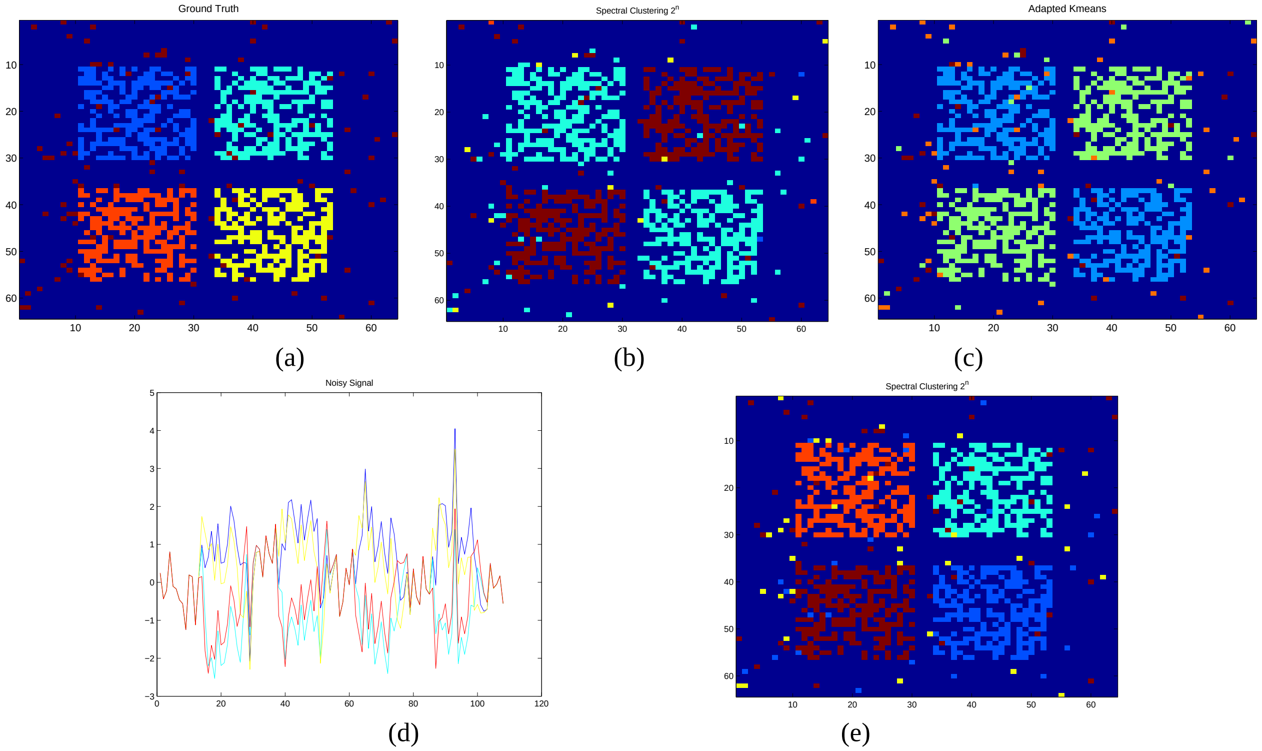
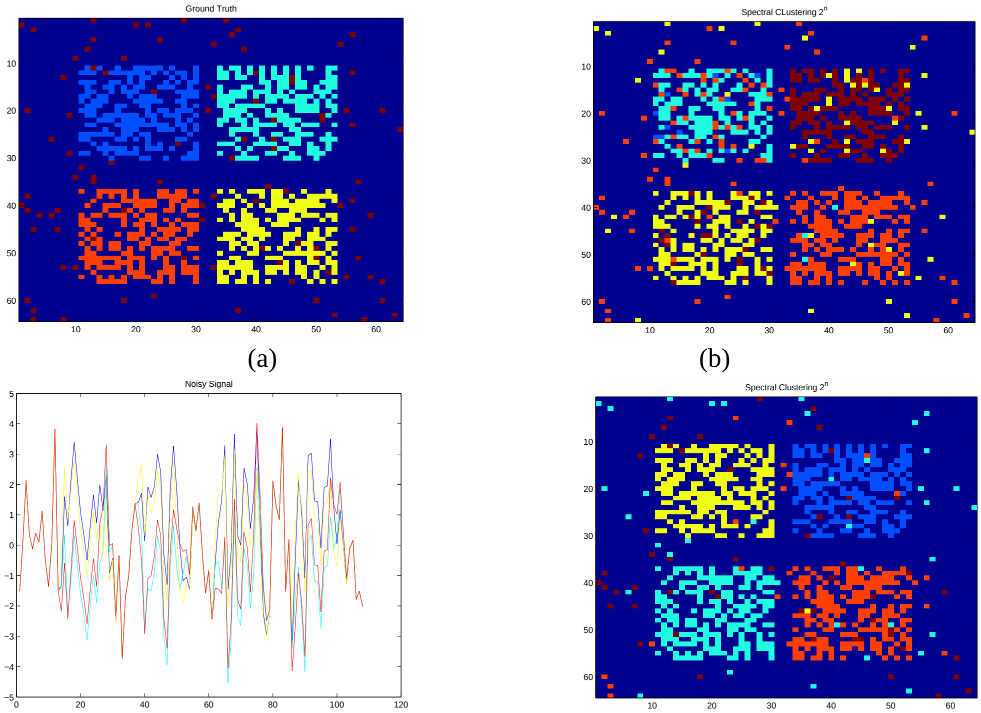


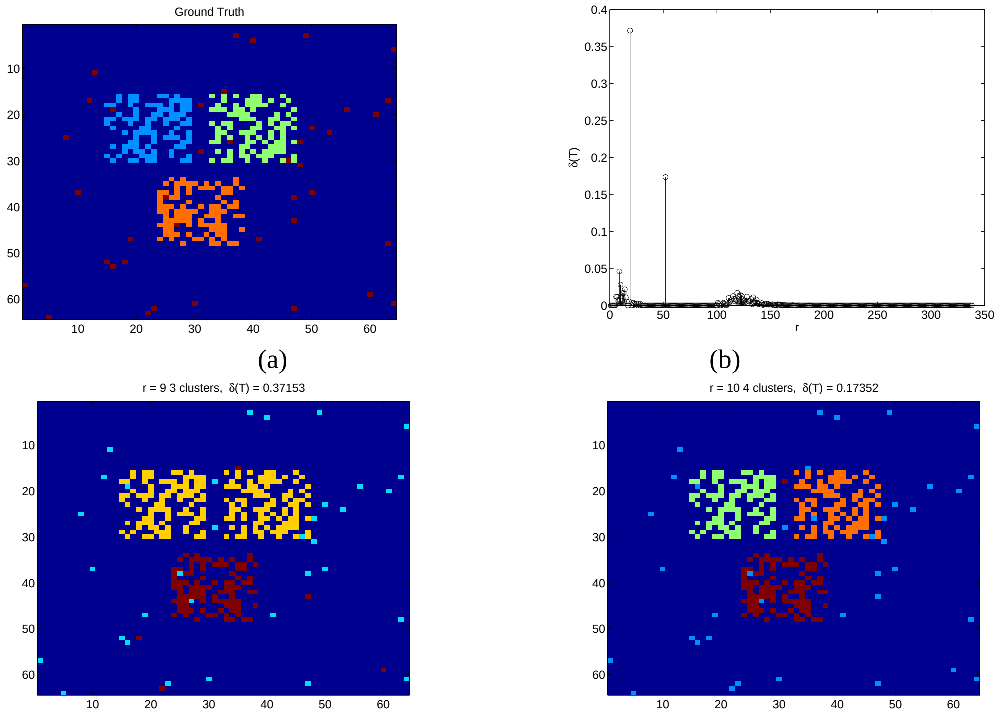
Figure 2: Results generated using signal to signal metrix from synthetic fMRI signals at SNR $= 1$. (a) is the group truth (b)(e) spectral clustering 2^n without and with Euclidean distance constraints (c) Adapted Kmeans clustering with Euclidean distance constraints. (d) input noisy signal. Colors of signals (red, blue, yellow and cyan) corresponds to respective colored blocks in (a)



(a) (b)

(c) (d)

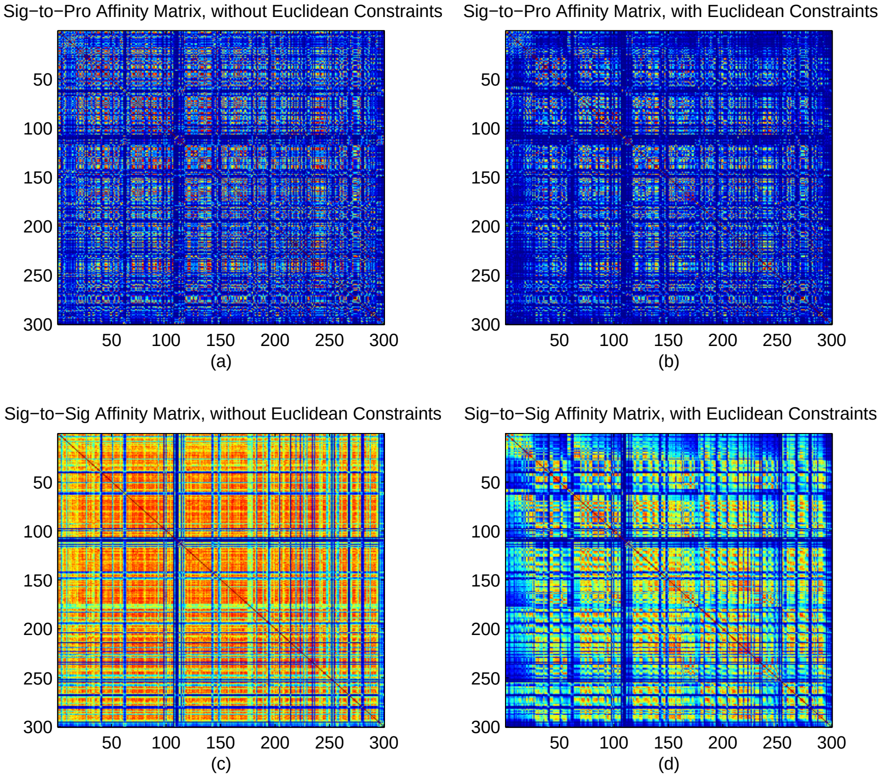
Figure 3: Results generated using signal to protocol metrix from synthetic fMRI signals at SNR $= 0.3$. (a) is the group truth (b)(d) spectral clustering 2^n without and with Euclidean distance constraints (c) input noisy signal. Colors of signals (red, blue, yellow and cyan) corresponds to respective colored blocks in (a)



(a) (b)

(c) (d)

Figure 4: Stochastics Clustering with signal to signal metric using synthetic fMRI data at SNR $= 2$. (b) illustrates $\delta(T(r))$ (c)(d) clustering result corresponds to the two largest $\delta(T)$ with XYZ constraint.
Clusters with only 1 member are displayed as cyan and blue respectively.



(a)

(b)

(c)

(d)

Figure 5: (a)(b) signal to protocol affinity matrix corresponding to with and without Euclidean constraints. (c)(d) signal to signal affinity matrix corresponding to with and without Euclidean constraints.

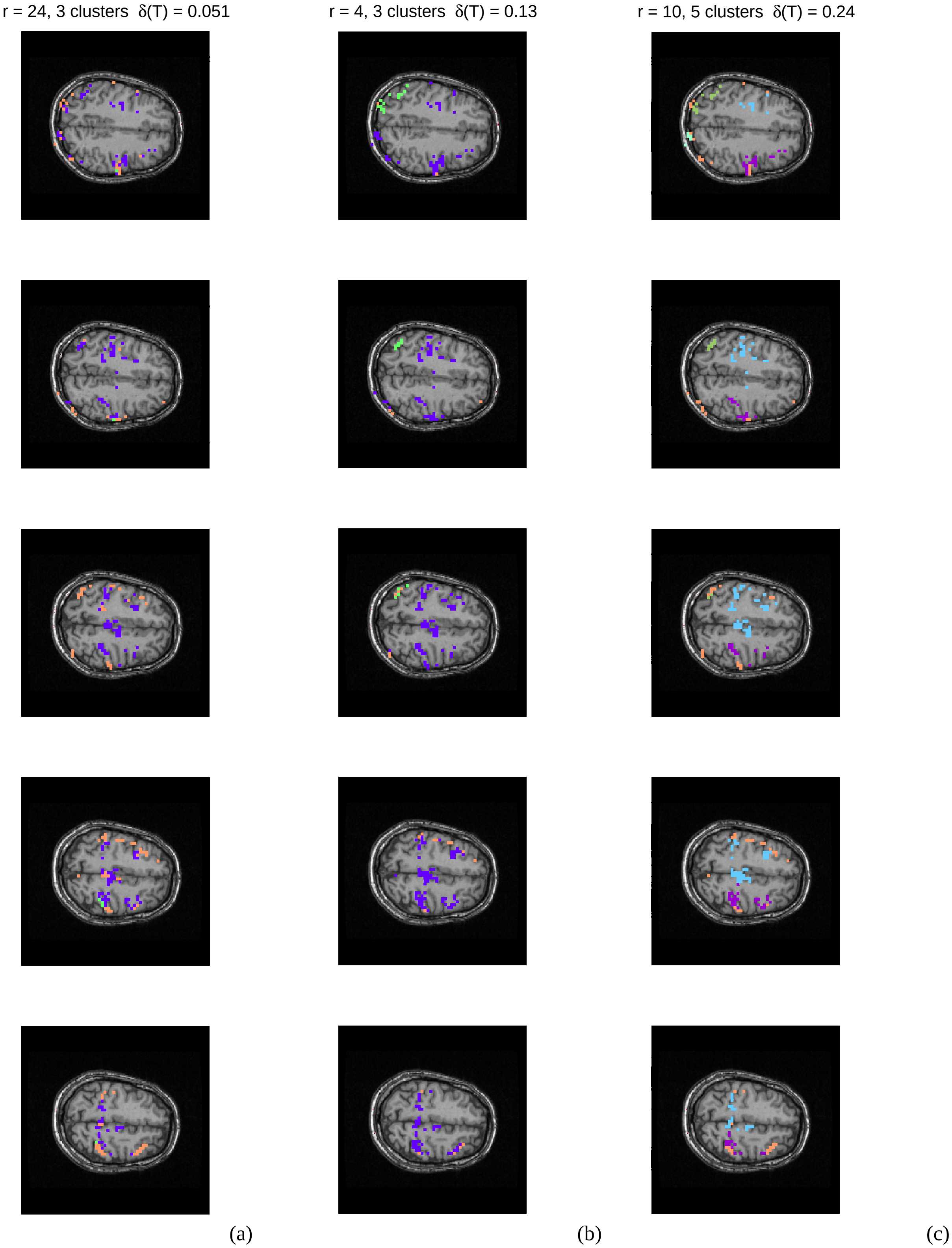

r = 24, 3 clusters δ(T) = 0.051
r = 4, 3 clusters δ(T) = 0.13
r = 10, 5 clusters δ(T) = 0.24
(a)
(b)
(c)

Figure 6: Stochastics Clustering with signal to signal metric. Clusters with only 1 member are displayed as orange. (a) clustering result corresponds to the largest $\delta(T)$ without XYZ constraint (b)(c) clustering results corresponds to the top two $\delta(T)$ with XYZ constraint.

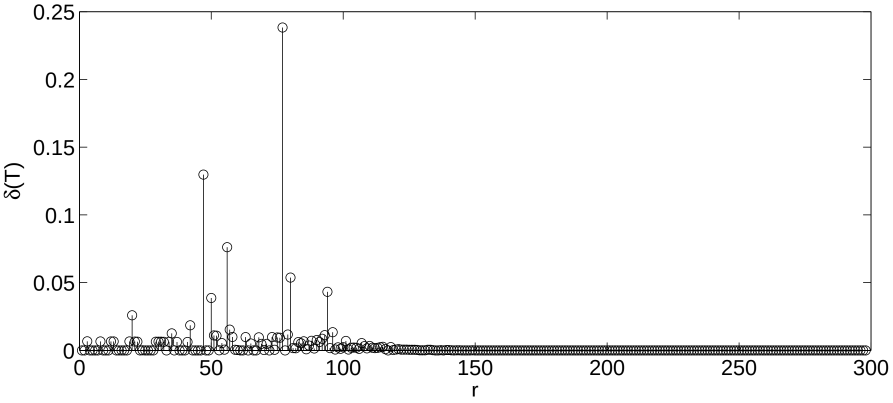


Figure 7: $\delta(T)$ of stochastics Clustering with signal to signal metric with XYZ constrain