Multiple scales for visualization large data

****

****

Abstract. Nowadays, the large datasets become more and more common. However, traditional visualization techniques, which although allow to visually analyze and explore data, can not scale well with the large one. This restrains the ability of detecting, recognizing and classifying phenomena of interest, such as patterns, clusters, trends, etc. This paper proposes a method for interactive multi-resolution visualization to overcome the problem of traditional visualization techniques when working with a large dataset via hierarchical clustering. Based on hierarchical clustering, users can not only examine the dataset at different levels of detail, but also can explore many regions of interest. The basic idea underlying this method is to choose multiple scales from the hierarchical tree for representing the data at different levels of abstraction, which creates an easy environment for interactive exploration without re-run the clustering algorithm. Moreover, we also define a criterion for evaluating the multiple scale representation based on the concept of *split factor*. An experiment of applying the proposed method into the task of interactive visualization in a clinical case study of the large dMRI (diffusion magnetic resonance imaging) data is also carried out. The results show that our proposed method efficiently provides a friendly tool for visualization the large data.

Keywords: Visualization, Hierarchical Clustering, Machine Learning, Large Data, Tract Segmentation, dMRI Data

1 Introduction

To support human in analyzing and exploring large data, it is an important task to graphically present the data [1]. Users, in one side, have a requirements of looking at complex and intricate data to find out some facts or trends that are not easy to find. On the other side, they want to explore data in details to examine each data points. In fact, the overall premise is that users have a deeper understanding about their data when they interact with the presented information and view it at different levels of abstraction [2]. During the last two decades, many interactive visualization techniques and system have been emerged [1, 3, 4]. As large data sets become more and more common, with the size over $1K$, it has been clear that most of the current visualization approaches lose their effectiveness due to they have no ability to visualize and manage the large number of data points simultaneously. In such scenario, clustering is considered a suitable method for understanding and exploring large data [5, 6].

However, clustering usually results in one partition of the data, and this leads to a dramatic drawback that the validation process is not straightforward due to the lack of ground truth data [7]. One solution for this is hierarchical clustering [8], which organizes data in an intuitive and interpretable structure, namely *dendrogram*, not only one partition as the traditional clustering methods. Such structure allows users to explore

in a simple way the clusters and the relationships between instances, and leads to many applications for visualization [9–11]. Nevertheless, when dealing practically with a large size dendrogram, it becomes difficult since the number of nodes grows exponentially with the depth of the tree and makes users lose the overview of the whole dataset. To deal with a large size dendrogram, many approaches have been suggested [6, 10, 12]. However, these methods are either display at one time only a sub-part of the structure [10, 12], or display whole dendrogram but rely on other clustering technique [6].

In this work, we propose a method of offering a complete and interactive visualization of the large data based on hierarchical clustering. Our method allow users to apply their perceptual abilities to make sense of data. The core of the problem is to obtain the multiple scales representation large data, in order to comply with the requirements of human interactive visualization. The proposed solution combines three steps. First, the dendrogram would be created by running the hierarchical clustering. Second, the *goodness* function is used as a measurement to select the most relevant scales for representing the dendrogram (it is an extension of the "relevant function", proposed in [13]). Lastly, we evaluate the multiple scales based on a statistical criteria, called *split factor*.

Moreover, we conceive an experiment of applying our method in a clinical case study of dMRI data. Recently, from dMRI data, tracking algorithms [14, 15] allow to reconstruct the $3D$ pathways of axons within the white matter as a set of streamlines, called tractography. A *streamline* is a vectorial representation of thousands of neuronal axons expressing structural connectivity, and *tractography* is a set of N streamlines ($N \sim 3 \times 10^5$ usually). It is an important task of segmentation the tractography into some real anatomical structures of interest, such as cortinal spinal tract [16, 17] involving to the amiotrophic (ALS) disease. In this experiment, we conceive a novel computer-assisted interactive segmentation process based on our method of multiple scales for representation the large tractography.

The paper is organized as follows. Section 2 formally introduces the problem of multiple scales for representation large data. After that, Section 3 describes the detail of the method for selecting multiple scales. The evaluating the goodness of the representation is presented in the next Section 4. In section 5, we describe an experiment of applying the proposed solution in the context of the tractgoraphy segmentation, provide figures to evaluate the viability of the proposed solution. We conclude with a summary of our contribution and open areas for future work in the last section 6.

2 Problem statement

In this part, after introducing the hierarchical clustering we formally describe the problem of multiple scale representation.

2.1 Hierarchical clustering

Given a set of input patterns denoted as $\mathcal{X} = \{\mathsf{x}_1, \ldots, \mathsf{x}_j, \ldots, \mathsf{x}_N\}$ where each data point $\mathsf{x}_j = (x_{j1}, x_{j2}, \ldots, x_{jd})^T \in \Re^d$ and each measure x_{ji} is said to be a feature (attribute, dimension, or variable). A hierarchical tree (or dendrogram) of $\mathcal{X}$ is defined as following:

Definition 1. *A **hierarchical tree** $\mathcal{H}$ of an N-object set $\mathcal{X} = \{\mathsf{x}_1, \ldots, \mathsf{x}_j, \ldots, \mathsf{x}_N\}$ is a collection of Q partitions on $\mathcal{X}$: $\mathcal{H} = \{\mathsf{P}_0, \ldots, \mathsf{P}_Q\}$, with $Q \leq N$, such that $\mathsf{P}_0 = \mathcal{X}$ and $C_i \in P_m, C_j \in P_l, m > l$ imply $C_i \subseteq C_j$ or $C_i \cap C_j = \emptyset$, for all $i, j \neq i, m, l = 1, \ldots, Q$.*

The hierarchical clustering algorithm [8] builds nested clusters by merging them successively, and this hierarchy of clusters represented as a tree/dendrogram. The root of the tree is the unique cluster that gathers all the samples, the leaves being the clusters with only one sample. It produces a structure of clusters of $\mathcal{X}$ that is more informative than the unstructured set of clusters returned by flat clustering. This characteristic meets the requirement of creating multiple scales of one original dataset $\mathcal{X} = \{\mathsf{x}_1, \ldots, \mathsf{x}_j, \ldots, \mathsf{x}_N\}$ without re-running the clustering algorithm again. Moreover, it leads to the capability of visualizing $\mathcal{X}$ in many levels of abstraction, and the users can browse the value of level from 1 to N, to see the clusters immediately.

Hierarchical clustering algorithms are either top down or bottom up. Bottom-up algorithms treat each streamline as a singleton cluster at the outset and then successively merge (or agglomerate) pairs of clusters until all clusters have been merged into a single cluster that contains all tracts. Bottom-up hierarchical clustering is therefore called Hierarchical Agglomerative Clustering (HAC). Top-down clustering requires a method for splitting a cluster. It proceeds by splitting clusters recursively until individual streamlines are reached [8].

2.2 Multiple scales for visualization

The hierarchical tree $\mathcal{H}$ structures and presents dataset $\mathcal{X}$ at different levels of abstraction. A non-leaf cluster is composed of all its child clusters, while a leaf cluster contains only a single data item. The collection of all leaf-clusters presents exactly every data items x_i of $\mathcal{X}$, while the root is a cluster containing whole dataset $\mathcal{X}$ as one single node of the tree.

Definition 2. *Each cluster C_i (node) of the tree $\mathcal{H}$, let $s(C_i)$ be the **level of detail** of that cluster. This measurement $s(C_i)$ satisfies the following criteria: if C_i is an ancestor of C_j, then $s(C_i) \geq s(C_j)$.*

There are many properties of a cluster which could be used to measure $s(C_i)$. Among these, two common uses are the radius of a cluster (maximum distance between all pair samples of cluster C_i: $r_i = max_{\forall \mathsf{x}_a, \mathsf{x}_b \in C_i, \mathsf{x}_a \neq \mathsf{x}_b}\{d(\mathsf{x}_a, \mathsf{x}_b)\}$); and the hierarchical level of C_i in the tree $\mathcal{H}$ [1]: $s(C_i) = \frac{height(C_i)}{h}$, where $heigh(C_i)$ is the heigh of the cluster C_i, and h is the heigh of the tree $\mathcal{H}$.

Definition 3. *The **range of scale** of a hierarchical tree $\mathcal{H}$ is $[s_{min}, s_{max}]$, where $s_{min} = \min_{\forall C_i \in \mathcal{H}}\{s(C_i)\}$, and $s_{max} = \max_{\forall C_i \in \mathcal{H}}\{s(C_i)\}$*

Depending on which property is used to measure the level of detail s_i, the value of s_{max} and s_{min} would be different. In the case of using the hierarchical level, the scale range is from $[0, 1]$, where $s_{min} = 0$ corresponds to the leaf with zero heigh, to $s_{max} = 1$ is at the root of the tree $\mathcal{H}$. However, in the case of using cluster radius, there is no guarantee that $s_{min} = 0$ and $s_{max} = 1$.

Definition 4. ***A cut** $\mathfrak{L}$ of a hierarchical tree $\mathcal{H}$ at a given scale $w \in [s_{min}, s_{max}]$ is $\mathfrak{L}(w)$:*

$$\mathfrak{L}(w) = \{C_i | (s(C_i) \leq w \wedge s(parent(C_i)) > w)\} \tag{1}$$

where $parent(C_i)$ is the direct parent node of the cluster C_i

In general, $\mathfrak{L}(w)$ is a partition of $\mathcal{X}$, denoting a subset of the tree $\mathcal{H}$. The cut at s_{min}, $\mathfrak{L}(s_{min})$ is a set of all leaf clusters, while the $\mathfrak{L}(s_{max})$ is a single cluster representing the whole dataset $\mathcal{X}$. Intuitively, $\mathfrak{L}(w)$ changes smoothly with the variance of the scale parameter w, which serves as the abstraction level of the dataset $\mathcal{X}$. It could be imagined that $\mathfrak{L}(w)$ is a cut across a vertically oriented hierarchical tree $\mathcal{H}$ that satisfies criteria: $\mathfrak{L}(w)$ intersects each path of the tree $\mathcal{H}$, from the root to the leaf, only exactly at one point. The cutting point would depend on the value of parameter w. It should close to the root of the tree $\mathcal{H}$ when w is high, and reversely. Moreover, the cut can be horizontal or unhorizontal (like zigzag) as long as for each path from the root to the leaf of the tree $\mathcal{H}$, there is only one crossing with $\mathfrak{L}(w)$. It is an open approach for cutting the tree comparing with the traditional one which only accepts the horizontal cut.

Definition 5. *Let P and Q be two partitions of dataset $\mathcal{X}$, $P = \{C_1^P, \ldots, C_l^P\}$ and $Q = \{C_1^Q, \ldots, C_m^Q\}$. Partition P is **nested in** partition Q, denoted as $P \preceq Q$, if and only if*

$$P \preceq Q \leftrightarrow \forall C_i^Q \in Q, \exists C_{i_1}^P, \ldots, C_{i_k}^P \in P : C_i^Q = \cup_{t=1}^{k} C_{i_t}^P \tag{2}$$

Definition 6. *Given the scale range $[s_{min}, s_{max}]$ of a tree $\mathcal{H}$, the **multiple scales representation** for the tree $\mathcal{H}$ is an* ordered set *of k scale values from $[s_{min}, s_{max}]$: $B = \{b_1, b_2, \ldots, b_k\}, b_i \in [s_{min}, s_{max}], \forall i \in [1, k]$, where k is the order of set B, which satisfies the following condition:*

$$\forall i \in [1, \ldots, k-1] : \mathfrak{L}(b_i) \preceq \mathfrak{L}(b_{i+1}) \tag{3}$$

Multiple scale representation problem: Given a hierarchical tree $\mathcal{H}$ on a dataset $\mathcal{X}$, with the scale range $[s_{min}, s_{max}]$. How to choose the multiple scales representing for the tree $\mathcal{H}$: $B = \{b_1, b_2, \ldots, b_k\}, b_i \in [s_{min}, s_{max}], \forall i \in [1, k]$?

It is an $NP - problem$, and there is no general solution for it. Usually, it is chosen that $b_1 = s_{min}$, where the whole elements of $\mathcal{X}$ are presented, and $b_k = s_{max}$, which corresponds to only one virtual representation of $\mathcal{X}$. However, the value of k is an open question and totally depends on the application. In the next section, we will discuss about how to define the k value and also how to select each b_i from $[s_{min}, s_{max}]$.

3 Methods

In this part, we present a simple and efficient method to determine the multiple scales $B = \{b_1, b_2, \ldots, b_k\}, b_i \in [s_{min}, s_{max}], \forall i \in [1, k]$, where $[s_{min}, s_{max}]$ is the range scale of the hierarchical tree $\mathcal{H}$, constructed from dataset $\mathcal{X}$. The multiple scales B, moreover, have to satisfy the condition in Definition 6.

With each cluster $C_i \in \mathcal{H}$, let $(\alpha_{min}^{C_i}, \alpha_{max}^{C_i})$ be two scale factors at which the cluster C appears and disappears from the tree $\mathcal{H}$.

Definition 7. *Given a cluster C_i in a hierarchical tree $\mathcal{H}$, with range scale $[s_{min}, s_{max}]$, the pairwise $(\alpha_{min}^{C_i}, \alpha_{max}^{C_i})$ is defined as:*

$$\begin{aligned} \alpha_{min}^{C_i} &= \min\{w_j | w_j \in [s_{min}, s_{max}] \wedge C_i \in \mathfrak{L}(w_j)\} \\ \alpha_{max}^{C_i} &= \max\{w_j | w_j \in [s_{min}, s_{max}] \wedge C_i \in \mathfrak{L}(w_j)\} \end{aligned} \tag{4}$$

Pascal et. al. [13] proposed a method to compute B from pairwise $(\alpha^C_{min}, \alpha^C_{max})$. It is considered that the good clusters would be presented for a wide range of scale factors. Thus, the goodness of a cluster could be measured as $\alpha^C_{min} - \alpha^C_{max}$) and the best scale representing C_i as $\alpha = \frac{\alpha^C_{max} - \alpha^C_{min}}{2}$.

Definition 8. *The **goodness function** $R(C)$ **of a cluster** C at a scale w is:*

$$R_w(C) = \frac{\alpha^C_{max} - \alpha^C_{min}}{2} + \frac{2(\alpha^C_{max} - w)(w - \alpha^C_{min})}{\alpha^C_{max} - \alpha^C_{min}} \tag{5}$$

Definition 9. *Given a scale $w \in [s_{min}, s_{max}]$, the **goodness fucntion** $R(C)$ **of a scale** w is:*

$$R(w) = \frac{1}{N} \sum_{C \in \mathfrak{L}(w)} |C| R_w(C) \tag{6}$$

A plot line of the $R(w)$ function can be found in the Figure 1. Obviously, $R(w)$ is a quadratic function of w, and can be used for determining the scale factors corresponding to the good clusters. By focussing on the local maxima of $R(w)$, we can estimate good scales for representing the tree $\mathcal{H}$, and thus getting the $B = \{b_1, b_2, \ldots, b_k\}, b_i \in [s_{min}, s_{max}], \forall i \in [1, k]$. In another way, the first derivative of $R(w)$ is set to zero, and we arrive a set of multiple scales B.

$$B = \{b_i | b_i \in [s_{min}, s_{max}] \wedge R'(b_i) = 0\} \tag{7}$$

The most difficult task is to compute the pairwise $(\alpha^{C_i}_{min}, \alpha^{C_i}_{max})$ for each cluster $C_i \in \mathcal{H}$. Pascal et. al in [13] proposed a method to calculate $(\alpha^C_{min}, \alpha^C_{max})$ based on the concept of *relevant community*. However, the proposed procedure is computational cost, and the complexity is between $O(n \log n)$ and $O(n^2)$ with an average value in $O(n\sqrt{n})$. As the meanwhile, the hierarchical order of the tree $\mathcal{H}$ provides a good hint about the scales where each cluster appears or disappears. By exploring this information, we suggest a more easy and efficient way with the complexity $O(1)$: $\alpha^{C_i}_{min} = s(C_i)$ and $\alpha^{C_i}_{max} = s(parent(C_i))$, where $s(C_k)$ is the level of detail of cluster C_k. Obviously, the suggested computing $(\alpha^{C_i}_{min}, \alpha^{C_i}_{max})$ is intuitively, as the Definition 2.

4 Criteria for evaluation

In this part we describe criteria for evaluating the multiple scales representation $B = \{b_1, b_2, \ldots, b_k\}$ for the dataset $\mathcal{X}$. It is the real fact that, at a certain time, users can only exam about the total of 50 (λ_1) clusters which are currently displaying on the screen. Among of the visible clusters, users usually select around 15 (λ_2) clusters to explore or exam the data [18]. Driven from that, we propose a method to evaluate the represented multi-scale set B based on the *split factor* as following.

Definition 10. ***Split factor** ξ **of a cluster** $C \in \mathcal{H}$ to a scale $w \in [s_{min}, s_{max}]$ is $\xi(C, s)$*

$$\xi(C, s) = card(P(C, s)) \tag{8}$$

where $P(C, s) = \{C_j | (C_j \in \mathcal{H}) \wedge (s(C_j) = w) \wedge (C_j \subseteq C)\}$

Definition 11. ***Split factor*** ξ *of a* ***set of clusters*** $P = \{C_1, C_2, .., C_m\} \subseteq \mathcal{H}$ *to a scale* $s \in [s_{min}, s_{max}]$ *is* $\xi(P, s)$

$$\xi(P, s) = \sum_{C_i \in P} \xi(C_i, s) \tag{9}$$

Definition 12. *The set of scales* $B = \{b_1, b_2, \ldots, b_k\}$ *is called* ***the best scales for representation*** *of the tree* $\mathcal{H}$*, given* λ_1 *and* λ_2*, if the following condition satisfies*

$$\forall b_i \in B : \lambda_1 - \Delta \leq \xi(S_{(b_i,\lambda_2)}, b_{i-1}) \leq \lambda_1 + \Delta \tag{10}$$

where $S_{(b_i,\lambda_2)}$ *is a Gaussian distribution subset of the cut* $\mathcal{H}$ *at scale* b_i*,* $\mathfrak{L}(\mathfrak{b}_\mathfrak{i})$*, with the order of* λ_2*:*

$$S_{(b_i,\lambda_2)} = \{C_1, \ldots, C_{\lambda_2}\}, C_j \in \mathfrak{L}(\mathfrak{b}_\mathfrak{i}), \forall j \in [1, \ldots, \lambda_2] \tag{11}$$

In the case of b_1, the split factor is computed to the leaf: $\xi(S_{(b_1,\lambda_2)}, 0)$. The pseudo code of evaluation procedure is presented in algorithm 1

Algorithm 1: Evaluate the set of cut scales, based on split factor

Require: a hierarchical tree $\mathcal{H}$ **and**
a set of cut scales $\mathsf{B} = \{b_1, b_2, \ldots, b_k\}$
Ensure: accept B as a good represent for $\mathcal{H}$ or not

1: *accept* $\leftarrow$ **true** {initialization}
2: $i \leftarrow 1$
3: **while** $(i \leq k - 1)$ **and** (*accept*) **do**
4: $w \leftarrow b_i$
5: $l \leftarrow b_{i+1}$
6: $S \leftarrow$ choose λ_2 clusters uniformly at random from $\mathfrak{L}(w)$
7: $t \leftarrow \xi(S, l)$ {split factor of S to scale l}
{check quality of the cut b_i based on equation 10}
8: **if** $(t \geq \lambda_1 + \Delta)$ **and** $(t \leq \lambda_1 - \Delta)$ **then**
9: *accept* $\leftarrow$ **false** {update the result}
10: $i \leftarrow i + 1$;
11: **return** *accept*

5 Experiments

In the following we briefly describe our experiment for visualizing the real large tractography for the validation of the proposed approach. The evaluation based on split factor and one heuristic trick to improve the visualization result are also presented.

5.1 dMRI and tractography segmentation

Let the polyline $s = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x}_{n_s}\}$, where $\boldsymbol{x} \in \mathbb{R}^3$, be a *streamline* reconstructed from dMRI data by deterministic tractography algorithms [14]. Let the *tractography* $\mathbb{T} = \{s_1, \ldots, s_N\}$ be defined as a set of N streamlines. Our experiment is motivated by a clinical research hypothesis about the characterisation of the amiotrophic (ALS) disease,

which is known to be affected by the corticospinal tract (CST) [16, 17]. The first task is to segment the CTS from the full brain tractography $\mathbb{T}$.

In spite that recently there is an increasing literature in automatic tractography segmentation using machine learning techniques [19, 20], applications in the clinical domain rely on manual segmentation. The manual segmentation process consumes a lot of time and effort due to the large number of streamlines, in the order of 3×10^5, which make it intrinsically difficult both to inspect and to unfold the anatomical structures. Moreover, it is claimed that there is a lack of software tools to support and to simply this segmentation process [18]. In this experiment, we conceive a novel computer-assisted interactive process based on the method of multiple scales for representation the large tractography described in Section 3. After computing the set of multiple scales $B = \{b_1, b_2, \ldots, b_k\}$, our tool first displays $\mathbb{T}$ as the cut at b_1, $\mathfrak{L}(b_1)$; and let user select some of clusters to identify a superset of the streamlines of interest. This superset is then to be displayed at the next scale and again the user is requested to select the relevant clusters. The process of re-display and manual selection is iterated until the remaining streamlines faithfully represent the desired anatomical structure of interest.

ALS dataset: the data we used in this experiment is recorded with a $3T$ scanner at Utah Brain Institute. It consisted the recordings of 12 ALS patients and 12 healthy controls; 64 (+1, i.e. $b = 0$) gradients; b-value= 1000; anatomical scan ($2 \times 2 \times 2mm^3$). We reconstruct the streamlines using EuDX, a deterministic tracking algorithm [21] from the DiPy library [1].

Dissimilarity representation: due to the fact that each streamline has different length and different number of points we need to find a representation ϕ of streamline in a vectorial space, by mapping a streamline s from its original space $\mathbb{T}$ to a vector of $\mathbb{R}^d$ - $\phi : \mathbb{T} \mapsto \mathbb{R}^d$, where d is the dimension of the new space. One suggestion for this is the *dissimilarity representation* [22]. It is a lossy Euclidean embedding algorithm was previously proposed in [23] for streamlines. The dissimilarity representation is defined as $\phi^d_{\Pi}(X) : \mathcal{X} \mapsto \mathbb{R}^p$ s.t. $\phi^d_{\Pi}(X) = [d(X, \tilde{X}_1), \ldots, d(X, \tilde{X}_p)]$, where d is a distance function between streamlines, and $\Pi = \{\tilde{X}_1, \ldots, \tilde{X}_p\} \subset \mathcal{X}$ is a set of p streamlines called *prototypes*. More detail can be found in [23, 24].

By applying the hierarchical clustering algorithm (section 2.1) on the dissimilarity approximation, the hierarchical tree $\mathcal{H}$ of the tractography $\mathbb{T}$ could be created.

5.2 Multiple scales for representation

As the measurement for computing the level of detail of a cluster $s(C_i)$, we use the height of the cluster C_i within the hierarchical tree $\mathcal{H}$. The reason is that this measurement leads to continuous and thus provides smooth transitions on our hierarchical display.

Let h be the height of hierarchical tree $\mathcal{H}$: $h = height(\mathcal{H})$, at the leaf C_{leaf} of $\mathcal{H}$, the heigh is in the order of zero, thus $s_{min} = 0$. In the similar way, $s_{max} = 1$ because at the root C_{root} of the tree $\mathcal{H}$, $heigh(C_{root}) = h$. The range scale of $\mathcal{H}$ is $[s_{min}, s_{max}] = [0, 1]$, and $\forall C_i \in \mathcal{H}, s(C_i) = \frac{height(C_i)}{h}$, where $heigh(C_i)$ is the heigh of the cluster C_i [1]. Intuitively, this measurement satisfies the condition of Definition 2 about the level of detail in, because if C_i is an ancestor of C_j, then $heigh(C_i) \geq heigh(C_j)$ and thus, $s(C_i) \geq s(C_j)$

By looking at the local maxima of the goodness score as definition in 9, we can estimate the most relevant scale factors for representing the tree $\mathcal{H}$, and thus getting

[1] `http://www.dipy.org`

the multiple scale representation $B = \{b_1, b_2, \ldots, b_k\}$. The figure 1 shows the plot lines of two goodness scores of subject 109(left) and control 205(right) from ALS dataset. For example with subject 109 (left) the multiple scale representation B_1 could be concluded as $B_1 = \{\frac{8}{h_1}, \frac{10}{h_1}, \frac{12}{h_1}, \frac{18}{h_1}, \frac{22}{h_1}, \frac{25}{h_1}, \frac{27}{h_1}\}$, where h_1 is the heigh of the hierarchical tree$\mathcal{H}_{109}$ of subject 109: $h_1 = height(\mathcal{H}_{109})$. Similarly, with control 205, $B_2 = \{\frac{10}{h_2}, \frac{16}{h_2}, \frac{19}{h_2}, \frac{24}{h_2}, \frac{27}{h_2}\}$, where $h_2 = height(\mathcal{H}_{205})$. Taking into account that b_i should be chosen from the small scale factor to the large one in order to satisfy the condition of an ordered set in the Definition 6, of which the underlying idea is to make sure a continuous and smooth order of visualization when users switch among these levels. This experiment exams the ability of our proposed method to compute the multiple scales representation for a large data. Note that we just present here the two samples of results, we also run on other subjects and get the equivalent multiple scale representation for each of them.

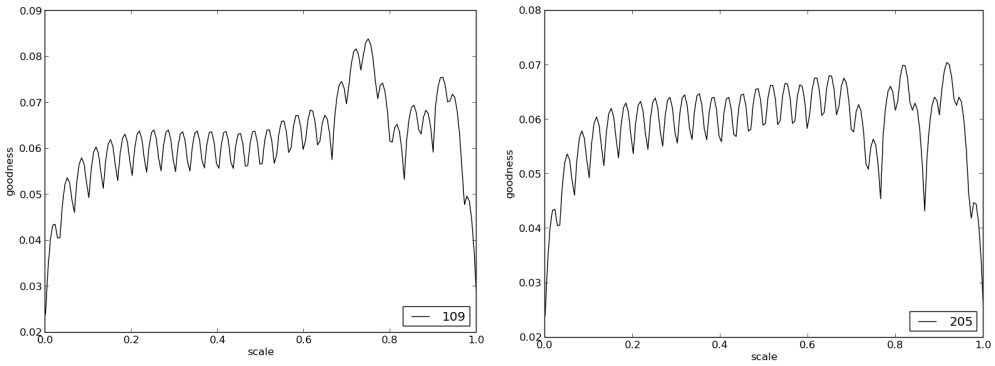

Fig. 1. Goodness score of subject 109 and 205 from ALS dataset

We are now in the state of being ready to evaluate the multiple scale representation $B = \{b_1, b_2, \ldots, b_k\}$. Based on the Definition 12 about the goodness of a scale factor $b \in B$, we implemented a program as the pseudo code in 1 with $\lambda_1 = 50$ and $\lambda_2 = 15$. In figure 1 - left, we plot the mean goodness score of each multiple scale representation for subject 109 and control 205 from ALS dataset, together the standard derivation for 20 iterations. Note that on the horizontal axis, the cut scales l represented on the figure is the index of the corresponding real scale $\frac{l}{h}$.

Exept for the first chosen scale b_1, almost other scale $b_k \in B, k \neq 1$, the split factor $\xi(S_{(b_i,\lambda_2)}, b_{i+1})$ satisfies the condition in Equation 10. Note that from the leaf (scale 0) the goodness score increases linearly, reaches the peak at scale B_1, and then gets a fluctuating variety. It shows that the split factor of the cut at b_1 to the leaf (scale 0), $\xi(S_{(b_1,\lambda_2)}, 0))$, is usually very large comparing with other split factors. However, in the point view of visualization, all the split factor $\xi(S_{(b_i,\lambda_2)}, b_{i+1})$ should be around λ_1. Due to this, we add an heuristic constrain to the chosen scale set: the distance between b_i and b_{i-1} should not exceed a threshold δ: $d(b_i, b_{i-1}) \leq \delta$ (in the case of b_1, the distance with the leaf, $d(b_1, 0)$, is used). The results after adding constrain are showed in the figure 1 - right (with $\delta = 4/h$) where the mean split factor is close to $\lambda_1 = 50$. This approach is run on many subjects from ALS dataset, the sizes of which vary from $200K$ to $300K$. These results demonstrate that our proposed method of choosing multiple scales for visualization large data is efficient and robust to the size of the large data.

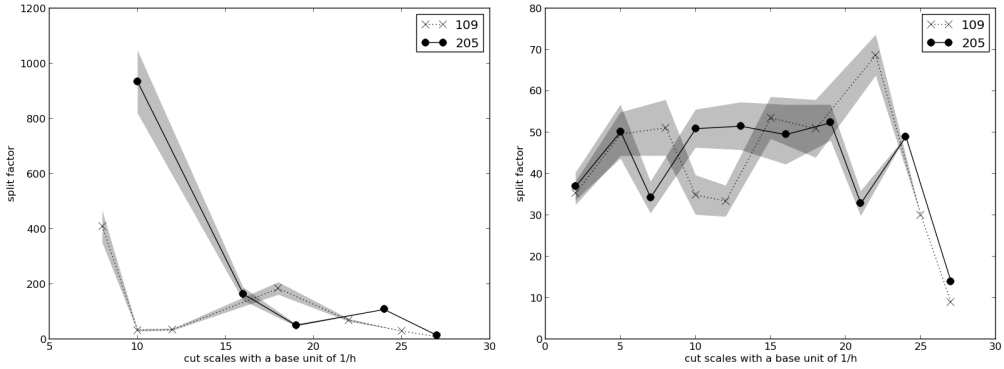

Fig. 2. Split factor before(left) and after(right) adding heuristic constrains

6 Conclusion

In this paper, we presented a method for addressing the problem faced when attemping to interactively visualize a large dataset. The core principle behinds the framework was to choose *multiple scales for representing* the data from the hierarchical clustering. Moreover, we also proposed a function to evaluate the goodness of each chosen scale based on the concept of *split factor*. We instantiate this framework with an application of building the interactive visualization large dMRI data in the procedure of tractogaphy segmentation, and provide concrete result on it performance. Experiments have shown that our method provide a significant improvement for visualization the large data at different scales, which verifies the effectiveness of the interactive hierarchical visualization. Beside, we are convinced that this method can be easily integrated to any current display techniques without having to vary the data or the interactive exploration tool.

As mention is the section 3, the level of detail of each cluster, $s(C_i)$, can be computed based on radius or heigh [1]. In this paper we choose the multiple scales only based on the heigh of cluster. The same job but based on the radius needs to be investigated. Moreover, this work is a part of an going resarch project focusing on computer-aided tractography segmentation, where machine learning techniques are used to assist medical practitioners to do the segmentaion task more easily, flexibly and effectively. In the future, we want to further improve the interactive segmentation tool by providing the function of adding or eleminating data points x into or from the current dataset $\mathcal{X}$, and updating the visualization result without re-run the clustering algorithm.

References

1. Yang, J., Ward, M.O., Rundensteiner, E.A.: Interactive hierarchical displays: a general framework for visualization and exploration of large multivariate data sets. Computers & Graphics **27**(2) (April 2003) 265–283
2. Roberts, J.C.: State of the Art: Coordinated & Multiple Views in Exploratory Visualization. In: Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV '07. 5th Intl Conference on, Washington, DC, USA, IEEE (July 2007) 61–71
3. Stroe, I., Rundensteiner, E., Ward, M.: Scalable Visual Hierarchy Exploration. In Ibrahim, M., Küng, J., Revell, N., eds.: Database and Expert Systems Applications. Volume 1873 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2000) 784–793

4. Fua, Y.H., Ward, M.O., Rundensteiner, E.A.: Structure-Based Brushes: A Mechanism for Navigating Hierarchically Organized Data and Information Spaces. IEEE Transactions on Visualization and Computer Graphics **6**(2) (April 2000) 150–159
5. Berkhin, P.: A Survey of Clustering Data Mining Techniques. In Kogan, J., Nicholas, C., Teboulle, M., eds.: Grouping Multidimensional Data. Springer Berlin Heidelberg, Berlin/Heidelberg (2006) 25–71
6. Bisson, G., Blanch, R.: Improving Visualization of Large Hierarchical Clustering. In: Information Visualisation (IV), 2012 16th Intl Conference on, IEEE (July 2012) 220–228
7. Candillier, L., Tellier, I., Torre, F., Bousquet, O.: Cascade Evaluation of Clustering Algorithms. In Fürnkranz, J., Scheffer, T., Spiliopoulou, M., eds.: Machine Learning: ECML 2006. Volume 4212 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2006) 574–581
8. Johnson, S.: Hierarchical clustering schemes. Psychometrika **32**(3) (September 1967) 241–254
9. Heard, J., Kaufmann, W., Guan, X.: A novel method for large tree visualization. Bioinformatics (Oxford, England) **25**(4) (February 2009) 557–558
10. von Landesberger, T., Kuijper, A., Schreck, T., Kohlhammer, J., van Wijk, J.J., Fekete, J.D., Fellner, D.W.: Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. Computer Graphics Forum **30**(6) (September 2011) 1719–1749
11. Mahé, P., Vert, J.P.: Graph kernels based on tree patterns for molecules. Machine Learning **75**(1) (April 2009) 3–35
12. Furnas, G.W.: A fisheye follow-up: further reflections on focus + context. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM (2006) 999–1008
13. Pons, P., Latapy, M.: Post-processing hierarchical community structures: Quality improvements and multi-scale view. Theor. Comput. Sci. **412**(8-10) (March 2011) 892–900
14. Mori, S., van Zijl, P.C.M.: Fiber tracking: principles and strategies, a technical review. NMR Biomed. **15**(7-8) (2002) 468–480
15. Zhang, S., Correia, S., Laidlaw, D.H.: Identifying White-Matter Fiber Bundles in DTI Data Using an Automated Proximity-Based Fiber-Clustering Method. IEEE Transactions on Visualization and Computer Graphics **14**(5) (September 2008) 1044–1053
16. Cosottini, M., Giannelli, M., Vannozzi, F., Pesaresi, I., Piazza, S., Belmonte, G., Siciliano, G.: Evaluation of corticospinal tract impairment in the brain of patients with amyotrophic lateral sclerosis by using diffusion tensor imaging acquisition schemes with different numbers of diffusion-weighting directions. Journal of computer assisted tomography **34**(5) (2010) 746–750
17. Sage, C.A., Van Hecke, W., Peeters, R., Sijbers, J., Robberecht, W., Parizel, P., Marchal, G., Leemans, A., Sunaert, S.: Quantitative diffusion tensor imaging in amyotrophic lateral sclerosis: revisited. Human brain mapping **30**(11) (November 2009) 3657–3675
18. Olivetti, E., Nguyen, T.B., Avesani, P.: Fast Clustering for Interactive Tractography Segmentation. the 3rd IEEE Intl Workshop on Pattern Recognition in NeuroImaging (2013)
19. Wang, X., Grimson, W.E., Westin, C.F.F.: Tractography segmentation using a hierarchical Dirichlet processes mixture model. NeuroImage **54**(1) (January 2011) 290–302
20. Olivetti, E., Avesani, P.: Supervised segmentation of fiber tracts. In: Proceedings of SIMBAD'11. SIMBAD'11, Berlin, Heidelberg, Springer-Verlag (2011) 261–274
21. Garyfallidis, E.: Towards an accurate brain tractography. PhD thesis, University of Cambridge (2012)
22. Pekalska, E., Paclik, P., Duin, R.P.W.: A generalized kernel approach to dissimilarity-based classification. J. Mach. Learn. Res. **2** (2002) 175–211
23. Olivetti, E., Nguyen, T.B., Garyfallidis, E.: The Approximation of the Dissimilarity Projection. IEEE Intl Workshop on Pattern Recognition in NeuroImaging **0** (2012) 85–88
24. Pekalska, E., Duin, R., Paclik, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recognition **39**(2) (February 2006) 189–208