Fast Dimensionality Reduction for Brain Tractography

Eleftherios Garyfallidis[1,3], Matthew Brett[2], Ian Nimmo-Smith[3]

[1]University of Cambridge, Cambridge UK [2]University of California, Berkeley CA
[3]MRC Cognition and Brain Sciences Unit, Cambridge UK

Overview

Algorithms for clustering and classifying diffusion imaging white matter tractographies are typically quite slow. They often needing days or weeks of processing on a single CPU core. Much of the computational burden comes from the need for detailed geometric comparisons between pairs of tracks in large datasets often containing hundreds of thousands of tracks. We have developed an approach that leaves this more detailed comparison to a later stage after an initial first pass through the dataset to create prototype bundles.

The method presented here is a fast method. Less than 5 minutes are required to produce preliminary clusters from a whole brain tractography dataset of ~250,000 tracks.

Our algorithm is inspired by the BIRCH algorithm (Zhang et al. 1996) with a geometrically additive track metric.

When clusters are held in a tree structure this permits upwards amalgamations to form bundles out of clusters, and downwards disaggregation to split clusters into finer sub-clusters corresponding to a lower distance threshold. The algorithm consists of 2 phases.

Split Phase: Select the first track t_1, and place it in the first cluster r_1<-{t_1}. Then for all remaining tracks n <= 2N (where N is the number of tracks):

1: Goto next track t_n.

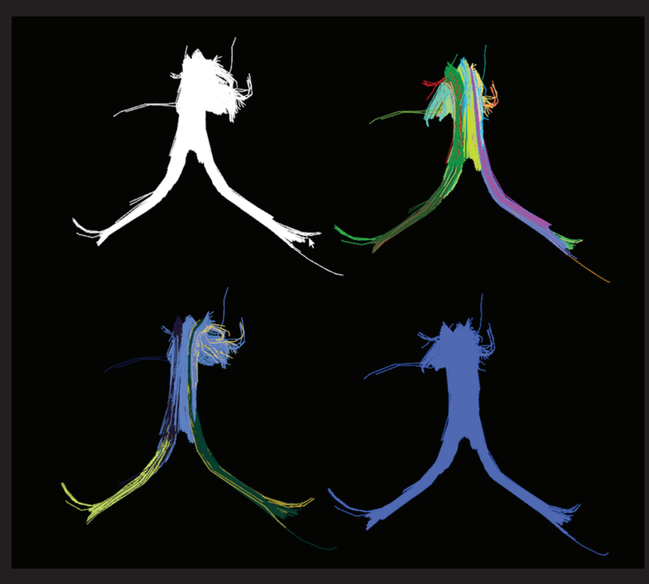
2: Calculate 3TED between this track and virtual tracks of all current clusters r_m (where 1 <= m <= M and M is the current number of clusters).

3: If the minimum 3TED distance is smaller than threshold, add the track to the cluster with the minimum 3TED, and update by joining t_n to r_m; otherwise create a new cluster r_{m+1}<-{t_n}; m <- m+1.

Merge phase: Create a higher node that aggregates nearby clusters by comparing their virtual 3-tracks. The new cluster is the union of the two previous clusters.

Hierarchical Track Clustering Algorithm

The figure shows how the algorithm clustered the *fornix* bundle from the Fall 2009 Pittsburgh Brain Competition (PBC, 2009). The bundle consists of 1076 tracks labeled by the neuroanatomist. [1] all the tracks in white. The rest of the figure shows detected clusters, with tracks in a cluster sharing the same unique color. [2] distance threshold of 5mm. Our method reduces the search space between tracks in large trajectory datasets from tractography. The algorithm has very low computation time and memory use. It may be used for first pass clustering, to reduce the number of detailed comparisons between full track descriptions. Our method is hierarchical; clusters can be split into sub-clusters by decreasing the distance threshold. Making a graph of the cluster structure can be rapidly traversed to look for similarity of clusters across different scales. The results here use only three points (the start, middle and end point). This is not intrinsic to our technique; we can use more points to approximate the tracks, and different distance measures (Corouge 2004; Jianu, 2009), to detect similarity. More detailed approximations consisting of more segments can be used at a later stage. There are 22 clusters. Left and right clusters are distinct. There are different clusters for short and long groups of tracks. The bottom left panel shows the 7 clusters found with a distance threshold of 10mm. The left and right long bundles remain distinct, but the central part of the fornix now has a single main cluster. The bottom right panel shows the single cluster that found with a distance threshold of 20mm. Figure 2 shows the result of our method for the whole brain from the first PBC dataset, consisting of 250K tracks. The left panel shows all the tracks in white. The middle panle shows the 158 clusters that result from the whole brain clustering with a distance threshold of 20mm. There is plausible differentiation between bundles - e.g. the well-differentiated desending cortico-spinal tracks. On the right we show the corresponding virtual 3-tracks. It took around 5 minutes to perform whole-brain clustering on 1 core of a 2.5 GHz Intel PC.



Dimensional Reduction and Track Metrics

Current high definition tractography can produce about 300,000 tracks. A track is usually a line defined as a series of several hundred connected points. To reduce the number of searches in this massive dataset we generate a graph where each node consists of a virtual (representative) low dimensional track, the number of tracks in the cluster and the indices of the tracks in the cluster. This virtual track is the arithmetic mean of all the downsampled tracks in the node. For the downsampling we found we could get useful results by approximating a track with just two connected line segments.

This first pass method is based on the observation that for two tracks to be considered similar at least the start, end, and middle points of the tracks should be close to each other. Each track in the dataset is approximated by a three-point track (two ends and the middle).

Clusters of 3-tracks are built by a fast agglomerative hierarchical clustering algorithm using the 3-track euclidean distance metric 3TED.

As we create clusters, we generate the virtual track (r) for the cluster, given by the centroid of the constituent 3-tracks.

Performance and Conclusion

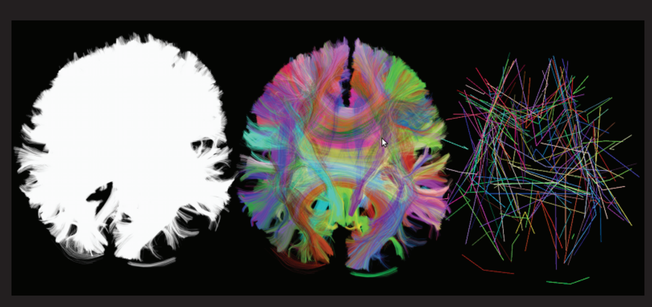
Performance

The algorithm is *online* as it does not all the data to be present at the outset. Instead it steps incrementally through the tracks with clusters automatucally updated when a new track is incorporated. This makes create average track datasets across several brains. Multiresolution is also supported. Fast searches can be done through the tree or graph structure that holds the data with near neighbours linked by the arcs of the graph.

Conclusion

Our method reduces the search space between tracks in large trajectory datasets from tractography. The algorithm makes very low computation time and memory demands. It may be used for first pass clustering, to reduce the number of detailed comparisons between full track descriptions. Our method is hierarchical; clusters can be split into sub-clusters by decreasing the distance threshold. Making a graph of the cluster structure can be rapidly traversed to look for similarity of clusters across different scales.

The results here use only three points (the start, middle and end point). This is not intrinsic to our technique; we can use more points to approximate the tracks, and different distance measures (Corouge 2004; Jianu, 2009), to detect similarity. More detailed approximations consisting of more segments can be used at a later stage.



References

Zhang, T. (1996), 'Birch: An Efficient Data Clustering Method for Very Large Databases', SIGMOD RECORD, vol. 25, no. 2, pp. 103-114. **Corouge, I. (2004)**, 'Towards a shape model of white matter fiber bundles using diffusion tensor MRI', ISBI, pp. 334-347. **Jianu, R. (2009)**, 'Exploring 3D DTI Fiber Tracts with Linked 2D Representations', IEEE Transactions on Visualization and Computer Graphics, vol. 15, no. 6, pp. 1449-1456. **PBC (2009)**, http://pbc.lrdc.pitt.edu

Software

The Python / Cython code for the development of these methods is published as part of the **`dipy`** project, which is hosted at **`http://nipy.org/dipy`**