QuickBundles for Real-Time Tractography Segmentation

Eleftherios Garyfallidis, Matthew Brett, Marta Morgado Correia, Guy B. Williams and Ian Nimmo-Smith

***Abstract*—Diffusion MR white matter tractographies are very large data sets which are hard to visualize, interact with, and interpret in a clinically acceptable time scale, despite numerous proposed approaches. As a solution we present a simple, compact, tailor-made clustering algorithm, QuickBundles (QB), that overcomes the complexity of these large datasets and provides anatomically meaningful clusters in seconds. Each QB cluster can be represented by a single track; collectively these representative tracks can be taken as a faithful representation of the tractography. We demonstrate how these presentative tracks can be used as input to clustering algorithms of higher order complexity which currently cannot be implemented on the full data set. We also show the further potential of QB to find landmarks, create atlases, compare and register tractographies.**

***Index Terms*—Tractography, Diffusion MRI, Fiber clustering, White matter atlas, Direct tractography registration, Clustering algorithms, DTI.**

I. Introduction

Following the acquisition of diffusion MR scans, processes of reconstruction and integration are performed to create a tractography: a data set composed of tracks, which are sequences of points in 3D space. Irrespective of the types of reconstruction and integration a tractography can contain a very large number of tracks (up to 10^6) depending principally on the number of seed points used to generate the tractography but also on how the propagation algorithm handles voxels with underlying fiber crossings.

The size of these tractographies makes them difficult to interpret and visualize. A clustering of some kind seems to be an obvious route to simplify the complexity of these data sets and provide a useful segmentation. As a result, during the last 10 years there have been numerous efforts by many researchers to address both unsupervised and supervised learning problems of brain tractography. Though these studies do provide many useful ideas, all these methods suffer ultimately from lack of practical efficiency.

Current clustering techniques and the principal studies that have applied them to tractographies include: *Hierarchical clustering* [1], [2], [3], [4], [5]; *k-means* [6], [7]; *Adaptive mean shift* [8], [9]; *Graph theoretic segmentation* [10]; *k-nearest neighbours* [11]; *Generalized Procrustes Analysis and Principal Components Analysis (PCA)* [12], [13], [14]; *EM clustering* [15], [16], [17], [18]; *Spectral clustering* [19], [20]; *Affinity propagation* [21], [22]; *Hierarchical Dirichlet process mixture model* [23]; *Current models* [24], [25]. For recent reviews see [20], [23].

E. Garyfallidis is with the University of Cambridge, Cambridge, UK, e-mail: garyfallidis@gmail.com.

M. Brett is with the University of California, Henry H. Wheeler, Jr. Brain Imaging Center, Berkeley, CA.

M. M. Correia and I. Nimmo-Smith are with the MRC Cognition and Brain Sciences Unit, Cambridge, UK.

G. B. Williams is with the Wolfson Brain Imaging Centre, University of Cambridge, Cambridge, UK.

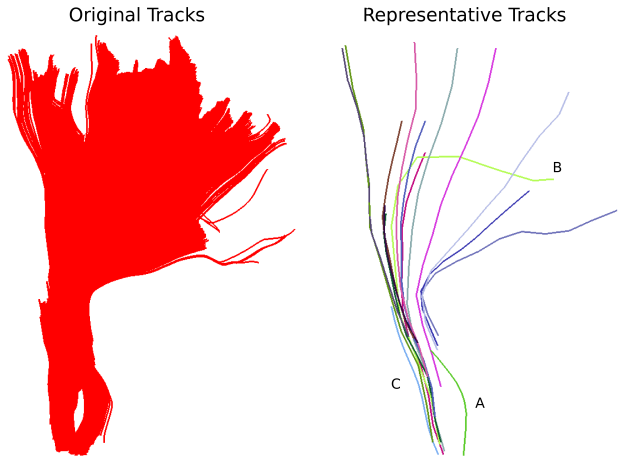

Figure 1. Part of the CST bundle (red) consisting of 11041 tracks labelled by an expert. At first glance it looks as though all tracks have a similar shape, and possibly converge towards the bottom, and fan out towards the top. However, this is a misreading caused by the opaque density when all the tracks are visualised. QB can help us see the finer structure of the bundle and identify its elements. On the right hand side we see the 14 QB representative tracks of the CST. We can now clearly see that several parts which looked homogeneous are actually broken bundles e.g. dark green (A), light blue (C), or bundles with very different shape e.g. light green (B). To cluster this bundle took 0.1 seconds.

Most of these proposed tractography clustering algorithms are very slow and many need to calculate a matrix of inter-track distances of size $\mathcal{O}(N^2)$. This number of computations puts a very heavy load on clustering algorithms, making them hard to use for everyday analysis as it is difficult to compute all these distances or store them in memory. For the same reason, no current algorithm is practical for real time clustering on a large number of tracks. The heavy computational demands of clustering add a further overhead to the use of tractography for clinical applications but also puts a barrier on understanding and interpreting the quality of diffusion MR data sets.

To address these key issues of time and space we

present a stable, generally linear time clustering algorithm that can generate meaningful clusters of tracks in seconds with minimum memory consumption. Our approach is extremely straightforward and we do not need to calculate all pairwise distances unlike most existing methods. Furthermore we can update our clustering online or in parallel. In this way we can overcome the previous barriers of space and time.

We show that we can generate these clusters ~1000 times faster than any other available method before even applying further acceleration through parallel processing, and that it can be used to cluster from a few hundred to many millions of tracks.

Moreover our new algorithm leads to many valuable additional results. QB can either be used on its own to explore the neuroanatomy directly, or it can be used as a precursor tool which reduces the dimensionality of the data, which can then be used as an input to other algorithms of higher order complexity, resulting in their greater efficiency. Beyond the use of this algorithm to simplify tractographies, we show how it can help find landmarks, create atlases, and compare and register tractographies.

II. Methods

A. The QB algorithm

QuickBundles (QB) is a surprisingly simple and very fast algorithm which can reduce tractography representation to an accessible structure in a time that is linear in the number of tracks N. QB is an extended update on our preliminary work [26].

In QB each item, a track, is a fixed-length ordered sequence of points in $\mathbb{R}^3$, and QB uses metrics and amalgamations which take account of and preserve this structure. Moreover each item is either added to an existing cluster on the basis of the distances between the cluster descriptor of the item and the descriptors of the current list of clusters. Clusters are held in a list which is extended according to need. Unlike amalgamation clustering algorithms such as k-means [27], [28] or BIRCH [29], there is no reassignment or updating phase – once an item is assigned to a cluster it stays there, and clusters are not amalgamated.

A track is a ordered sequence of points in $\mathbb{R}^3$. The clustering algorithm needs a measure of distance between two tracks, and QB uses a particular distance measure that we call minimum average direct flip (MDF). The MDF measure requires that each track be resampled to have K points. We describe the MDF measure and the resampling in section II-B.

We index the tracks with $i = 1 \dots N$ where $\mathbf{s}_i$ is the $K \times 3$ matrix representing track i.

QB stores information about clusters in *cluster nodes*. The cluster node is defined as $c = (I, \mathbf{h}, n)$ where I is the list of the integer indices $i = 1 \dots N$ of the tracks in that cluster, n is the number of tracks in the cluster, and $\mathbf{h}$ is the *track sum*. $\mathbf{h}$ is a $K \times 3$ matrix which can be updated online when a track is added to a cluster and is equal to:

$$\mathbf{h} = \sum_{i=1}^{n} \mathbf{s}_i \tag{1}$$

where $\mathbf{s}_i$ is the $K \times 3$ matrix representing track i, Σ here represents matrix addition, and n is the number of tracks in the cluster. One summary of the cluster node is the centroid or *virtual* track $\mathbf{v}$ where:

$$\mathbf{v} = \mathbf{h}/n \tag{2}$$

Input: $T = \{\mathbf{s}_1, ..., \mathbf{s}_i, ..., \mathbf{s}_N\}$, θ
Output: $C = [c_1, ..., c_k, ..., c_M]$

```
# create first cluster
c_1 ← ([1], s_1, 1)
C ← [c_1]
M ← 1
for i = 2 to N do
    t ← T_i
    alld ← infinity(M) # distance buffer
    flip ← zeros(M) # flipping check buffer
    for k = 1 to M do
        v ← c_k.h / c_k.n
        d ← d_direct(t, v)
        f ← d_flipped(t, v)
        if f < d then
            d ← f
            flip_k ← 1
        end if
        alld_k ← d
    end for
    m ← min(alld)
    l ← argmin(alld)
    if m < θ then
        # append to current cluster
        if flip_l is 1 then
            c_l.h ← c_l.h + reverse(t)
        else
            c_l.h ← c_l.h + t
        end if
        c_l.n ← c_l.n + 1
        append(c_l.I, i)
    else
        # create new cluster
        c_{M+1} ← ([i], t, 1)
        append(C, c_{M+1})
        M ← M + 1
    end if
end for
```

Algorithm 1: QuickBundles

The algorithm proceeds as follows. At any one step in the algorithm we have M clusters. Select the first track $\mathbf{s}_1$ and place it in the first cluster $c_1 \leftarrow (\{1\}, \mathbf{s}_1, 1)$; $M = 1$ at this point. For each remaining track in turn $i = 2 \dots N$: (i) calculate the MDF distance between track $\mathbf{s}_i$ and the virtual tracks $\mathbf{v}_e$ of all the current clusters c_e,

$e = 1 \ldots M$, where $\mathbf{v}$ is defined on the fly as $\mathbf{v} = \mathbf{h}/n$; (iii) if any of the MDF values m_e are smaller than a distance threshold θ, add track i to the cluster e with the minimum value for m_e; $c_e = (I, \mathbf{h}, n)$, and update $c_e \leftarrow (\mathbf{append}(I, i), \mathbf{h} + \mathbf{s}, n + 1)$; otherwise create a new cluster $c_{M+1} \leftarrow ([i], \mathbf{s}_i, 1)$, $M \leftarrow M + 1$.

Popular amalgamation clustering algorithms such as k-means [27], [28] require iterative reassignment of items to different clusters and hence do not run in linear time. Others such as BIRCH [29] have linear time performance while imposing limits on the size and spread of clusters; BIRCH clusters may be contiguous and the BIRCH algorithm requires further phases of cluster amalgamation to optimise search time when adding future items. By contrast when QB assigns an item to a node it stays there, and clusters are not amalgamated.

Choice of orientation can become an issue when using the MDF distance and adding tracks together, because the diffusion signal is symmetric around the origin, and therefore the $K \times 3$ track can equivalently have its points ordered $1 \ldots K$ or be flipped with order $K \ldots 1$; the diffusion signal does not allow us to distinguish betweeen these two directions. A step in QB takes account of the possibility of needing to perform a flip of a track before adding it to a representative track according to which direction produced the MDF value.

The complete QB algorithm is described in formal detail in Algorithm 1 and a simple step by step visual example is given in Fig. 10. One of the reasons why QB has on average linear time complexity derives from the structure of the cluster node: we only save the sum of current tracks $\mathbf{h}$ in the cluster and the sum is cumulative; moreover there is no recalculation of clusters, the tracks are passed through only once and a track is assigned to one cluster only.

B. Track distances and preprocessing

A wide range of approaches have been taken in the literature for representing or coding for tractographies. The approach we have taken with track coding has gone in parallel with the selection of appropriate metrics for inter-track distances. Numerous inter-track distance metrics have been proposed [30], [31], [4]. The most common is the Hausdorff distance [13, and many other studies]. There are two primary disadvantages of this metric: (1) it ignores the sequential nature of the tracks and treats each track simply as a cloud of points, and (2) its computation requires every point on the first track to be compared with every point on the second track, and vice versa. For these reasons we have opted to use a very simple symmetric distance [26], [1] which we call Minimum average Direct-Flip (MDF) distance $MDF(\mathbf{s}, \mathbf{s}')$ between track $\mathbf{s}$ and track $\mathbf{s}'$, see Eq. (3). This distance can be applied only when both tracks have the same number of points. Therefore we assume that an initial downscaling of tracks has been implemented, where all segments on a track have approximately the same length, and all tracks have the same number of points K, and segments $K - 1$, which are much less than the number of points or segments in the typical raw track. Under this assumption MDF is defined as:

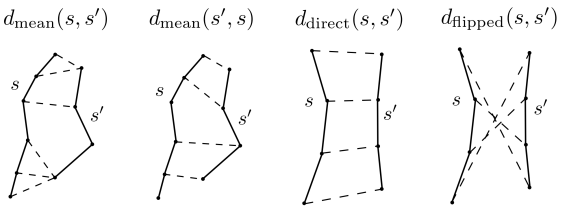

Figure 2. The principal distance used in this work is minimum average direct flip distance MDF $= \min(d_{\text{direct}}, d_{\text{flipped}})$ which is a symmetric distance that can deal with the track bi-directionality problem; it works on tracks which have the same number of points. Another distance we use is the mean average distance which is again symmetric but does not require the tracks to have the same number of points: $\text{MAM}_{\text{mean}} = (d_{mean}(\mathbf{s}, \mathbf{s}') + d_{mean}(\mathbf{s}', \mathbf{s}))/2$ (see Eq. (A.4)). In this figure the components of both distances are shown; the tracks are drawn with solid lines, and then with dashed lines we connect the pairs of points of the two tracks whose distances contribute to the overall metric. Note that we cannot calculate the MDF between the tracks on the left of the figure because they have different numbers of points.

$$
\begin{aligned}
\text{MDF}(\mathbf{s}, \mathbf{s}') &= \min(d_{\text{direct}}(\mathbf{s}, \mathbf{s}'), d_{\text{flipped}}(\mathbf{s}, \mathbf{s}')) \quad (3)\\
d_{\text{direct}}(\mathbf{s}, \mathbf{s}') &= \frac{1}{K} \sum_{i=1}^{K} |\mathbf{x}_i - \mathbf{x}'_i|, \text{ and}\\
d_{\text{flipped}}(\mathbf{s}, \mathbf{s}') &= \frac{1}{K} \sum_{i=1}^{K} |\mathbf{x}_i - \mathbf{x}'_{K-i}|.
\end{aligned}
$$

Here K is the number of points $\mathbf{x}_i$ and $\mathbf{x}'_i$ on the two tracks $\mathbf{s}$ and $\mathbf{s}'$ and $|\mathbf{x} - \mathbf{x}'|$ denotes the euclidean distance between two points $\mathbf{x}$ and $\mathbf{x}'$.

The main advantages of the MDF distance are that it is fast to compute, it takes account of track direction issues through consideration of both direct and flipped tracks, and that its behaviour is easy to understand, from the simplest case of parallel equi-length tracks to the most complicated with very divergent tracks. Another advantage is that it separates short tracks from long tracks; a track $\mathbf{s}$ that is half the length of track $\mathbf{s}'$ will be relatively poorly matched on MDF to $\mathbf{s}'$.

Another important advantage of having tracks with the same number of points is that we can easily do pairwise calculations on them; for example add two or more tracks together to create a new average track. We saw in the previous section how track addition is a key property that we exploit in the QB clustering algorithm.

Care needs to be given to choosing the number of points required in a track (track downsampling). We always keep the endpoints intact and then downsample in equidistant segments. One consequence of short tracks having the same number of points as long tracks is that more of the curvature information from the long tracks is lost relative to the short tracks i.e. the short tracks will have higher resolution. We found empirically that this is

not an important issue and that for clustering purposes even downsampling to only $K = 3$ points in total can be useful [26]. Depending on the application, more or fewer points can be used. In the results presented in often use $K = 12$ which is a good trade-off between track resolution and dimensionality reduction.

C. Exemplar tracks

The virtual tracks created by QB have very nice properties as they represent an average track which can stand as the most characteristic feature of the cluster that they belong to. However, now that we have segmented our tractography into small bundles, we can calculate many more potentially important descriptors for the cluster. One of the most useful approaches is the calculation of exemplars.

Here the idea is to identify an actual track belonging to the tractography which corresponds in some way to the virtual track. In other words to find an exemplar (medoid) track. Virtual tracks do not necessarily coincide with real tracks as they are just the outcome of large amalgamations. There are many strategies for how to select good exemplars for the bundles. A very fast procedure that we use in our work is to find which real track from the cluster c is closest (by MDF distance) to the virtual track $\mathbf{v}$. We will call this exemplar track $\mathbf{e}_1$, i.e. $\mathbf{e}_1 = \arg\min_{\mathbf{s}\in c} \text{MDF}(\mathbf{v}, \mathbf{s})$. The computational complexity of finding $\mathbf{e}_1$ is still linear in cluster size, and that will be very useful if we have created clusterings with clusters containing more than ~5000 tracks (depending on system memory).

A different exemplar can be defined as the most typical track among all tracks in the bundle, which we denote by $\mathbf{e}_2 = \arg\min_{\mathbf{s}\in c} \sum_{s'\in c} \text{MDF}(\mathbf{s}', \mathbf{s})$, or if we want to work with tracks with possibly different numbers of points we could instead use $\mathbf{e}_3 = \arg\min_{\mathbf{s}\in c} \sum_{\mathbf{s}'\in c} \text{MAM}(\mathbf{s}', \mathbf{s})$. Here MAM can any one of the Hausdorff metrics (A). Identification of exemplar tracks of type $\mathbf{e}_2$ and $\mathbf{e}_3$ will be efficient only for small bundles of less than 5000 tracks because we need to calculate all pairwise distances in the bundle. We show below many applications of the exemplars.

In summary, a virtual (centroid) track is the average of all tracks in the cluster. We call it virtual because it does not really exist in the real data set, and to distinguish it from exemplar (medoid) tracks which are again descriptors of the cluster but are represented by real tracks.

D. Bundle Adjacency Comparisons

We have found rather few systematic ways to compare different clustering results for tractographies in the literature [32]. Being able to compare results of clusterings is crucial for creating stable brain imaging procedures, and therefore it is necessary to develop a way to compare different clusterings of the same subject or different subjects. Although we recognise that this is a difficult problem, we propose the following approach with a novel metric which we call Bundle Adjacency (BA). This metric works as follows. Let us assume that we have gathered the exemplar tracks from clustering A in $E_A = [\mathbf{e}_1, ..., \mathbf{e}_{M_A}]$ and from clustering B in $E_B = [\mathbf{e}'_1, ..., \mathbf{e}'_{M_B}]$ where M_A , M_B denote the number of exemplar tracks of the two clusterings. M_A does not need to be equal to M_B. Next we calculate all pairwise MDF distances between the two sets and store them in rectangular matrix D_{AB}. The mimima of the rows of D_{AB} provide the distance to the nearest track in B of each track in A ($m_{A\to B}$) and similarly the minima of the columns of D_{AB} the distance to the nearest track in A of each track in B ($m_{B\to A}$). From these correspondences we only keep those distances that are smaller than a tight threshold θ. Then we define BA to be

$$BA = \frac{1}{2}\left(\frac{|m_{A\to B} \leq \theta|}{M_A} + \frac{|m_{B\to A} \leq \theta|}{M_B}\right) \qquad (4)$$

where $|m_{A\to B} \leq \theta|$ denotes the number of exemplars from A which had a neighbour in B that is closer than θ and similarly for $|m_{B\to A} \leq \theta|$. In other words, BA is mean of the fraction of row minima of D_{AB} that are less than θ and the fraction of column minima less than θ. $BA = 0$ when every exemplar from the one set is further than θ to all exemplars in the other set. $BA = 1$ when all exemplars from one set have a close neighbour in the other set. This metric is very useful especially when comparing tractographies from different subjects because it does not require $M_A = M_B$.

E. Merging two sets of bundles

We can merge bundles using exemplar tracks or virtual tracks. We first set a distance threshold θ usually the same as the one we used for the QBs in the previous step. Assume now that we have gathered the virtual tracks from clustering A in $V_A = [\mathbf{v}_1, ..., \mathbf{v}_{M_A}]$ and from clustering B in $V_B = [\mathbf{v}'_1, ..., \mathbf{v}'_{M_B}]$ where M_A and M_B denote the number of virtual tracks of each clustering. M_A can be different from M_B. (1) For every $\mathbf{v}'_i$ in V_B we find the closest $\mathbf{v}_j$ in V_A and store the distance between these two tracks. Therefore we now have a set of minimum distances from V_B to V_A. The size of this set is equal to $M_B|$. (2) We merge those clusters from B whose virtual tracks have minimum distances (e.g. MDF) smaller than θ into the corresponding clusters of A, and if a virtual track in V_B has no sub-threshold neighbour in V_A then its cluster becomes a new cluster in the final clustering. In that way clusters from the two sets ho have very similar features will merge together, and, if not, new clusters will be created, and we will not have any loss of information from the two sets of clusters.

F. Data sets

We applied QuickBundles to a variety of data sets: simulations, 10 human tractographies collected and processed by ourselves, and one tractography with segmented bundles which was available online.

Simulated trajectories. We generated 3 different bundles of parametric paths sampled at 200 points. The tracks were made from different combinations of sinusoidal and helicoidal functions. Each bundle contained 150 tracks. For the red bundle in Fig. 4 a pencil of helical tracks all starting at the same point on a cylinder was generated by linearly varying the pitch of the helices; the green bundle was made up from a divergent pencil of rays on a sinusoidally corrugated sheet; the blue bundle is similarly made from a divergent rays on a sinsusoidally corrugated sheet, with the rays undergoing sinusoidal modulated lateral bending over a range of amplitudes.

Human subjects. We collected data from 10 healthy subjects at the Medical Research Council Cognition and Brain Sciences Unit 3 Tesla scanner (TIM Trio, Siemens), using Siemens advanced diffusion work-in-progress sequence, and STEAM [33], [34] as the diffusion preparation method. The field of view was $240 \times 240\,\mathrm{mm}^2$, matrix size 96×96, and slice thickness 2.5 mm (no gap). 55 slices were acquired to achieve full brain coverage, and the voxel resolution was $2.5 \times 2.5 \times 2.5\,\mathrm{mm}^3$. A 102-point half grid acquisition [35] with a maximum b-value of $4000\,\mathrm{s/mm}^2$ was used. The total acquisition time was $14' 21''$ with TR=8200 ms and TE=69 ms. The experiment was approved by the Cambridge Psychology Research Ethics Committee (CPREC).

For the reconstruction of the 10 human data sets we used Generalized Q-samping [35] with diffusion sampling length 1.2 and for the tractography propagation we used EuDX (Euler integration with trilinear interpolation, [36]) with 10^6 random seeds, angular threshold 60°, total weighting 0.5, propagation step size 0.5 and anisotropy stopping threshold 0.0239 (see Fig. 6 and Fig. 9).

PBC human subjects. We also used a labelled data sets (see Fig. 1 and 5), from the freely available tractography database used in the Pittsburgh Brain Competition Fall 2009, ICDM (pbc.lrdc.pitt.edu).

III. Results

In this section we first justify our claims about the speed and linear complexity of QB (III-A). Next we demonstrate the robustness of QB as a method for clustering tractographies. In III-C we illustrate the high level of clarity and simplification that the QB representation can bring to a tractography, and how adjustment of the distance threshold θ can be used to merge or split clusters. In III-D we show how the QB representations for a group of subjects can be mapped into a common space and then reclustered to generate a multi-subject atlas of white matter tractographies. We show moreover how the resulting atlas can be used to access clusters in the individual tractographies that have a good correspondence bewteen subjects. Both the speed of performance and the data reduction provided by QB make this innovation a practical research tool. Tractography clustering algorithms have not to date been able to work on full tractography data sets. We show in III-E how using the QB representation of the tractography as input to higher complexity clustering methods brings full tractographies within their effective range. We indicate (III-F) exemplars tracks from the QB representation can improve on interaction with tractographies amd can be used for the direct registration of tractographies (III-G). Finally, quality control of tractographies is an important issue and we show how QB can be used as a fast tool to identify clusters containing long or short tracks (III-H).

A. Complexity and timings

To apply QB to a data set we need to specify three key parameters: K, the fixed number of downsampled points per track; θ the distance threshold, which controls the heterogeneity of clusters; and N the size of the subset of the tractography on which the clustering will be performed. When θ is higher, fewer more heterogeneous clusters are assembled, and conversely when θ is low, more clusters of greater homogeneity are created.

The complexity of QB is in the best case linear time $\mathcal{O}(N)$ with the number of tracks N and worst case $\mathcal{O}(N^2)$ when every cluster contains only one track. The average case is $\mathcal{O}(MN)$ where M is the number of clusters however because M is usually much smaller than N ($M \ll N$) we can neglect M and denote it only as $\mathcal{O}(N)$ as it is common in complexity theory. We created the following experiment to investigate this claim and we found empirically that the average case is actually $\mathcal{O}(N)$ for tractographies (see Fig.3). In this experiment we timed the duration of QB clustering of tractographies containing from 10^5 to 10^6 tracks, with different initial number of points per track (3, 6, 12 and 18) and different QB thresholds (10, 15, 20, 25 mm). These results were obtained using a single thread Intel(R) CPU E5420 at 2.50GHz on a standard PC. The results can be seen in Fig. 3. We observe how the linearity of the QB algorithm with respect to N only reduces slightly even when we use a very low threshold such as 10 mm which can generate many thousand of clusters. This experiment concludes that QB is suitable for fast and linear time clustering.

B. Stability of QB

One of the disadvantages of most clustering algorithms is that they give different results with different initial conditions; for example this is recognised with k-means, expectation-maximization [37] and k-centers [38] where it is common practice to try a number of different random initial configurations. The same holds for QB so if there are not distinct clusters such that the distance

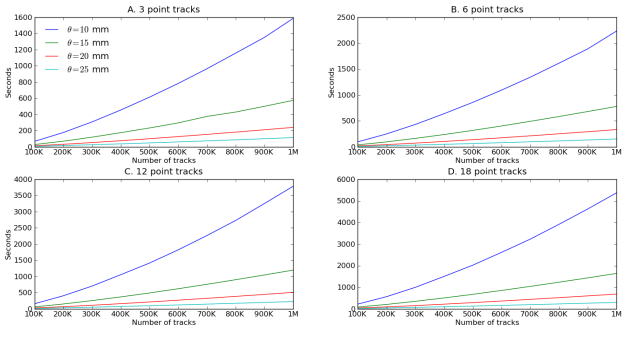

Figure 3. Time comparisons of QB using different number of points per track, different distance thresholds and different number of tracks. QB is a very efficient algorithm whose performance is controlled by just three parameters. (1) the initial downsampling K of the tracks exemplified in four sub-diagrams: 3 points (A), 6 points (B) 12 points (C), 18 points (D). (2) the distance threshold θ in millimeters shown in 4 colours: 10 mm (blue), 15 mm (green), 20 mm (red), 25 mm (cyan). The final parameter, not shown explicitly in these diagrams, is the underlying structure of the data which is expressed by the resulting number of clusters. We used a full tractography to generate these figures without removing or preselecting any parts. Random subsets of the tractography were chosen with size N from 10^5 to 10^6 (x-axis)

between any pair of clusters is supra-threshold, then with different permutations of the same tractography we will typically see similar number of clusters but different underlying clusters. We will examine the robustness of QB in this respect here.

As a first step towards examining the robustness of QB in this respect we recorded the numbers of QB clusters in 20 different random orderings of the tractographies of 10 human subjects acquired as described in section II-F. We removed short tracks shorter than 40 mm and downsampled the tracks at 12 points. Then we applied QB with threshold at 10 mm. The mean number of clusters was 2645.9 (min 1937.6; max 3857.8; s.d. 653.8). There is therefore a considerable between-subject variation in this metric. By contrast the within-subject variability of the number of clusters across random orderings is rather small, with mean standard deviation 12.7 (min 7.3; max 17.4). This suggests an encouraging level of robustness in terms of the numbers of clusters that QB creates.

We ran an experiment were we evaluated BA (4) between pairs of 10 subjects with their tractographies warped in MNI space. This generated 45 BA values with $\theta = 10$ mm. We did this experiment twice; first by keeping only the bundles with more than 10 tracks (BA10) and secondly by keeping only the bundles with more than 100 tracks (BA100). The average value for BA10 was 47% and standard deviation 2.6%. As expected BA100 (bigger landmarks) did better with average value of 53% and standard deviation 4.9%. The difference between BA10 and BA100 is highly significant: Student's t= 4.692, df=88, $p = 1.97 \times 10^{-5}$, two-sided; and, as a precaution against non-normality of the underlying distributions, Mann-Whitney U = 530., $p = 5.65 \times 10^{-5}$. If we think that the small bundles of size < 100 are more idiosyncratic or possibly more likely to reflect noise in the data, whereas larger bundles are more indicative of substantial

Table I
PERFORMANCE TIMINGS FOR QB COMPARED WITH SOME TIMINGS REPORTED IN THE LITERATURE.

Number of tracks (N)	Algorithms	Timings (secs)	QB (secs)	Speedup
1000	[23]	30	0.07	429
60 000	[23]	14 400	14.7	980
400 000	[1]	75 000	160.1	468

structures and landmarks in the tractographies, then we are encouraged to see that on average the virtual tracks of 50% of larger bundles of each tractography lie within 10 mm of those of the other tractographies. This supports the notion that QB can be used to find agreements between different brains by concentrating on the larger (more important) clusters. We will see further evidence for this below (section III-D).

As a further test we compared QB with 12 point tracks and distance threshold at $\theta = 10$ mm versus some timings reported from other state of the art methods found in the literature (Table I). Unfortunately timings were very rarely reported up till now as most algorithms were very slow on full data sets. This experiment was performed using a single CPU core. Nonetheless the speedup that QB offers is obviously of great importance and holds out the prospect of real-time clustering on data sets of fewer than 20 000 tracks (see Table I).

C. The QB Representation

One of the major benefits of applying QB to tractographies is that it can provide meaningful simplifications and find structures that were previously invisible or difficult to locate because of the high density of the tractography. For example we used QB to cluster the corticospinal tract (CST). This bundle was part of the data sets provided by the Pittsburgh Brain Competition (PBC2009-ICDM) and it was selected by an expert. The QB representation is clearly shown in Fig.1 where every cluster is represented by a single virtual track. To generate this clustering we used a tight threshold of 10 mm. We observe that only a few virtual tracks travel the full distance from bottom to top and that they are many tracks that are broken (i.e. shorter than what was initially expected) or highly divergent.

Another interesting feature of QB is that it can be used to merge or split different structures by changing the distance threshold. This is shown in Fig. 4; on the left we see simulated paths made from simple sinusoidal and helicoidal functions packed together. The colour coding is used to distinguish the three different structures. With a lower threshold the three different structures remain separated but when we use a higher threshold the red and blue bundles are represented by only one cluster indicated by the purple virtual track.

Similarly, with the simulations shown in Fig.4 we can see the same effect on real tracks, e.g. those of the fornix shown at the left panel of Fig. 5 where we can obtain different number of clusters at different thresholds. In

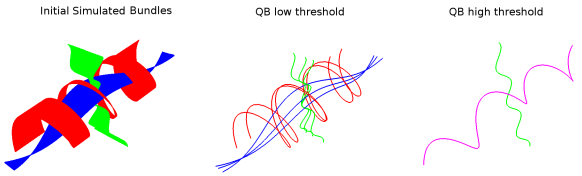

Figure 4. Left: 3 bundles of simulated trajectories; red, blue and green consisting of 150 tracks each. All 450 tracks are clustered together using QB. Middle and Right: virtual tracks using thresholds 1 and 8 respectively. At low threshold the underlying structure is reflected in a more detailed representation. At higher threshold, closer bundles merge together. Here the red and blue bundle have merged together in one cluster represented by the purple virtual track.

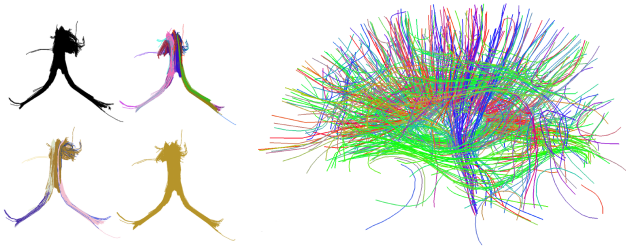

Figure 5. Left: QB clustering of the fornix bundle. The original fornix is shown in black (1076 tracks). All tracks were equidistantly downsampled at 3 points. With a 5 mm threshold QB generates 22 clusters (top right). With 10 mm it generates 7 (bottom left) and with 20 mm the whole fornix is determined by one cluster only (bottom right). Right panel: an example of a full tractography (0.25×10^6 tracks) being clustered using QB with a distance threshold of 10 mm. 763 virtual tracks were produced which is a huge simplification of the initial tractography. Every track shown here represents an entire cluster from 10 to 5000 tracks each. These can be thought as fast access points to explore the entire data set. The colour here encodes track orientation.

that way we can stress thinner or larger sub-bundles inside other bigger bundles.

A full tractography containing 250 000 tracks was clustered using QB with a distance threshold of 10 mm (Fig. 5). We produced a useful reduction of the initial tractography leaving only 763 virtual tracks. Bundles smaller than 10 tracks were removed. Every track shown here represents an entire cluster containing from 10 to 5000 tracks each.

The virtual tracks can be thought as fast access points to explore the entire data set (see Fig. 5). With an appropriate visualization tool we can click on a track and obtain the entire cluster/bundle that it represents. Visualizing an entire data set of that size is impossible on standard graphic cards and most visualization tools e.g. Trackvis (trackvis.org) or DSI Studio (dsi-studio.labsolver.org) can only show a small random sample of the full tractography at real time. In addition we have developed fast and efficient ways of identifying broken or wandering tracks by using the rapid data compression of QB to identify short or erratic virtual tracks.

D. Tractographic atlases and landmarks

We have used QB to construct a robust tractographic atlas in MNI space with data from 10 subjects. First we used the FSL toolbox using steps similar to those in [39] to register each subject's FA volume with the standard FSL template (`FMRIB58`) in MNI space. Next we applied the resulting non-linear transformations to the tractographies. This required technical solutions which are detailed in the DIPY software package.

Having all tractographies in MNI space is especially useful because we can now compare them against available templates or against each other and calculate different statistics. However this is not where we stop; we proceed to generate a tractographic atlas using QB clusterings.

Tractographic Atlases For each subjects, (a) load the warped tractography, (b) downsample the tracks to have only 12 points, (c) calculate and store QB clustering with a 10 mm threshold, (d) merge all clusterings again with a 10 mm threshold as explained in section II-E. When creating an atlas by merging many different subjects the most important issue is what one removes from the atlas as outliers. QB here provides a possible solution for this problem. From the distribution of cluster sizes we find that 20% of the largest clusters had more than 90% of all tracks. This shows that there is much agreement between the biggest bundles of different subjects. We will use this property to create a solid atlas in which we keep the biggest bundles (landmarks) and remove small bundles (outliers).

This atlas has been constructed without considering whether there are inter-subject differences in white matter organisation. Outliers might represent genuine differences in anatomy and the corresponding atlas clusters can be explored to see how many subjects they relate to. With a larger database QB can thus be used to create a probabilistic tractography atlas with which to identify differences in morphology in an analogous manner to that achieved for the cingulate and parasingulate sulci by [40].

Finding and Using Landmarks One can use this atlas or similar atlases created from more subjects in order to select specific structures and study these structures directly in different subjects without using any of the standard ROI based methods.

A simple example is given in Fig. 6. In the first row we see a tractographic atlas joined by merging the QB clusterings of 10 healthy subjects as described in the previous section. Then from these clusters represented by their virtual tracks we keep only 196 biggest clusters i.e. those which contain the highest number of tracks, so that we are sure that there is enough agreement from the different tractographies. From these we pick by way of an example 19 virtual tracks (see Tab. II which correspond to well known bundle structures in the literature:

These 19 tracks are coloured randomly. Then on the

Table II
THE 17 FIBER BUNDLES CHOSEN FOR THE ANALYSIS IN FIG. 6.

Abbreviation	Structure
GCC	genu of corpus callosum
BCC	body of corpus callosum
SCC	splenium
CP	pons cerebellar peduncle
ARC-L	left arcuate fasciculus
ARC-R	right arcuate fasciculus
IFO-L	left inferior occipitofrontal fasciculus
IFO-R	right inferior occipitofrontal fasciculus
FX-R	right fornix
FX-L	left fornix
OR	optic radiation
CGL-L	left cingulum
CGL-R	right cingulum
CST-L	left corticospinal tract
CST-R	right corticospinal tract
UNC-L	left uncinate
UNC-R	right uncinate

second row we show, for the first 6 of these selected representative tracks, the tracks closer than 20 mm from 3 arbitrarily selected subjects. Similarly, on the third row the tracks closer than 15 mm to the next 7 selected tracks. Finally on the last row we bring the tracks from the same 3 subjects which are closer than 18 mm. The colours used for the selected tracks are automatically assigned from the colours assigned to the tracks picked from the atlas. We can see that there is a significant reliability and continuity both within and between subjects even though we have only selected a very small number of representative tracks. Using a similar procedure we could create a list of bundles of every subject and then compare the subjects at the level of bundles.

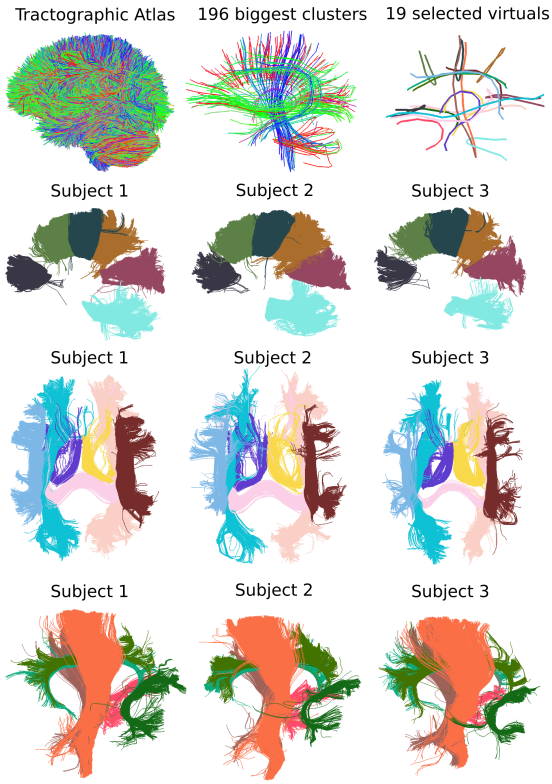

Figure 6. A novel way to do comparisons between subjects. Correspondence between different subjects (last 3 rows) and a few landmarks picked from the tractographic atlas generated by merging QB clusterings of 10 subjects (top row). The fact that we can see this amount of agreement and continuity on the last 3 rows from so few virtual tracks provides a new robust way to statistical comparison between tractographic data sets. (Rows 3 and 4 are from different viewpoints from that in rows 1 and 2.)

E. QB as input to higher complexity methods

We found that QB is of great value as an adjunct to many less efficient algorithms e.g. hierarchical clustering, affinity propagation, nearest neighbours, spectral clustering and other unsupervised and supervised machine learning methods. We present here one example with QB as input to affinity propagation and one with QB as input to hierarchical clustering.

Most clustering algorithms need to calculate all pairwise distances between tracks; that means that for a medium sized tractography of 250 000 tracks we would need 232 GBytes of RAM with single floating point precision. Something which is not and will not be available soon in personal computers. In those cases some people might hope that sparse matrices could provide a nice approximation; however dense tractographies produce very dense distance matrices. The straightforward solution to this problem is to use QB in order to first segment in small clusters and then use the representative tracks (i.e. exemplar or virtual tracks) of these clusters with other higher complexity operations and merge the clusters together in bigger clusters.

The required steps are: (1) cluster each tractography using QB as explained in section III-D); (2) gather up the virtual tracks; (3) calculate MDF distances between the virtual tracks; (4) use any other clustering method to segment this much smaller distance matrix D.

In the left panel of Fig. 7 we show a result where we used hierarchical clustering with single linkage for step (4) with a threshold of 20 mm using the Python software package `hcluster` [41]. A known drawback of single linkage is the so-called chaining phenomenon: clusters may be brought together due to single elements being close to each other, even though many of the elements in each cluster may be very distant to each other. Chaining is usually considered as a disadvantage because it is too driven by local neighbours. Nevertheless, we can take advantage of this property to cluster the entire corpus callosum (CC) together (shown in dark red in left top of Fig. 7) creating a fully automatic CC detection system. Furthermore, we can use different cutting thresholds on the underlying dendrogram to amalgamate together different structures (see e.g. the cingulum bundles in the same panel).

In the right panel of Fig. 7 we see an implementation of step (4) using the more recent algorithm, affinity propagation (AP) [42], which has been identified [22]) as being difficult or impossible to be used for group

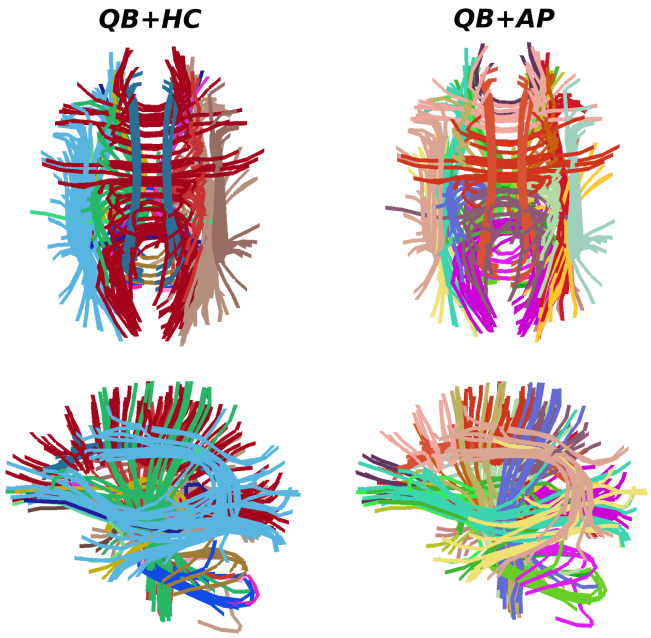

Figure 7. QB was used to cluster an entire set of 10 tractographies together (2 500 000 tracks). The resulting QB representation was used as input to hierarchical clustering (HC) using single linkage (left) and to affinity propagation (AP, right). Colours encode cluster labels. HC results in 19 clusters, AP in 23. QB facilitates significantly the operation of the other two algorithms which would not be able to cluster the entire data sets on current computers. Note the top left panel where QB+HC have managed to cluster the entire CC in one bundle.

analysis or to cluster entire tractographies of many thousands of tracks. Here we see in the bottom right panel of (see Fig. 7) how nicely AP, after the simplification provided by QB, has clustered arcuate, longitudinal occipitofrontal fasciculus and other structures known from the literature. The input of AP was the negative distance matrix $-D$, the preference weights were set to matrix $\texttt{median}(-D)$ and the hierarchical clustering parameter was set to 20 mm. For affinity propagation we used the Python library `scikit-learn`.

While clustering methods of higher order complexity such as HC and AP have considerable theoretical appeal and have a deserved reputation for being able discover meaningful structure in data sets they are simply impracticable when more than a few thousand items are being clustered. Using QB to pre-process the tractography puts the whole data set, condensed to its virtual tracks, within the feasible range of these higher order algorithms.

F. Exemplars replace ROI masks

Medical practitioners and neuroanatomists often argue that when they use multiple spherical or rectangular masks to select some bundles many tracks are thrown away because they are small and the mask operations cannot get hold of them. Our method provides a solution to this problem as it can identify broken or smaller bundles inside other bigger bundles which are otherwise very difficult or even sometimes impossible to identify visually or with the use of masks. Our method attacks this problem and suggests a very efficient and robust solution which sets the limit for unsupervised clustering of tractographies and facilitates tractography exploration and interpretation. The point here is that one can now use exemplar tracks as access points into the full tractography and with a single click on that exemplar track obtain the entire bundle. Therefore a super-bundle can be created just with with a few clicks based on a selection from exemplar tracks.

In order to create this system we implemented a 3D visualization/interaction system for tractographies based on QB in Python and OpenGL. This code is available online at `fos.me`.

G. Direct Tractography Registration

Direct tractography registration is a recently described problem with only a small number of publications [43], [44], [45], [46], [25], [8], [9], [47], and so far as we know there are no publicly available solutions. By direct registration we mean that no other information apart from the tractographies themselves is used to guide the registration. This is in contrast to section III-D) where we used FA registration mappings applied to tractographies which is also most commonly used in the literature along with other Tensor based methods [48].

We now describe our algorithm which is efficient and simple to use, completely automatic and provides an evidently robust direct rigid tractography registration algorithm available in seconds. This algorithm could be of great use when comparing healthy versus severely diseased brains e.g. stroke or vegetative state patients when non-rigid registration is not recommended because of severe asymmetries in the diseased brains. The algorithm is based on the robustness of QB to find good representative descriptors.

Let T_A, and T_B the two tractographies to be aligned in native space. The required steps are: (1) all tracks with length smaller than 100 mm and longer than 300 mm are removed from the data sets. (This will reduce the size of each tractography to about $1/4$ of its initial size i.e. ~200 000 tracks. While all the subjects are adults, this filtering may have different effects depending on brain size. We have not investigated this question at present); (2) downsample both tractographies equidistantly to 12 points; (3) run QB with distance threshold at 10 mm for both tractographies; (4) collect all exemplar tracks from clusters containing more than 0.2% of all tracks; (5) assuming we have these now in E_A and E_B, calculate all pairwise distances $D = \texttt{MDF}(E_A, E_B)$ and save them in rectangular matrix D; (6) use the modified Powell's method [49] to minimize the symmetric distance function SMD $= \sum_i \min_j D(i,j) + \sum_j \min_i D(i,j)$ over rigid rotations of E_B starting with zeroed initial conditions; (7) after each iteration of the optimization, E_B is transformed by the resulting rigid rotation and SMD is recalculated.

To ensure smooth rotations we use the Rodriguez rotation formula.

In Fig.8 A we see the result of this algorithm applied to two tractographies represented by their exemplar tracks, depicted in orange and purple. We can see in the upper panel that the orange tractography is misaligned with respect to the purple one, and in the lower panel we see their improved alignment after applying our algorithm.

Metric. SMD is proposed here for registration of trajectory data sets, but one could equally use mutual information [50] or the correlation ratio [51] for registration of volumetric data sets. Nonetheless, the advantage of SMD is that it comes from robust landmarks generated by QB which bring together local and global components. Initially, it was not clear if we should use SMD or just the sum of all distances $\mathrm{SD} = \sum_{i,j} D(i,j)$. Therefore, we made a small experiment to validate the smoothness and convexity of these two cost functions. We plotted both functions under a single-axis translation or a single-angle rotation of the same tractography as show in Fig. 8 B and C. From, these two diagrams we can see that although for translations only the SD was entirely convex, with rotations the SD had stronger local minima which is not a good property for registration. Furthermore, the SMD had steeper gradients towards the global minimum which is a positive indicator for faster convergence.

Experiments. The first large scale experiment took place using the same tractography of a single individual copied and transformed 1000 times with range of all three angles from -45° to 45° and range of all x,y,z translations from -113 to 113 mm. Then we registered all transformed tractographies to the static one and calculated all pairwise MDF distances storing them in a square matrix D. We would expect that if the registration was correct then the sum of all diagonals elements of D would be close to 0. This was confirmed with both cost functions used SD and SMD getting close to zero 99.8% of the time however SMD was always closer to perfect alignment than SD. Consequently we chose SMD as a better cost function for direct tractography registration.

We used GQI-based tractographies from 10 subjects, and registered all 45 possible pairs. Comparing different tractographies is not a trivial problem however we can use the bundle adjacency (BA) metric explained in section II-D. We can report that the mean initial BA was $34.8\% \pm 8.0\%$ and the mean final BA after applying our direct registration method was $48.1\% \pm 6.1\%$. This was a statistically highly significant improvement ($t_{\mathrm{paired}}(44) = 11.2$, $p \leq 10^{-13}$). These BA values are comparable to those reported in section II-D. We are planning in the future to compare this registration method against other standard methods which are common in the literature.

It should be noted that both BA and SMD are derived from the same distance matrix D_{AB} between the two sets of virtual tracks. However, using BA to assess a registration that has minimised SMD is less circular than might appear at first sight; the minimization of SMD depends on all the entries of D as it is the mean of all those entries, while BA depends on the separate minimizations of the rows and columns of D.

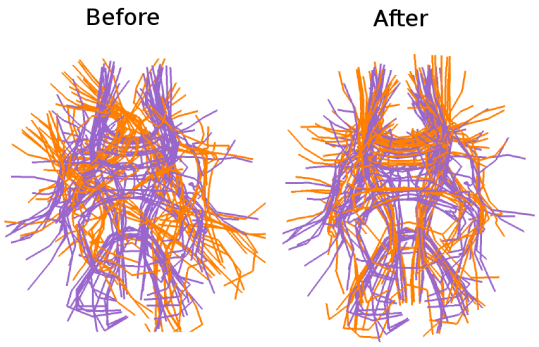

Figure 8. Tractographies from two different subjects before (left) and after (right) QB-based direct registration.

H. Quality Control

We have not so far taken account of short tracks in this report. That is perfectly valid because (a) the longer tracks are more likely to be used as useful landmarks when comparing or registering different subjects because it is more likely for them to exist in most subjects, (b) removing short tracks facilitates the usage of distance based clustering (no need for manually setting the distance threshold) and interaction with the tractography, (c) typically one first wants to see the overall representation of the tractography and later go to the details. Nonetheless, after having clustered the longer tracks there are many ways to assign the smaller bundles to their closest longer bundles. For this purpose we recommend to use a different distance from MDF for example the minimum version of MAM referred to as $\mathrm{MAM}_{\mathrm{min}}$ in Eq. (A.1).

Here we show some simple strategies for clustering short fibres. The first is for unsupervised clustering and the second one is for supervised learning.

1. Cluster the long tracks using QB with distance threshold at 10 mm and then cluster the short tracks (<100 mm) to a lower threshold and assign them to their closest long track bundle from the first clustering using the MAM_{min} distance.

2. Read the tractography of a single subject, use a tractographic atlas as the one created in section III-D and pick one or more close virtual tracks from that atlas and then find the closest tracks from the subject to that selected track using MDF, cluster the closest tracks found from the previous step and for each one of these new virtual tracks find the closest tracks using the $\mathrm{MAM}_{\mathrm{min}}$ distance. We should now have an amalgamation of shorter and longer tracks in one cluster.

An example of this second strategy is shown in Fig. 9. **A**: a track of interest from the arcuate fasciculus is selected from the tractographic atlas shown in Fig. 6

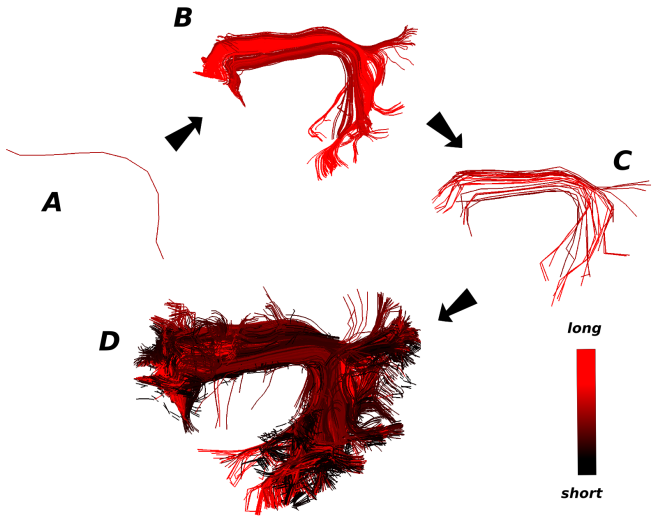

Figure 9. A simple and vigorous strategy for handling short and long tracks together by picking a track of interest from one of our atlases. Colour map here encodes track length. A: a single selected atlas track, B: for a single subject, 245 actual tracks closer than 15 mm (MDF distance), C: the tracks from B clustered with 23 virtual tracks using QuickBundles, D: 3421 actual tracks closer than 6 mm (MAM_{min} distance) from the virtual tracks in C are shown. We can see that a great number of short tracks have been brought together along with the tracks in B. In this way we managed to bring together in an automatic fashion an entire bundle consisting both of long and short tracks by just selecting a single track.

(top row, centre); **B**: the tracks of the subject closer than 15 mm (MDF) from the selected cluster are shown and clustered with a distance threshold of 6.25 mm in **C**; **D**: from every virtual track in **C** we find the closest tracks using the MAM_{min} distance from the entire tractography.

IV. Discussion and conclusion

We have presented a novel and powerful algorithm – QuickBundles (QB). This algorithm provides simplifications to the old problem of revealing the detailed anatomy of the densely packed white matter which has recently attracted much scientific attention; it can also be used for any trajectory clustering problem and it is recommended when large data sets are involved. QB can be used with all types of diffusion MRI tractographies which generate streamlines (e.g. probabilistic or deterministic) and it is independent of the reconstruction model.

In common with mainstream clustering algorithms such as k-means, k-centers and expectation maximization, QB is not a global clustering method therefore it can give different results under different initial conditions of the data set when there is no obvious distance threshold which can separate the clusters into meaningful bundles; for example we should expect different clusters under different permutations/orderings of the tracks in a densely packed tractography. However, we found that there is enough agreement even between two clusterings of the same tractography with different orderings. If the clusters are truly separable by distances then there is a global solution independent of orderings. This is often visible in smaller subsets of the initial tractography. We empirically found that this problem is minimized even with real data sets when a low distance threshold of about $10-20$ mm is used.

Furthermore the output of QB can be used as the input to another recent fast algorithm of quadratic time on average $\mathcal{O}(M^2)$ called affinity propagation where now $M \ll N$ therefore the overall time stays linear on the number of tracks N. Other algorithms previously too slow to be used on the entire tractography can now be used efficiently too e.g. spectral and hierarchical clustering.

We saw that QB is a linear time clustering method based on track distances, which is on average linear time $\mathcal{O}(N)$ where N is the number of tracks and with worst case $\mathcal{O}(N^2)$ when every track is a singleton cluster itself. Therefore QB is the fastest known tractography clustering method and even real-time on smaller tractographies (<20 000 tracks). We also showed that is uses a negligible amount of memory.

QB is fully automatic and very robust as when we use it we can find good agreements even between different subjects and can be used to create tractography atlases at high speed. Additionally, it can be used to explore multiple tractographies and find correspondences between tractographies, create landmarks used for registration or population comparisons.

QB can be used as well for reducing the dimensionality of the data sets at the time of interaction providing an alternative way to ROIs using BOIs (bundles of interest). We also showed that it can be used to find obscured tracks not visible to the user at first instance. Therefore QB opens up the road to create rapid tools for exploring tractographies of any size.

We have shown results with data from simulations, single and multiple real subjects. The code for QuickBundles is freely available at `dipy.org`.

References

[1] E. Visser, E. H. J. Nijhuis, J. K. Buitelaar, and M. P. Zwiers, "Partition-based mass clustering of tractography streamlines." *NeuroImage*, Jul. 2010.

[2] G. Gerig, S. Gouttard, and I. Corouge, "Analysis of brain white matter via fiber tract modeling," in *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, vol. 2. IEEE, 2004, pp. 4421–4424.

[3] P. Guevara, C. Poupon, D. Rivière, Y. Cointepas, M. Descoteaux, B. Thirion, and J.-F. Mangin, "Robust clustering of massive tractography datasets." *NeuroImage*, vol. 54, no. 3, pp. 1975–1993, Oct. 2010.

[4] S. Zhang and D. Laidlaw, "DTI fiber clustering and cross-subject cluster analysis," in *Int. Soc. Magn. Reson. Med*, 2005.

[5] R. Jianu, C. Demiralp, and D. Laidlaw, "Exploring 3D DTI fiber tracts with linked 2D representations," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 1449–1456, 2009.

[6] V. El Kouby, Y. Cointepas, C. Poupon, D. Rivière, N. Golestani, J. B. Poline, D. Le Bihan, and J. F. Mangin, "MR diffusion-based inference of a fiber bundle model from a population of subjects." *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 8, no. Pt 1, pp. 196–204, Jan. 2005.

[7] A. Tsai, C. Westin, A. Hero, and A. Willsky, "Fiber tract clustering on manifolds with dual rooted-graphs," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–6.

[8] O. Zvitia, A. Mayer, and H. Greenspan, "Adaptive mean-shift registration of white matter tractographies," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*. IEEE, 2008, pp. 692–695.

[9] O. Zvitia, A. Mayer, R. Shadmi, S. Miron, and H. K. Greenspan, "Co-registration of white matter tractographies by adaptive-mean-shift and Gaussian mixture modeling." *IEEE transactions on medical imaging*, vol. 29, no. 1, pp. 132–45, Jan. 2010.

[10] A. Brun, H. Knutsson, H. Park, M. Shenton, and C. Westin, "Clustering fiber traces using normalized cuts," *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2004*, pp. 368–375, 2004.

[11] Z. Ding, J. C. Gore, and A. W. Anderson, "Classification and quantification of neuronal fiber pathways using diffusion tensor MRI." *Magnetic Resonance in Medicine*, vol. 49, no. 4, pp. 716–21, 2003.

[12] I. Corouge, S. Gouttard, and G. Gerig, "A statistical shape model of individual fiber tracts extracted from diffusion Tensor MRI," *Medical Image Computing and Computer Assisted Intervention*, pp. 671–679, 2004.

[13] ——, "Towards a shape model of white matter fiber bundles using diffusion tensor MRI," in *International Symposium on Biomedical Imaging*, 2004, pp. 344–347.

[14] I. Corouge, P. T. Fletcher, S. Joshi, S. Gouttard, and G. Gerig, "Fiber tract-oriented statistics for quantitative diffusion tensor MRI analysis." *Medical image analysis*, vol. 10, no. 5, pp. 786–98, 2006.

[15] M. Maddah, A. U. Mewes, S. Haker, W. E. Grimson, and S. K. Warfield, "Automated atlas-based clustering of white matter fiber tracts from DTMRI." *International Conference in Medical Image Computing and Computer Assisted Intervention*, vol. 8, no. Pt 1, pp. 188–95, 2005.

[16] M. Maddah, W. Grimson, and S. Warfield, "Statistical modeling and EM clustering of white matter fiber tracts," in *IEEE Int. Symp. Biomedical Imaging*. Citeseer, 2006, pp. 53–56.

[17] M. Maddah, L. Zollei, W. Grimson, C. Westin, and W. Wells, "A mathematical framework for incorporating anatomical knowledge in dt-mri analysis," in *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE, 2008, pp. 105–108.

[18] U. Ziyan, M. Sabuncu, W. Grimson, and C. Westin, "Consistency clustering: a robust algorithm for group-wise registration, segmentation and automatic atlas construction in diffusion MRI," *International Journal of Computer Vision*, vol. 85, no. 3, pp. 279–290, 2009.

[19] L. Jonasson, P. Hagmann, J. Thiran, and V. Wedeen, "Fiber tracts of high angular resolution diffusion MRI are easily segmented with spectral clustering." in *Int. Soc. Mag. Reson. Med*, 2005, p. 1310.

[20] L. J. O'Donnell and C. F. Westin, "Automatic tractography segmentation using a high-dimensional white matter atlas." *IEEE Trans Med Imaging*, vol. 26, no. 11, pp. 1562–75, 2007.

[21] A. Leemans and D. Jones, "A new approach to fully automated fiber tract clustering using affinity propagation," in *Int. Symp. on Magnetic Resonance in Medicine (ISMRM)*, vol. 17, 2009, p. 856.

[22] J. Malcolm, M. Shenton, and Y. Rathi, "Filtered Tractography: State estimation in a constrained subspace," *Proceedings of MICCAI (Diffusion Modeling and Fiber Cup)*, pp. 122–133, 2009.

[23] X. Wang, W. Grimson, and C. Westin, "Tractography segmentation using a hierarchical Dirichlet processes mixture model," *NeuroImage*, 2010.

[24] S. Durrleman, P. Fillard, X. Pennec, A. Trouvé, and N. Ayache, "A statistical model of white matter fiber bundles based on currents." *Proceedings of the 21st Conference on Information Processing in Medical Imaging*, vol. 21, pp. 114–25, Jan. 2009.

[25] S. Durrleman, P. Fillard, X. Pennec, A. Trouvé, and N. Ayache, "Registration, atlas estimation and variability analysis of white matter fiber bundles modeled as currents," *NeuroImage*, 2010.

[26] E. Garyfallidis, M. Brett, and I. Nimmo-Smith, "Fast Dimensionality Reduction for Brain Tractography Clustering," *16th Annual Meeting of the Organization for Human Brain Mapping*, 2010.

[27] H. Steinhaus, "Sur la division des corp materiels en parties," *Bull. Acad. Polon. Sci*, vol. 1, pp. 801–804, 1956.

[28] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1. California, USA, 1967, pp. 281–297.

[29] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: A new data clustering algorithm and its applications," *Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 141–182, 1997.

[30] Z. Ding, J. Gore, and A. Anderson, "Classification and quantification of neuronal fiber pathways using diffusion tensor MRI," *Magn. Reson. Med.*, vol. 49, pp. 716–721, 2003.

[31] M. Maddah, W. M. Wells, 3rd, S. K. Warfield, C. F. Westin, and W. E. Grimson, "Probabilistic clustering and quantitative analysis of white matter fiber tracts." *Inf Process Med Imaging*, vol. 20, pp. 372–83, 2007.

[32] B. Moberts, A. Vilanova, and J. van Wijk, "Evaluation of fiber clustering methods for diffusion tensor imaging," in *IEEE Visualization*, 2005, pp. 65–72.

[33] K. Merboldt, W. Hanicke, H. Bruhn, M. Gyngell, and J. Frahm, "Diffusion imaging of the human brain in vivo using high-speed STEAM MRI," *Magnetic Resonance in Medicine*, vol. 23, no. 1, pp. 179–192, 1992.

[34] M. A. Bernstein, K. F. King, and X. J. Zhou, *Handbook of MRI Pulse Sequences*. Elsevier Academic Press, 2004.

[35] F. Yeh, V. Wedeen, and W. Tseng, "Generalized Q-sampling imaging," *IEEE Transactions on Medical Imaging*, vol. 29, no. 9, pp. 1626–1635, Mar. 2010.

[36] E. Garyfallidis, "Towards an accurate brain tractography," Ph.D. dissertation, University of Cambridge, 2012.

[37] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[38] T. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, pp. 293–306, 1985.

[39] S. M. Smith, M. Jenkinson, H. Johansen-Berg, D. Rueckert, T. E. Nichols, C. E. Mackay, K. E. Watkins, O. Ciccarelli, Z. Cader, P. M. Matthews, and T. E. Behrens, "Tract-based spatial statistics: Voxelwise analysis of multi-subject diffusion data," *NeuroImage*, vol. 31, pp. 1487–1505, 2006.

[40] T. Paus, F. Tomaiuolo, N. Otaky, D. MacDonald, M. Petrides, J. Atlas, R. Morris, and A. Evans, "Human cingulate and paracingulate sulci: pattern, variability, asymmetry, and probabilistic map," *Cerebral Cortex*, vol. 6, no. 2, pp. 207–214, 1996.

[41] D. Eads, "hcluster: Hierarchical clustering for scipy," scipy-cluster.googlecode.com, 2008.

[42] D. Dueck, "Affinity propagation: Clustering data by passing messages," Ph.D. dissertation, University of Toronto, 2009.

[43] A. Leemans, J. Sijbers, S. De Backer, E. Vandervliet, and P. Parizel, "Multiscale white matter fiber tract coregistration: A new feature-based approach to align diffusion tensor data," *Magnetic Resonance in Medicine*, vol. 55, no. 6, pp. 1414–1423, 2006.

[44] A. Mayer and H. Greenspan, "Bundles of interest based registration of White Matter tractographies," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*. IEEE, 2008, pp. 919–922.

[45] ——, "Direct registration of white matter tractographies with application to atlas construction," in *MICCAI 2007 Workshop Statistical Registration PairWise and GroupWise Alignment and Atlas Formation*, 2007.

[46] A. Mayer, G. Zimmerman-Moreno, R. Shadmi, A. Batikoff, and H. Greenspan, "A Supervised Framework for the Registration and Segmentation of White Matter Fiber Tracts," *Medical Imaging, IEEE Transactions on*, vol. 30, no. 1, pp. 131–145, 2011.

[47] U. Ziyan, M. R. Sabuncu, L. J. O'Donnell, and C. F. Westin, "Nonlinear registration of diffusion MR images based on fiber bundles." *International Conference in Medical Image Computing and Computer-Assisted Intervention*, vol. 10, no. Pt 1, pp. 351–8, 2007.

[48] A. Goh and R. Vidal, "Algebraic methods for direct and feature based registration of diffusion tensor images," *Computer Vision–ECCV 2006*, pp. 514–525, 2006.

[49] R. Fletcher, "Practical methods of optimization," *John and Sons, Chichester*, 1987.

[50] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *IEEE Transactions on Medical Imaging*, vol. 16, no. 2, pp. 187–198, 1997.

[51] A. Roche, G. Malandain, X. Pennec, and N. Ayache, "The correlation ratio as a new similarity measure for multimodal image registration," *International Conference in Medical Image Computing and Computer-Assisted Intervention*, p. 1115, 1998.

APPENDIX A
HAUSDORFF (MAM) INTER-TRACK DISTANCE

In some stages in the analysis of tractographies we find a use for metrics from the family of Hausdorff distances which for simplicity we denote as MAM distances – short for Minimum, or Maximum, or Mean, Average Minimum distance (MAM). We mostly use the Mean version of this family, see Eq. (A.4) but the others are potentially useful as they can weight different properties of the tracks. These distances are slower to compute but they can work with different number of segments on tracks that is useful for some applications. The equations below show the formulation of these distances:

$$d_{\text{mean}}(s,s') = \frac{1}{K_A}\sum_{i=1}^{K} d(x_i,s'),$$

$$d_{\text{min}}(s,s') = \min_{i=1,\ldots,K} d(x_i,s'), \quad \text{and} \tag{A.1}$$

$$d_{\text{max}}(s,s') = \max_{i=1,\ldots,K} d(x_i,s'), \quad \text{where} \tag{A.2}$$

$$d(x,s') = \min_{j=1,\ldots,K'} |x - x'_j|.$$

$$\text{MAM}_{\text{min}} = \min(d_{\text{mean}}(s,s'), d_{\text{mean}}(s',s)) \tag{A.3}$$

$$\text{MAM}_{\text{max}} = \max(d_{\text{mean}}(s,s'), d_{\text{mean}}(s',s))$$

$$\text{MAM}_{\text{mean}} = (d_{\text{mean}}(s,s') + d_{\text{mean}}(s',s))/2 \tag{A.4}$$

where the number of points $K = \#s$ and $K' = \#s'$ on the two tracks are not necessarily the same. For the same threshold value MAM_{min}, MAM_{max} and MAM_{mean} will give different results. For example, MAM_{min} will bring together more short tracks with long tracks than MAM_{max}, and MAM_{mean} will have an in-between effect. Finally, other distances than the average minimum based on the minimum see Eq. (A.1) or maximum distance see Eq. (A.2) can be used. However, we have not investigated them in this work in relation to clustering algorithms.

APPENDIX B
QB STEP-BY-STEP

Fig. 10 illustrates the operation of QB step by step. Initially in panel (i) 6 unclustered tracks ($A - F$) are presented; imagine that the distance threshold θ used here is the same as the MDF distance (Eq. 3) between B and E: $\theta = \text{MDF}(B,E)$. The algorithm starts and in (ii) we see that track A was selected; as no other clusters exist track A becomes the first cluster (labelled with purple color) and the virtual track of that cluster is identical with A as seen in (iii); next in (iv) track B is selected and we calculate the MDF distance between B and the virtual track of the other clusters. For the moment there is only one cluster to compare so QB calculates MDF (B,virtual-purple) and this is obviously bigger than threshold θ ($\theta = \text{MDF}(B,E)$). Therefore a new cluster is assigned for B, and B becomes the virtual track of that cluster as shown in (v). In (vi) the next track C is selected and this is again far away from both purple and blue virtuals; therefore another cluster is created and C is the virtual of the blue cluster as shown in (vii). In (viii) track D is selected and after we have calculated MDF(D,purple), MDF(D,Blue) and MDF(D,green) it is obvious that D belongs to the purple cluster as MDF(D,purple) is smaller and lower than threshold as shown in (ix). However we can now see in (x) that things change for the purple cluster because the virtual track is not anymore made by only one track but it is the average of D and A shown with a dashed line. In (xi) E is the current track and will be assigned to the green cluster as shown in (xii) because MDF(E,virtual green) = MDF(E,B) = θ, and in (xiii) we see the updated virtual track for the green cluster which is equal to $(B + E)/2$ where '+' means track addition. In (xiv) the last track is picked and compared with the virtual tracks of the other 3 clusters; obviously MDF(F,purple) is the only distance smaller than threshold, and so F is assigned to the purple cluster in (xv). Finally, in (xvi) the virtual purple track is updated as $(D + A + F)/3$. As there are no more tracks to select, the algorithm stops. We can see that all three clusters have been found and all tracks have been assigned successfully to a cluster.

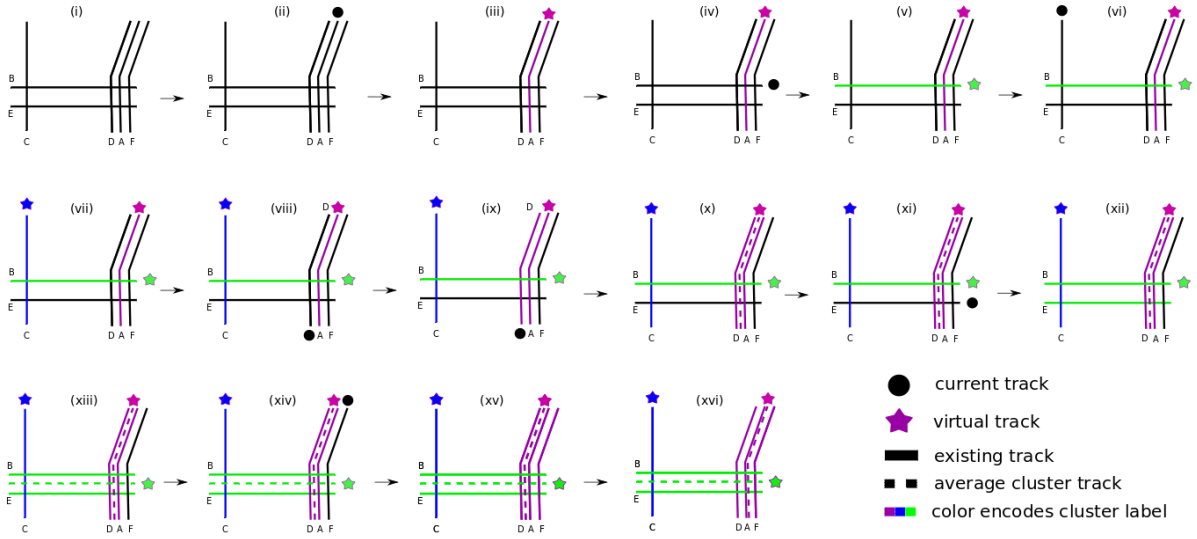

Figure 10. This is a step-by-step explanation of how QuickBundles proceeds when clustering six tracks (top left) resulting in three bundles (blue, red and green, bottom right)