Machine Learning for Tract Segmentation in dMRI Data

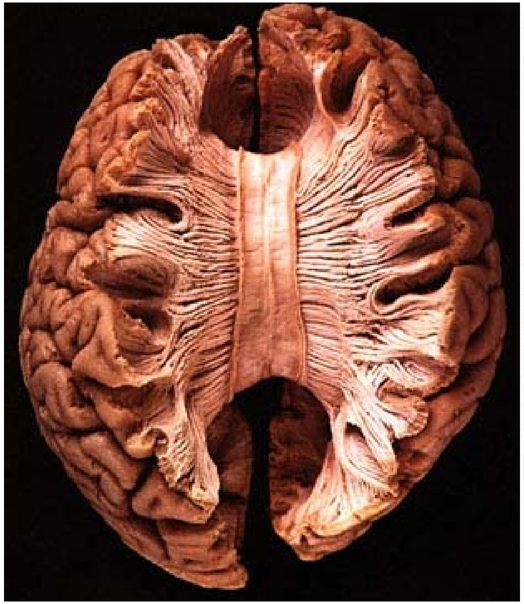
Bao Nguyen

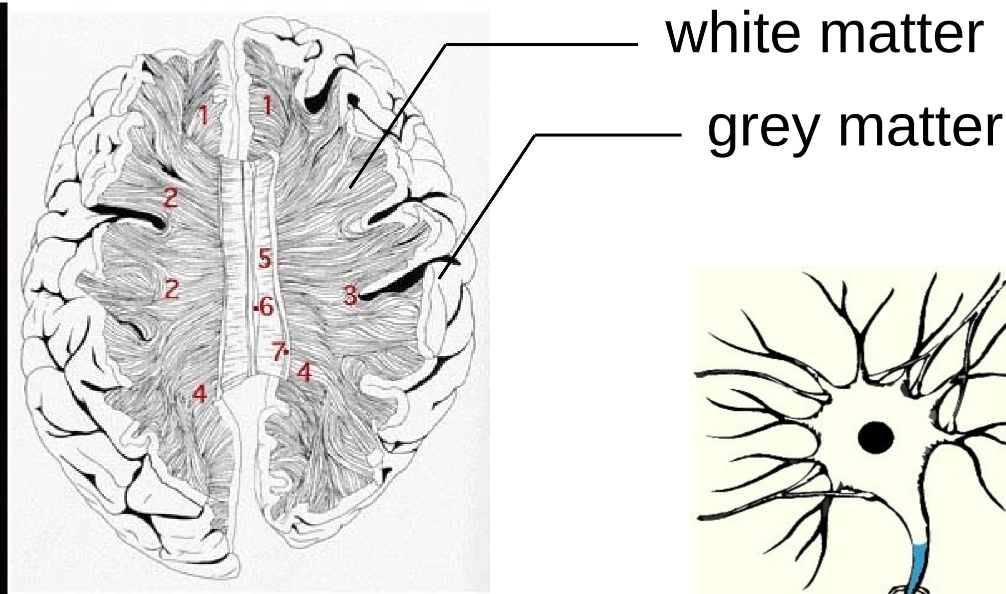
NeuroInformatics Laboratory (NILab)

Trento, January 2013

Contents

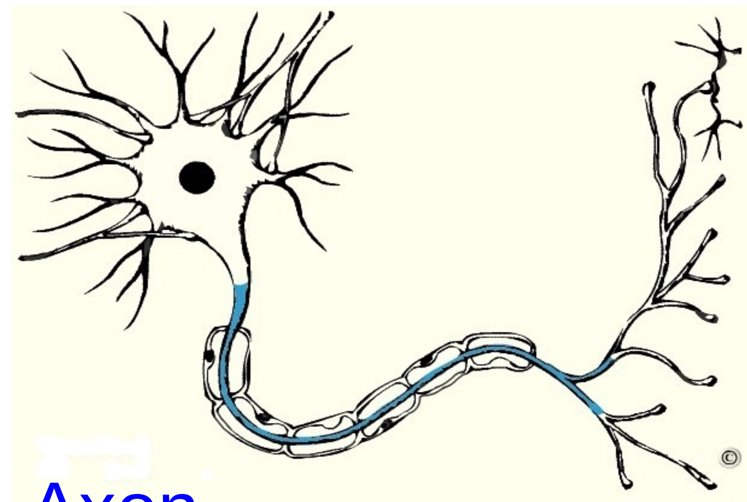
White matter





dMRI technique

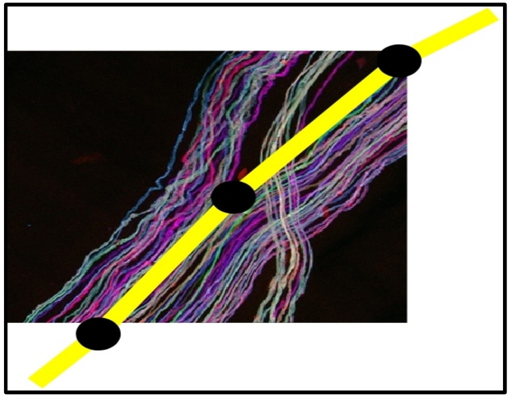
(diffusion Magnetic Resonance Imaging)
in vivo (not invasive)
Denis Le Bihan, 1984



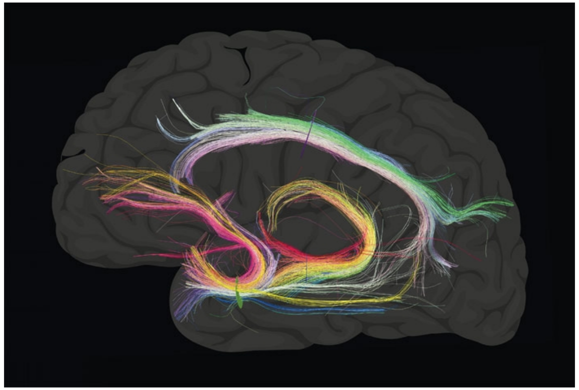
Axon

- Number: ~ 10^{12} axons
- Size: ~ 2-20µm

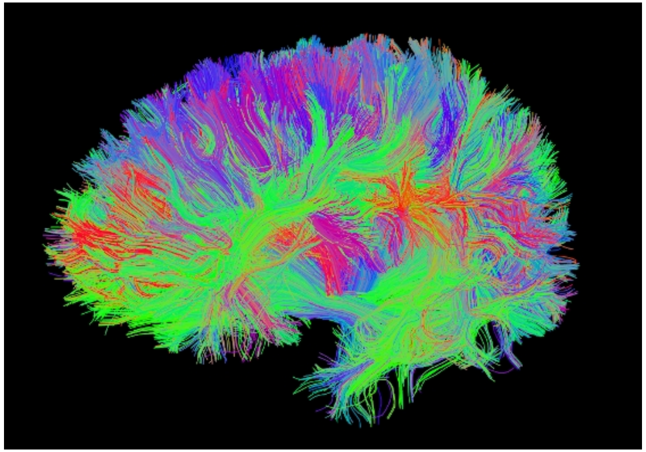
Streamline & Tractography



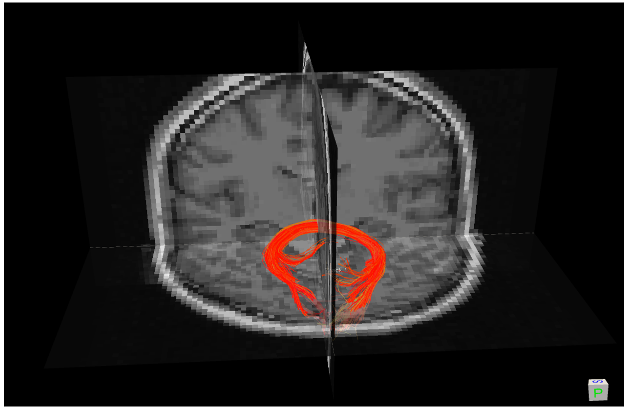
Streamline: a polyline representating thousands of axons. (fiber, track)



Bundle: a group of *'close'* streamlines

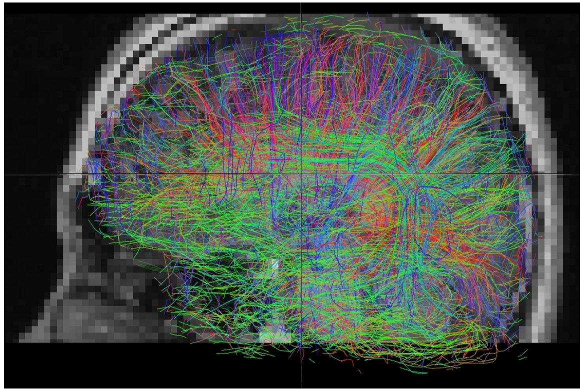


Tractography: presentation of whole brain by streamlines.

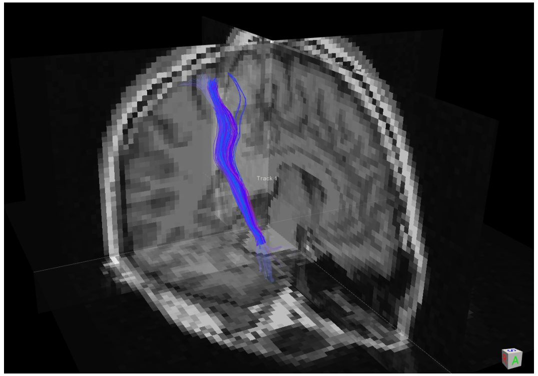


Tract: the real anatomical group of axons.

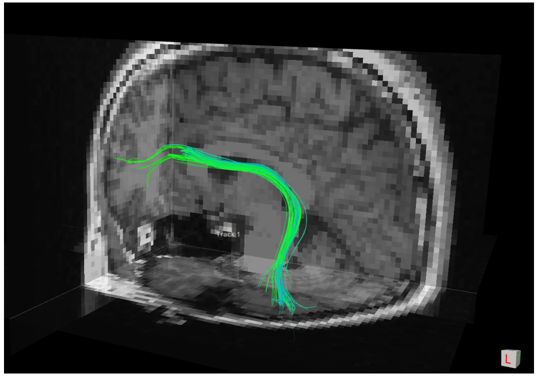
Tractography Segmentation



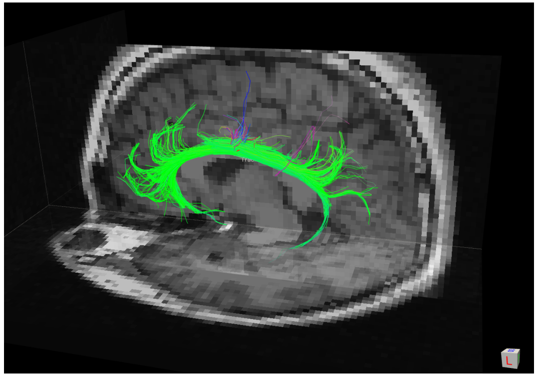
Tractography



Corticol Spnial Tract (CST)



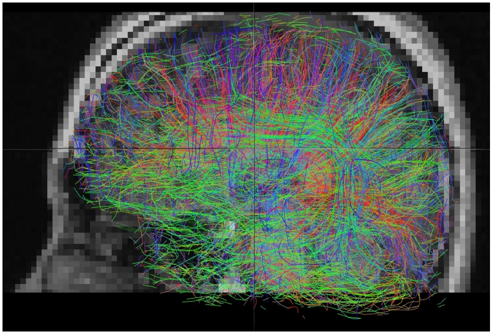
Arcuate Fasciculus Tract (AFT)



Cingulum Tract (CGT)

Group streamlines belonging to a common anatomical area into one segmentation

Tract segmentation approaches



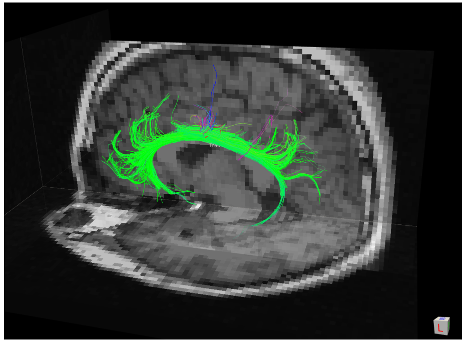
tractography

Atlas based

ROI based
(region of interest)

Unsupervised

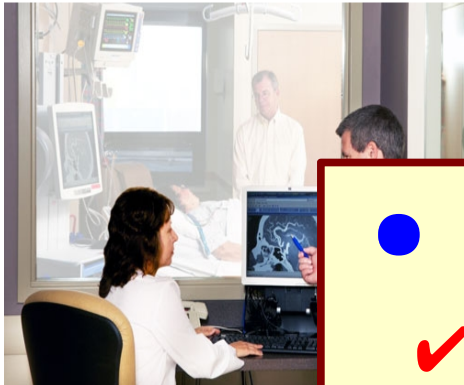
Supervised

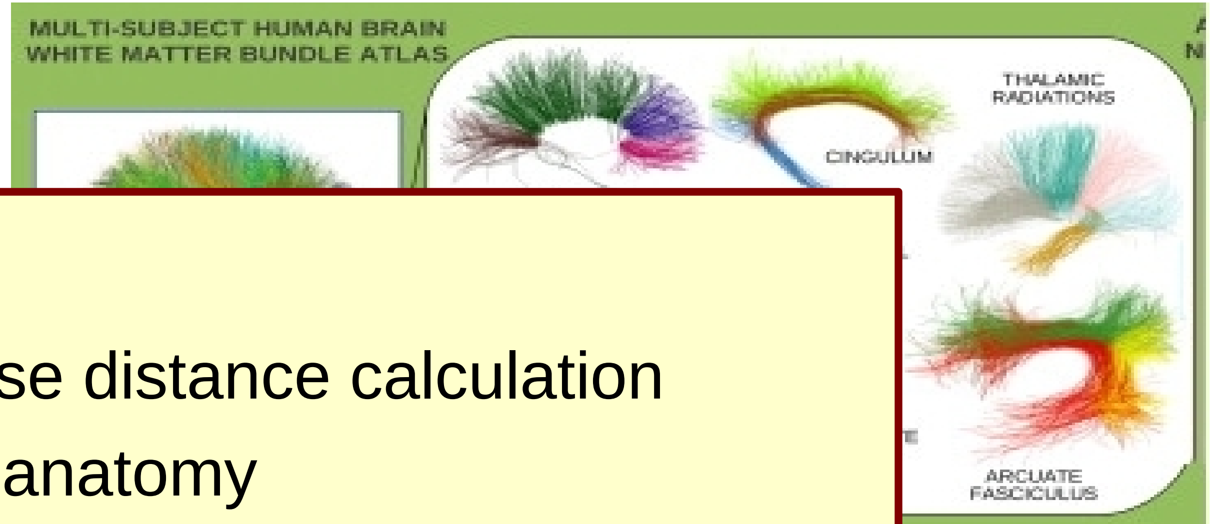


tract(s)

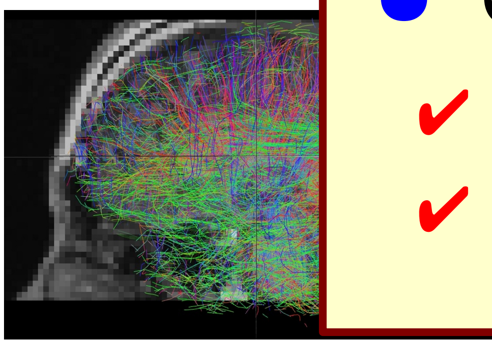
O'Donnell et al., 2007

Atlas based *Tract Segmentation*





- Pros
  - ✔ No pairwise distance calculation
  - ✔ Relate to anatomy
- Cons
  - ✔ Whole brain, not target tract
  - ✔ Brain variety → co-registration problem

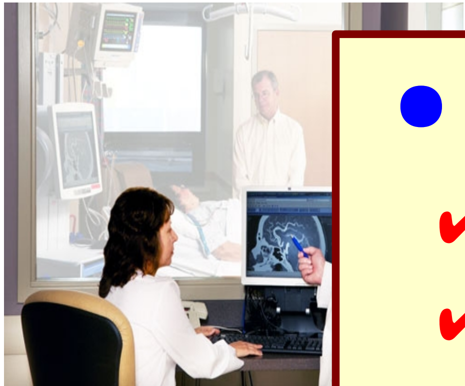


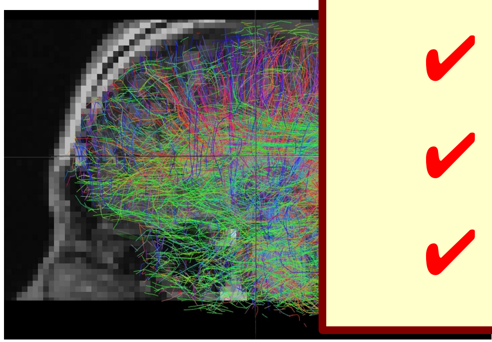
tractography

tract(s)

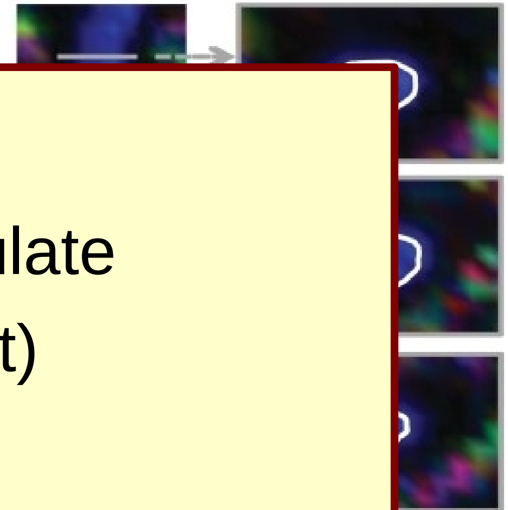
ROI based *Tract Segmentation*

(ROI:Region of Interest)
Wakana et al., 2007

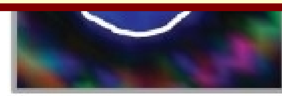




tractography



ROIs are drawn

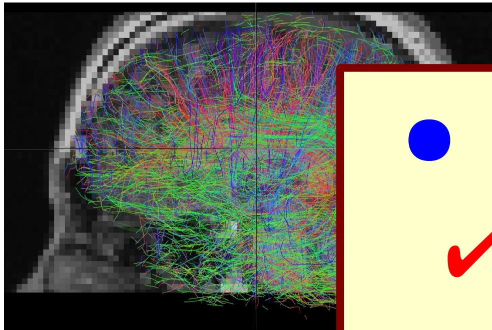


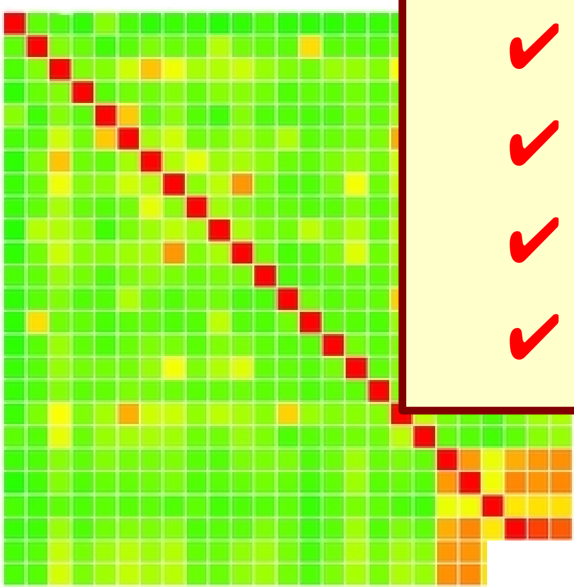
tract

- Pros
  - ✔ No pairwise distance calculate
  - ✔ Relate to anatomy (indirect)
  - ✔ Target tract
- Cons
  - ✔ Prior knowledge of tracjectory
  - ✔ Work on well characterized tracts
  - ✔ Co-registration

Unsupervised *Tract Segmentation*

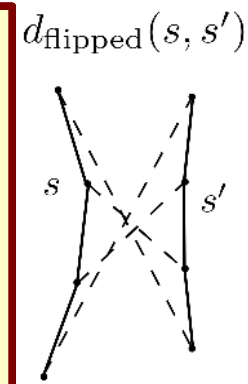
Wang et al., 2011





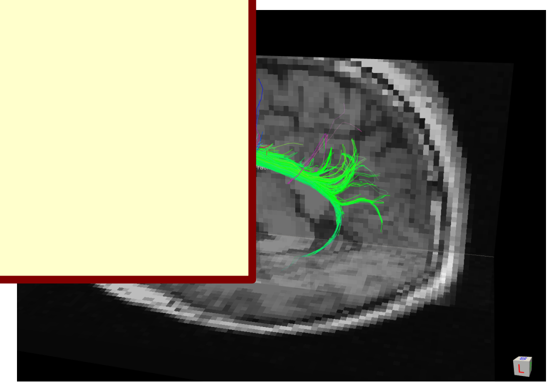
similarity matrix

Streamlines distance



- **Pros**
  - ✔ No co-registration
- **Cons**
  - ✔ Calculate pairwise distance
  - ✔ Whole brain not target tract
  - ✔ Not relate to anatomy
  - ✔ Number of cluster

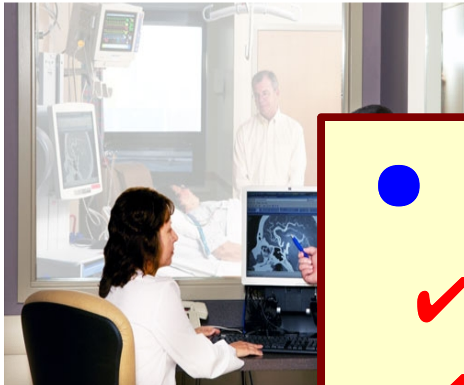
k-means, etc.)

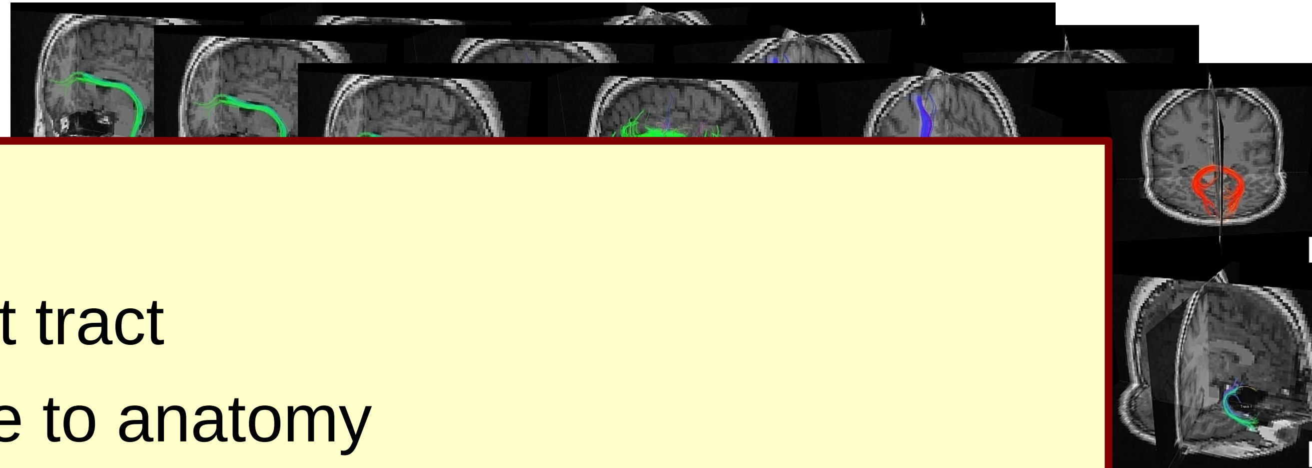


tract(s)

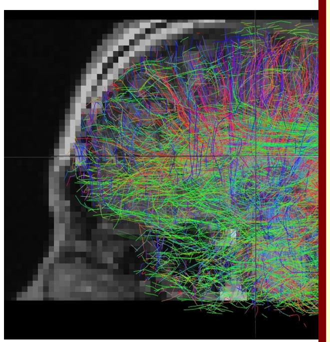
Olivetti et al., 2011

Supervised Tract Segmentation

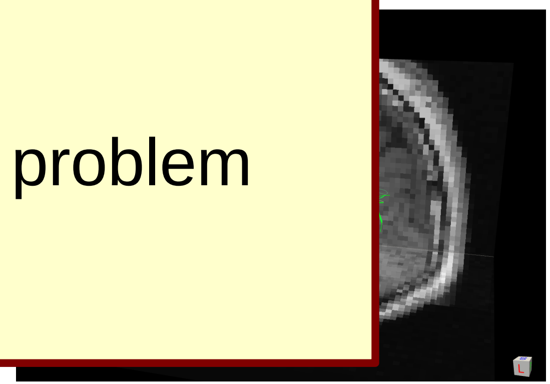




- **Pros**
  - ✔ Target tract
  - ✔ Relate to anatomy
- **Cons**
  - ✔ Calculate pairwise distance
  - ✔ Brain variety → co-registration problem
  - ✔ Ground truth data



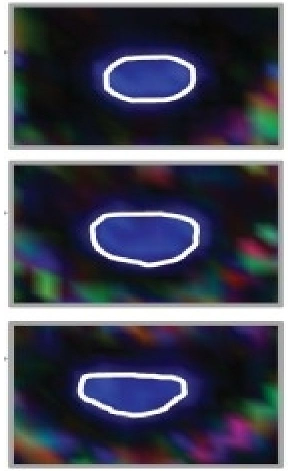
tractography

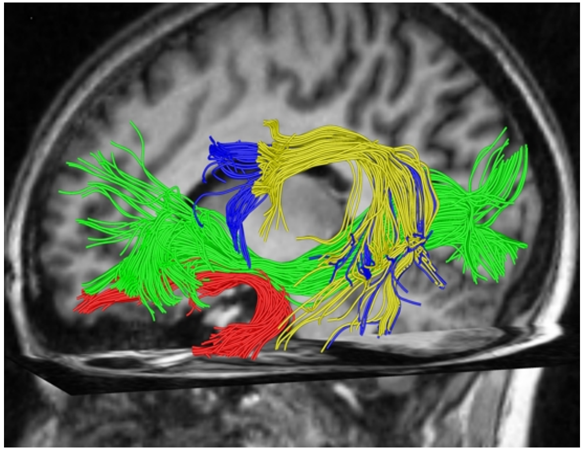


tract(s)

New strategy: BOI - Bundle of Interest

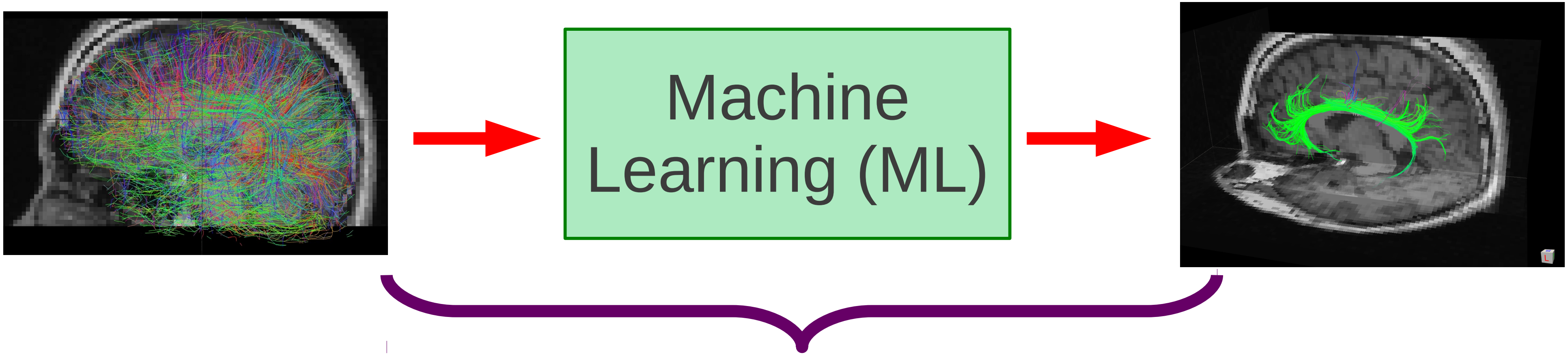
Focus directly on which bundle (cluster of streamlines) that user wants to work on





Approach	ROI	BOI
Anatomy related	Yes (indirect)	Yes
Visualization	No	Yes
Interaction	No	Yes
No prior knowledge of trajectory	No	Yes

Goals

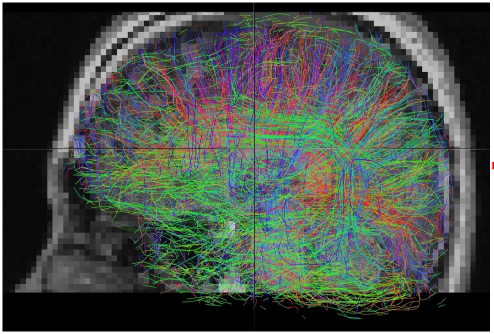


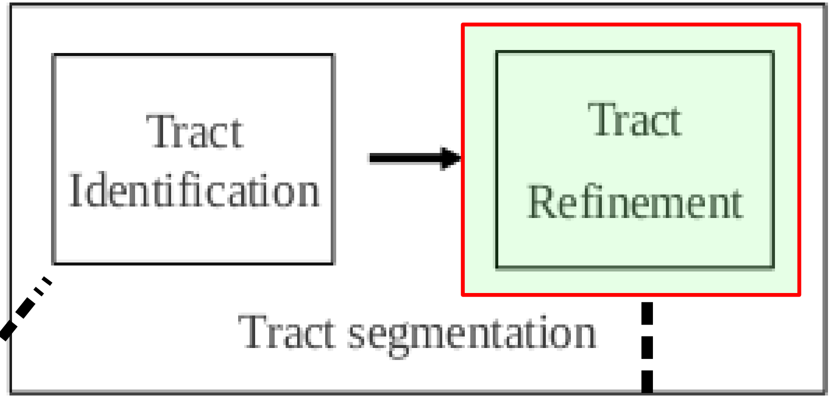
Improve the support of ML for tract segmentation

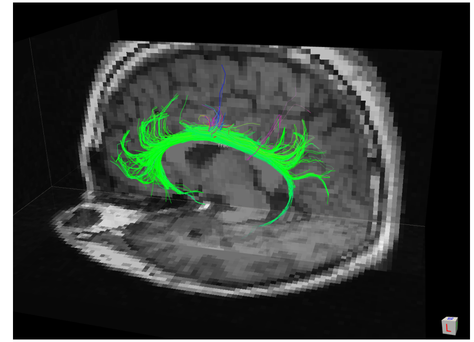
Challenges

- overcome disadvantages of Atlas, ROI
- combine both un-supervised and supervised
- design an effective method for tract segmentation

Process design: *interactive segmentation*

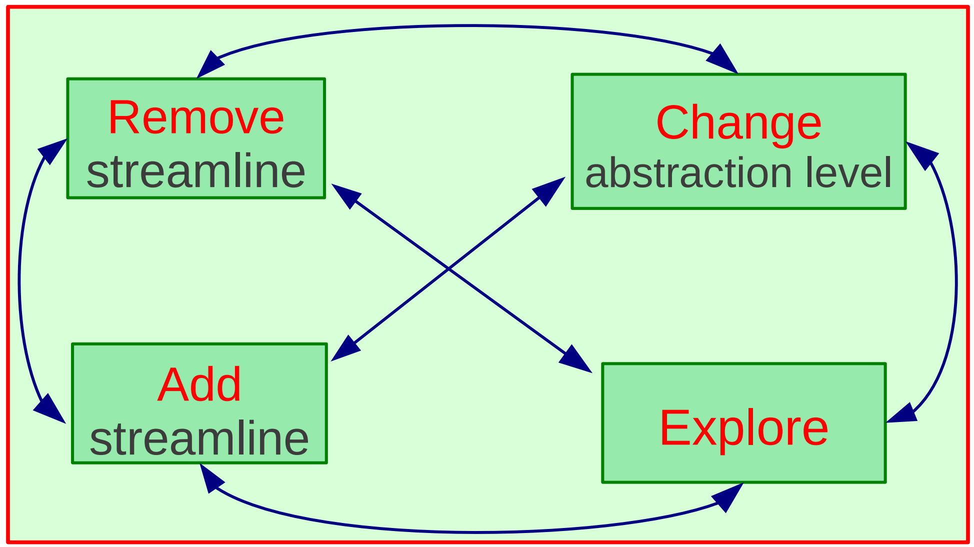






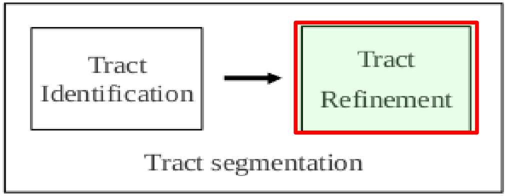
create tract candidate from set of samples

Olivetti et al., *Supervised segmentation of fiber tracts,* SIMBAD 2011



based on BOI

Interactive tract refinement



Demo of Spaghetti

Problem statement

- Given a set of N objects $\mathcal{X} = \{x_1, \ldots, x_N\}$
- Traditional partitional clustering: find one partition of $\mathcal{X}$

$$C = \{C_1, \ldots, C_K\} \text{ with } K \leq N$$

where C_i is a cluster of $\mathcal{X}$: $C_i = \{x^i_1, \ldots, x^i_j\}, j \leq N$

i $C_i \neq \emptyset, i \in [1, .., K]$

ii $\cup^K_{i=1} C_i = \mathcal{X}$

iii $C_i \cap C_j = \emptyset, \; i, j \in [1, .., K], i \neq j$

Interactive clustering

- Our approach: find a set of m partitions of $\mathcal{X}$

$$\mathcal{P} = \{ P_1, \ldots, P_m \}$$

where P_i is one partition of $\mathcal{X}$: $P_i = \{C^i_1, \ldots, C^i_{d_i}\}$

i P_i represents the *ith abstraction level* of $\mathcal{X}$

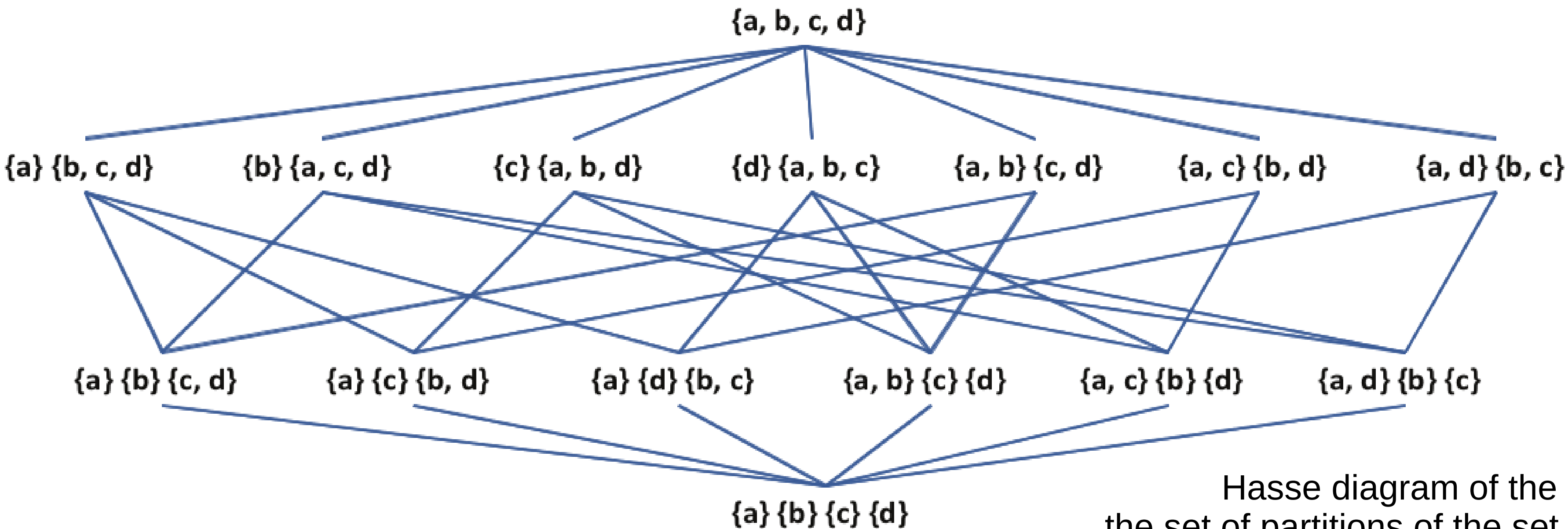
ii constraint γ: $\forall i \in [1,m-1]$, $P_i \preceq P_{i+1}$ *("nested in")*

- Denoted as a triple $\langle \mathcal{X}, \mathcal{P}, \gamma \rangle$

Interactive clustering: partial order relation

- $\mathcal{P}_{\mathcal{X}}$: set of all possible partitions of $\mathcal{X}$
- Over $\mathcal{P}_{\mathcal{X}}$, a partial order relation $\preceq$ ("nested in")

$$\forall P_a, P_b \in \mathcal{P}_{\mathcal{X}},\ P_a \preceq P_b \leftrightarrow \forall C^b_i \in P_b, \exists C^a_{i_1}, \ldots, C^a_{i_k} \in P_a : C^b_i = \cup^k_{t=1} C^a_{i_t}$$



Hasse diagram of the lattice of the set of partitions of the set {a,b,c,d}

Interactive clustering: update partitions

- Remove an old object $x_{r.m} \in \mathcal{X}$

$\mathcal{X} = \{x_1 \, . \, . \, , x_N\}$ ⇒ $\mathcal{X}' = \mathcal{X} \setminus \{ x_{\mathbf{r.m}} \}$

- Add a new object x_{add}

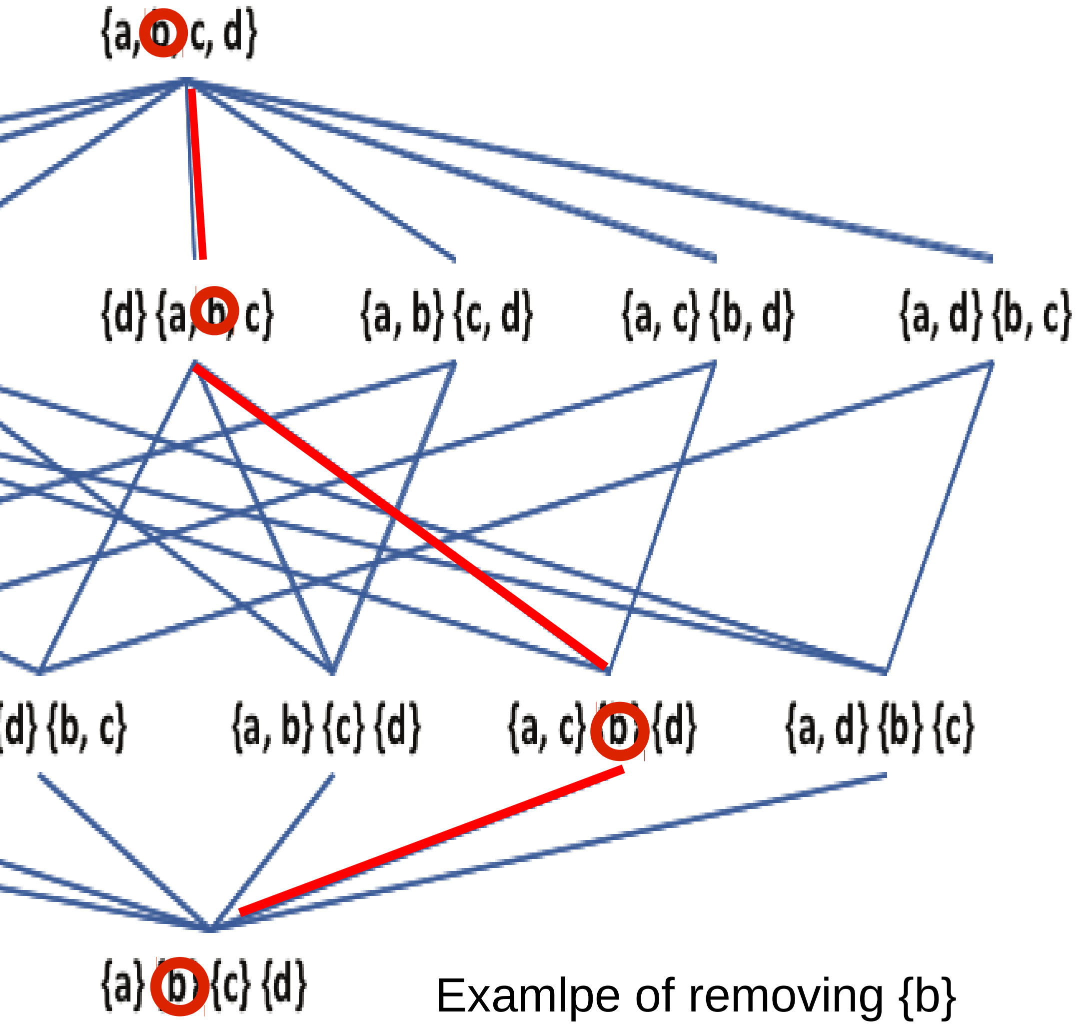
$\mathcal{X} = \{x_1 \, . \, . \, , x_N\}$ ⇒ $\mathcal{X}' = \mathcal{X} \cup \{ x_{\boldsymbol{add}} \}$

$\mathcal{P} = \{ P_1 , ... , P_m \}$ ⇒ $\mathcal{P}' = \{ P'_1 , ... , P'_{m'} \}$

$\langle \mathcal{X}, \mathcal{P}, \gamma \rangle, \gamma: P_i \preceq P_{i+1}$ $\quad i \in [1,m-1]$ ⇒ $\langle \mathcal{X}', \mathcal{P}', \gamma' \rangle, \gamma': P'_i \preceq P'_{i+1}$ $\quad i \in [1,m'-1]$

current partitions ⇒ updating partitions

Interactive clustering: *remove object*



Examlpe of removing {b}

Removing object $x_{r.m}$ from all partitions

$\forall P_i$, $i \in [1,..,m]$,

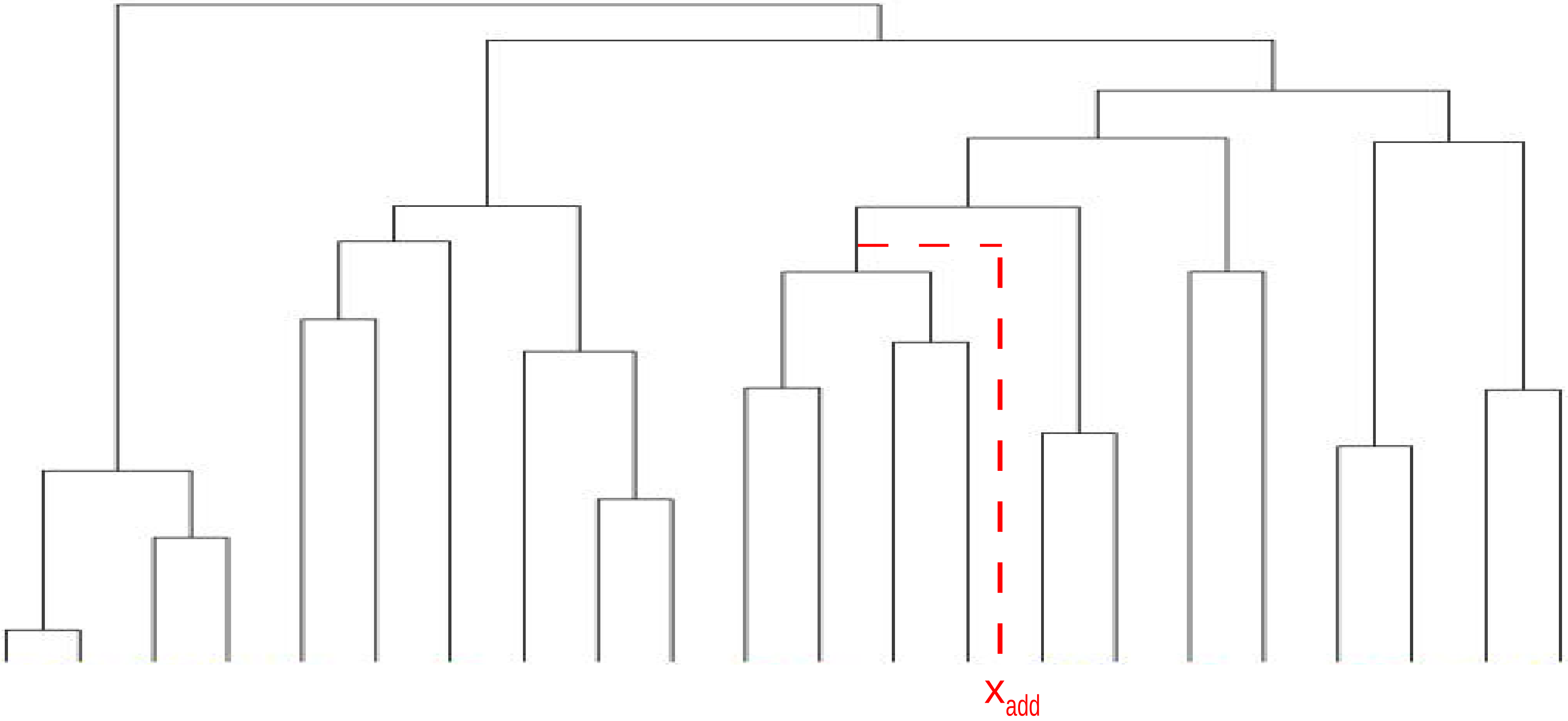
$\forall j \in [1,..,d_i]$:

if $x_{r.m} \in C_j^i$ then

$C_j^i = C_j^i \setminus \{x_{r.m}\}$,

if $C_j^i = \emptyset$, $P_i = P_i \setminus \{C_j^i\}$

Interactive clustering: *add object*



incremental clustering

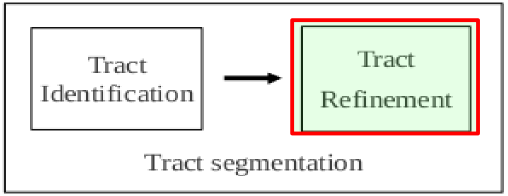
Incremental clustering – SoA

- Douglas H. Fisher, *Knowledge Acquisition Via Incremental Conceptual Clustering,* Machine Learning, Vol. 2, 1987
- Arnaud Ribert, Abdel Ennaji, Yves Lecourtier, *An Incremental Hierarchical Clustering,* Vision Interface, 1999
- Dwi H. Widyantoro & Thomas R. Ioerger, John Chen, *An incremental approach to building a cluster hierarchy,* IEEE ICDM, 2002
- Chien Y. Chen, Shien C. Hwang, Yen J. Oyang, *An incremental hierarchical data clustering algorithm based on gravity theory,* PAKDD 2002
- Samer Nassar, Jörg Sander, Corrine Cheng, *Incremental and effective data summarization for dynamic hierarchical clustering,* ACM SIGMOD MD 2004
- Ibai Gurrutxaga, Olatz Arbelaitz, José I. Martín, et al., *SIHC: A Stable Incremental Hierarchical Clustering Algorithm,* ICEIS 2009.
- M. Srinivas, C. K. Mohan, *Efficient clustering approach using incremental and hierarchical clustering methods,* IJCNN, 2010
- Dingding Wang, Tao Li, *Document update summarization using incremental hierarchical clustering,* CIKM 2010

Incremental clustering: critical factors

- Complexity
- Stability
  - Not drammatically change the current dendrogram
  - User can assimilate the changes
- Data summarization

Interactive clustering – close solutions



- Widyantoro et al., 2002

An incremental approach to building a cluster hierarchy

O(N log N) v.s build MST (O(N^2 log N)) + not stable structure

- Nassar et al., 2004

Incremental and effective data summarization for dynamic hierarchical clustering

O (m * N) v.s data summarization

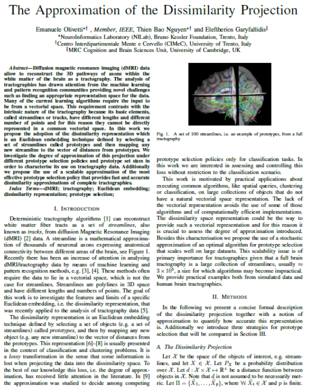
- Gurrutxaga et al., 2009

SIHC: A Stable Incremental Hierarchical Clustering Algorithm

Keep a stable structure v.s O(N^2 log N)

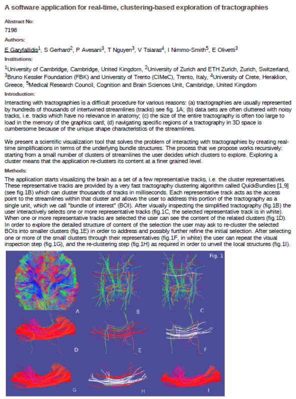
Preliminary Results

- (method) Dissimilarity representation


The Approximation of the Dissimilarity Projection

E. Olivetti, **T. B. Nguyen**, E. Garyfallidis,
The Approximation of the Dissimilarity Projection,
Pattern Recognition in NeuroImaging, PRNI 2012.

- (software/tool) Spaghetti: an interactive visualization tool for segmentation tractography


A software application for real-time, clustering-based exploration of tractographies

E. Garyfallidis, S. Gerhard, P. Avesani,
T. B. Nguyen, V. Tsiaras, I. N. Smith, E. Olivetti,
A software application for real-time, clustering based exploration of tractographies, OHBM 2012.

- (case study) ALS (amyotrophic lateral sclerosis) disease (in preparation)

Conclusion

- An effective method for tract segmentation: tract candidate (supervised) and tract refinement (clustering)
- An interactive visualization tool for tract segmentation
- ALS case study

Future works

- Implement the modified HAC for tractography
- Revise the solution for 'adding object' to partitions
- Integerate tract candidate (supervised) into Spaghetti

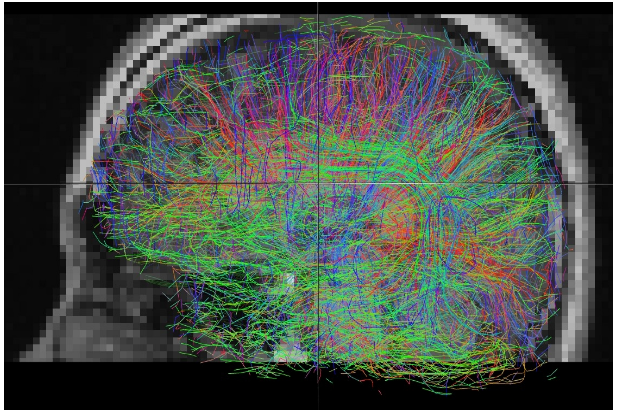
Credits

- **Nivedita Agarwal**, *S.Chiara Trento Hospital, Italy; University of Utah, USA*
- **Eleftherios Garyfallidis**, *University of Cambridge, UK; University of Sherbrooke, Canada*
- **Emanuele Olivetti**, *Fondazione Bruno Kessler, Italy*
- **Paolo Avesani**, *Fondazione Bruno Kessler, Italy*
- **Luigi Cattaneo**, *CiMeC, University of Trento, Italy*
- **Francesca Maule**, *CiMeC, University of Trento, Italy*

Thank you!

dMRI technique

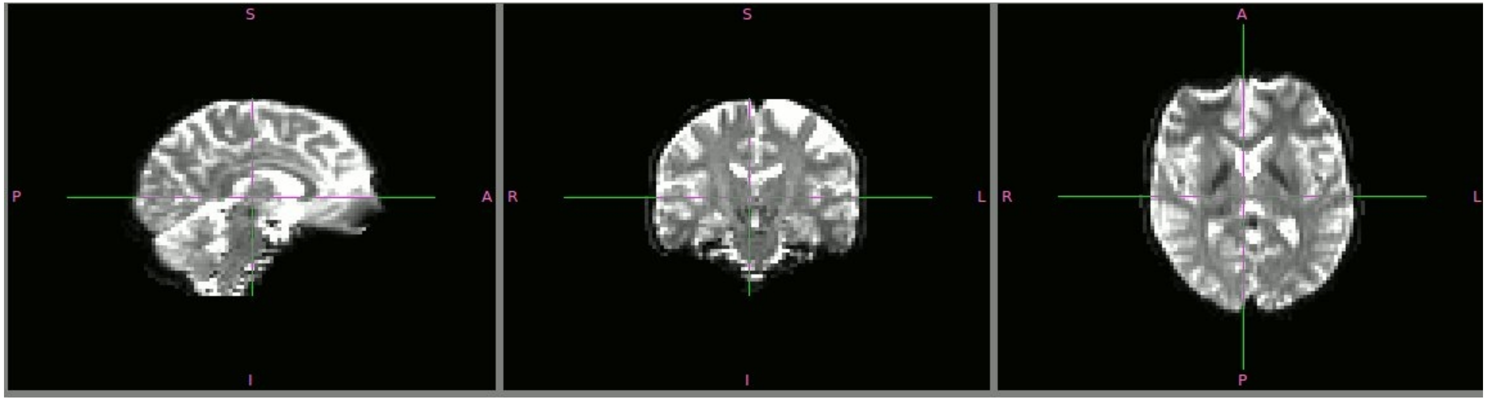
(diffusion Magnetic Resonance Imaging)
in vivo (not invasive)
Denis Le Bihan, 1984



Tractography in 3D

(*)

Garyfallidis et. al. 2012
(Towards an accurate brain tractography)



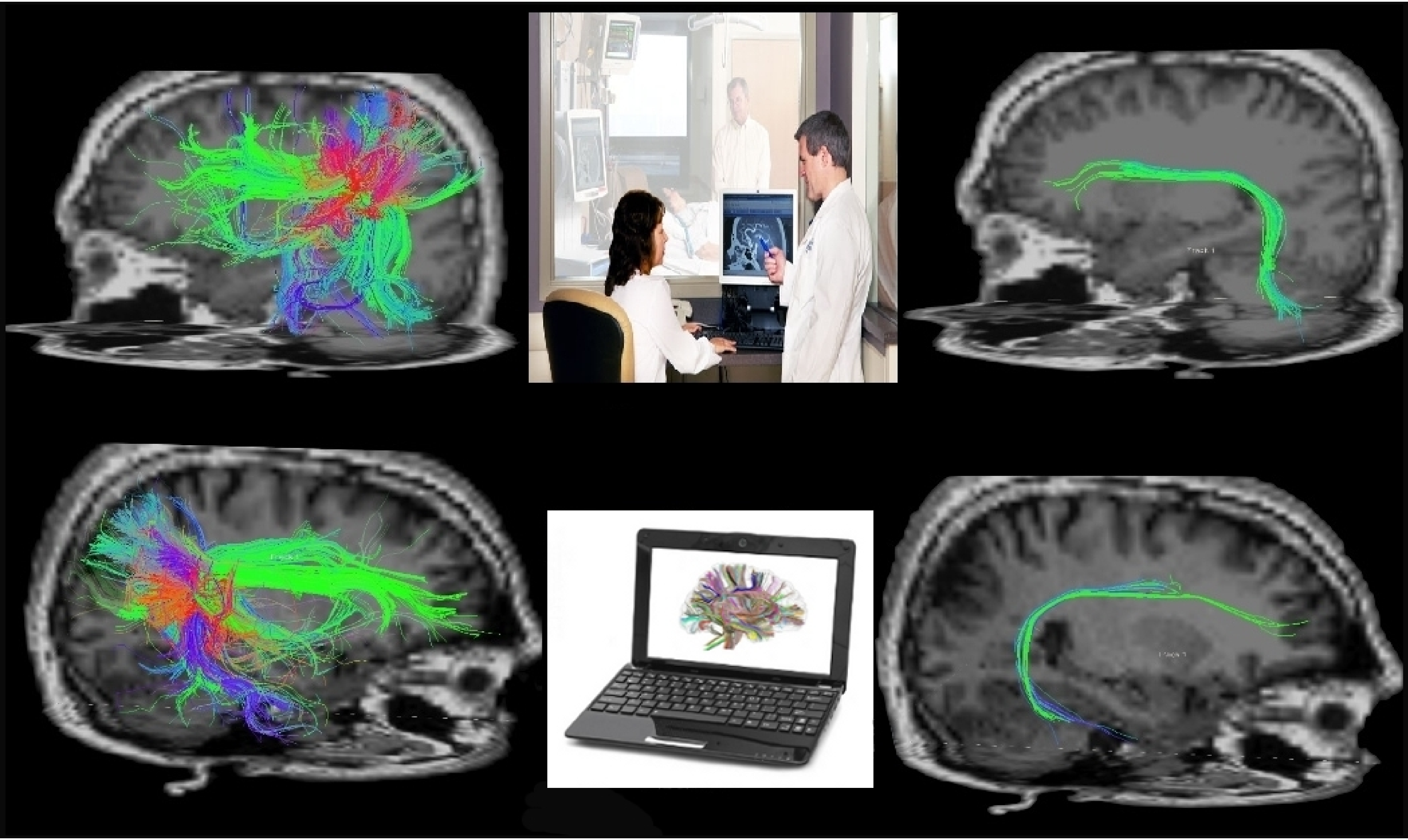
MRI images in 3D

Tract Segmentation: Human Expert

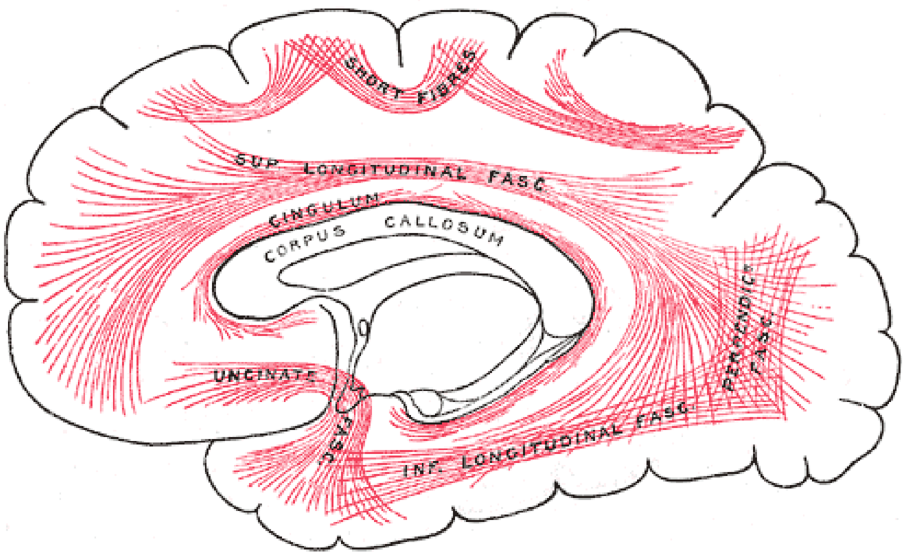
large amount of streamlines

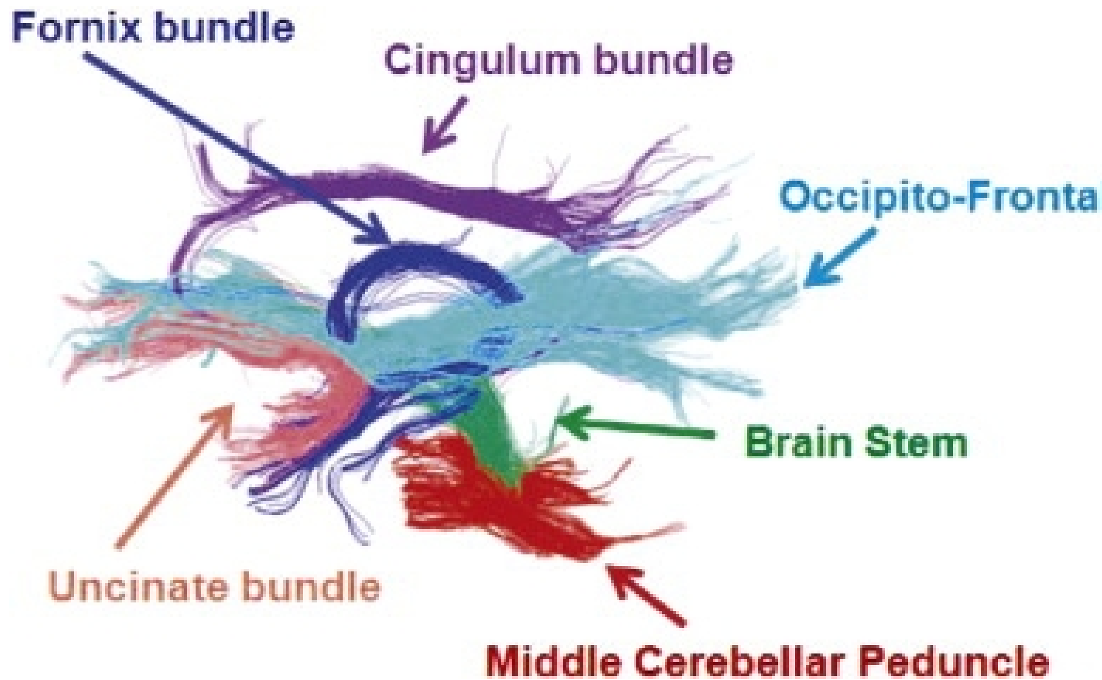
anatomical variability among subjects

segmentation difficult and lengthy



Tractography Segmentation





To group streamlines belonging to a common anatomical area into one segmentation

Survey of tractography segmentation methods

Approach	Target tract	Anatomy related	No co-registration	No pairwise distance	Visualization/ Interaction
Atlas	No	Yes	No	Yes	No
ROI	Yes (indirect)	Yes (indirect)	No	Yes	No
Unsupervised	No	No	Yes	No (costly)	No
Supervised	Yes	Yes (indirect)	No	Yes	No

pros

con

Comparison

Approach	Target tract	Related anatomy	No co-registration	No pairwise distance	Visualization \Interaction
Atlas	No	Yes	No	Yes	No
ROI	Yes (indirect)	Yes (indirect)	No	Yes	No
Unsupervised	No	No	Yes	No (costly)	No
Supervised	Yes	Yes (indirect)	No	Yes	No
Ours	Yes	Yes	No(*)	Yes	Yes

Data representation

Most of learning algorithms assume objects to be in an Euclidean feature space

Issue: streamlines $s = \{x_1, \ldots, x_{ns}\}$, where $x \in \mathbb{R}^3$

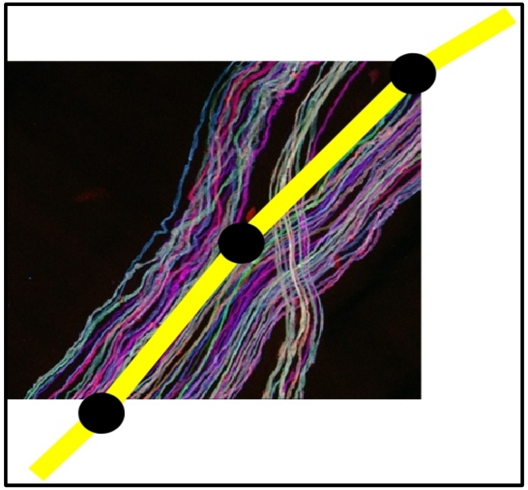
- The number of points is not the same
- Lengths are different

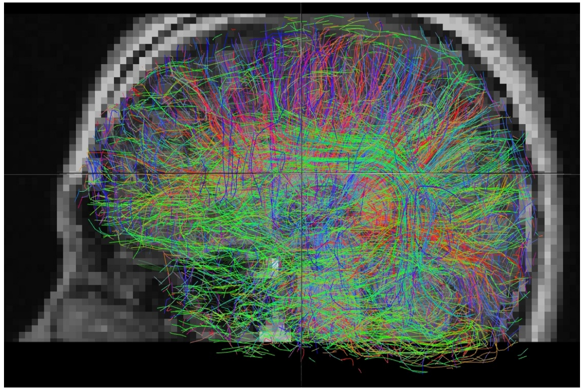
Find a representation ϕ of streamline $\phi : \mathcal{T} \rightarrow \mathbb{R}^d$

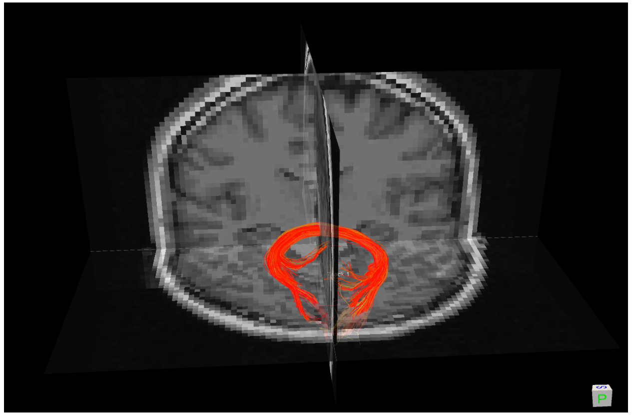
minimize *lossy* : $\phi = \operatorname{argmin} L(\phi^{-1}(s))$

Notations

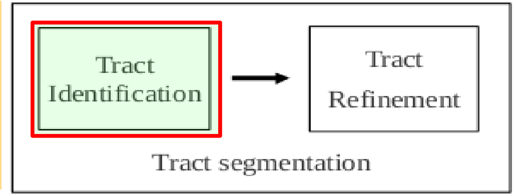
- Streamline: a polyline $s = \{x_1, \ldots, x_{ns}\}$, where $x \in \mathbb{R}^3$
- Tractography: a set of n streamlines $\mathcal{T} = \{s_1, \ldots, s_n\}$
- Taget tract: $T_{target} \subset \mathcal{T}$

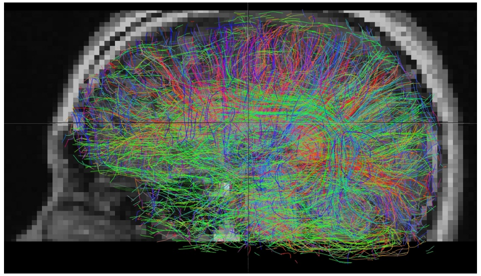






Tract identification



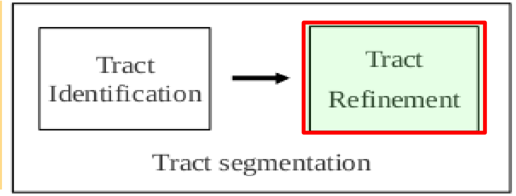


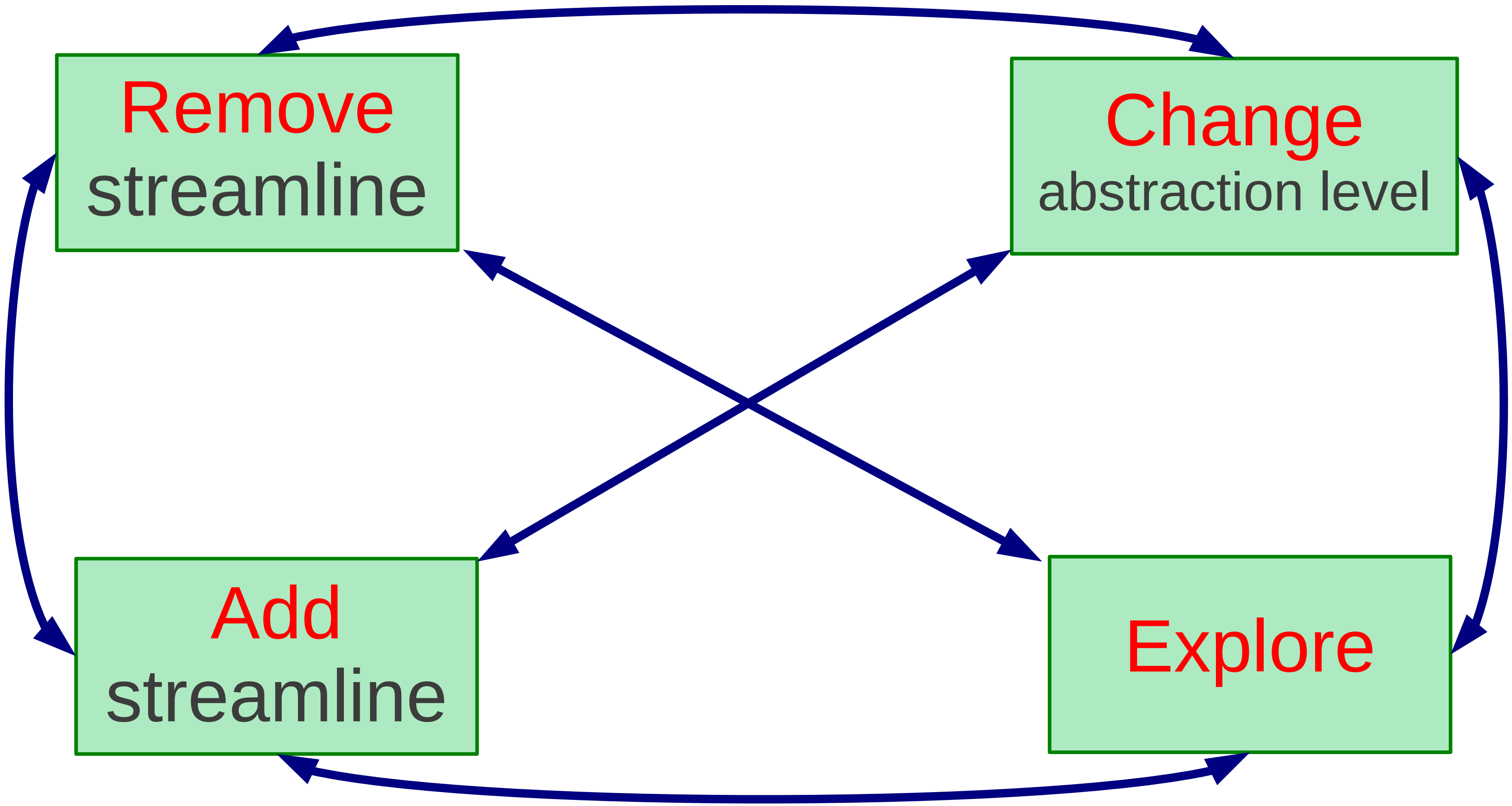
→ Supervised Learning → Tract candidate

(*)Olivetti et al., 2011

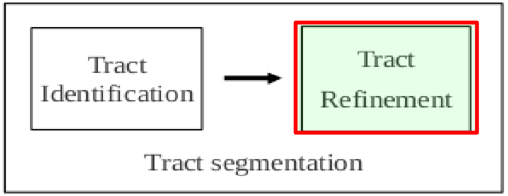
- Given a class-labeled sample $E = \{(s_1, y_1), \ldots, (s_N, y_N)\}$, with $y_i = 0$ if $y_i \in T_{target}$; $y_i = 1$ otherwise.
- Learn classifier f from E by minimizing the loss function L: $f = \text{argmin } L(f'(s), y)$
- Tract candidate: $T_{candidate} = \{ s_i \mid f(s_i) = 1, \ \forall s_i \in \mathcal{T} \}$

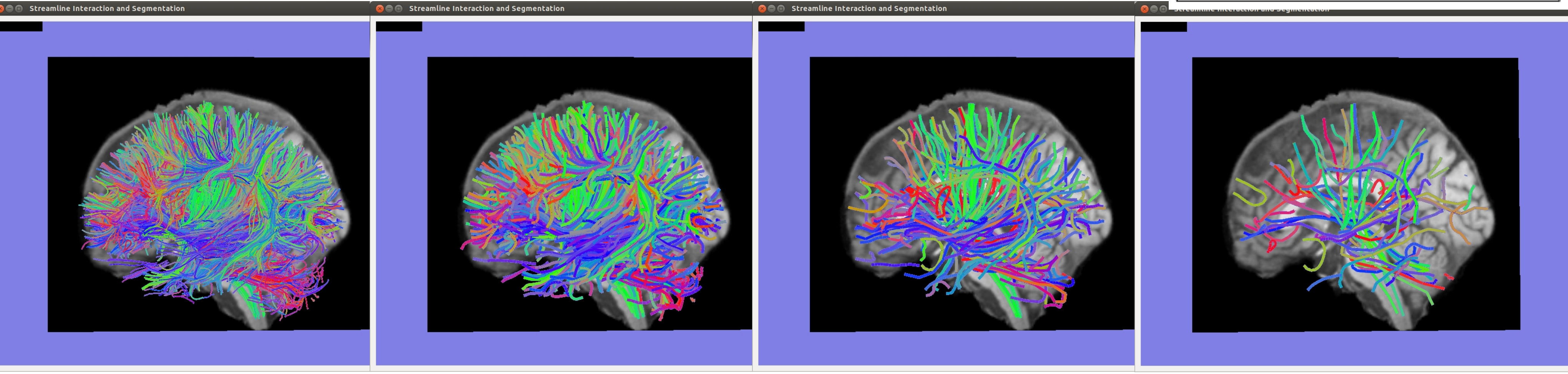
Interactive tract refinement





Interactive tract refinement

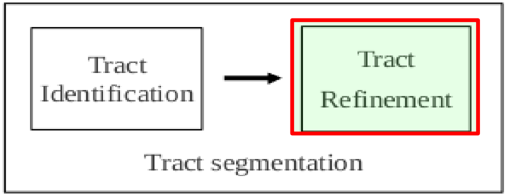




abstraction level

large ammount of streamlines → difficult to interpret and interact → **Cluster** in **multiple-level** of abstraction

Tractography clustering



- Given a set of N streamlines $\mathcal{T} = \{s_1, \ldots, s_N\}$

- Traditional approaches: find one partition of $\mathcal{T}$

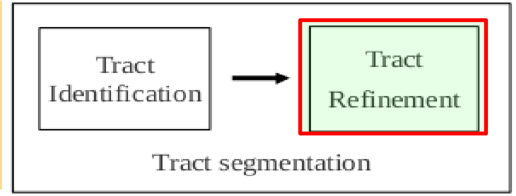
$$C = \{C_1, \ldots, C_K\} \text{ with } K \leq N$$

with C_i is a cluster of $\mathcal{T}$: $C_i = \{s^i_1, \ldots, s^i_j\}$, $j \leq N$

i $C_i \neq \emptyset$, $i = 1, \ldots, K$

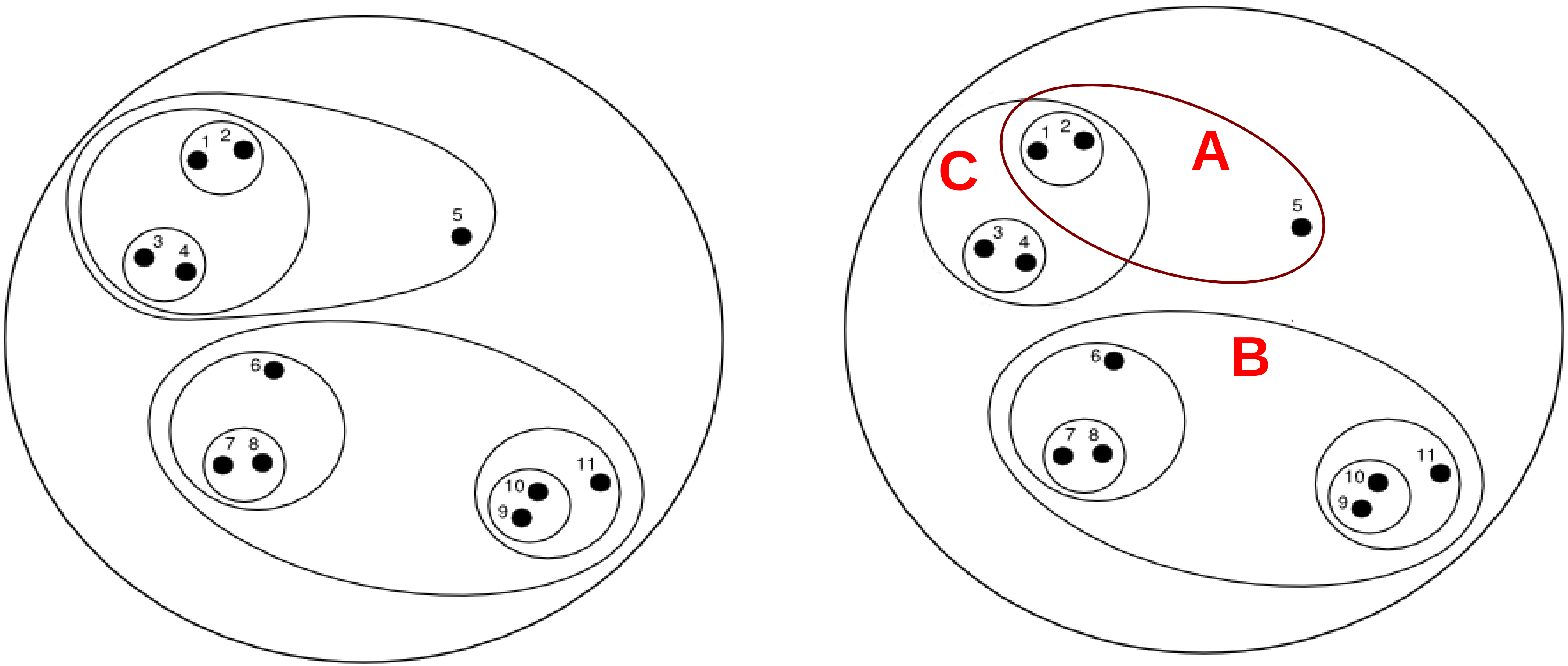
ii $\bigcup^{K}_{i=1} C_i = \mathcal{T}$

Interactive tractography clustering

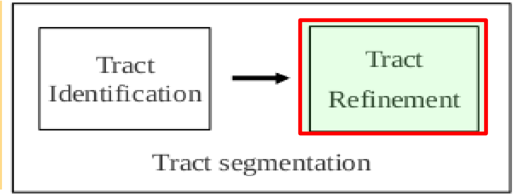


$$\forall P_i , P_j , i > j, \ (C^j_k \subseteq C^i_l) \ \vee \ (C^j_k \cap C^i_l \ = \emptyset)$$

$l \in [1,..,d_i]$,
$k \in [1,..,d_j]$



Neighbor checking



- Given a distance measure $d(s_a,s_b)$ on whole tractography T_B

$$neighbor(T,\theta) = \{s \mid d(s,s_t) \leq \theta,\ \forall s_t \in T,\ s \in T_B,\ s \neq s_t\}$$

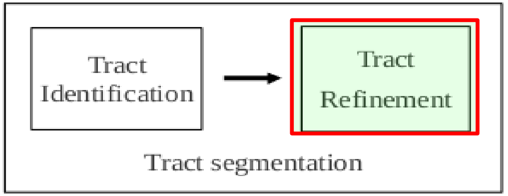
- Add streamline $s_a \in neighbor(T,\theta)$ to T ???
  - The current viewing partition is P_i
  - $C_{close}(s_a,P_i)$ is the cluster in P_i closest to s_a

$$C_{close}(s_a,P_i) = \mathrm{argmin}\ (d(s_a,C^i_k)),\ \forall C^i_k \in P_i$$

  - $d_{min}(P_i)$ is the min distance between two clusters of P_i

$$d_{min}(P_i) = \mathrm{argmin}\ (d(C^i_k,C^i_l)), \forall C^i_l,C^i_k \in P_i,\ k \neq l$$

Neighbor checking – *add streamline*



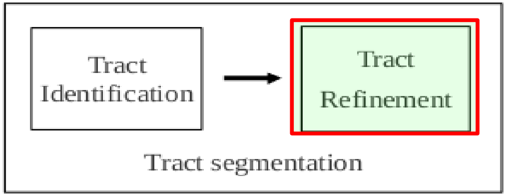
- Case 1: $d(s_a, C_{close}(s_a, P_i)) > d_{min}(P_i)$
  - s_a becomes a new cluster: $P'_i = P_i \cup \{s_a\}$
  - $<\mathcal{P}', \mathcal{T}', \gamma_2>$ where γ_2 is the constraint as:

$$\gamma_2 : \forall\, j \in [1, .., i], \exists C'^{j}_{k} \in P'_j : C'^{j}_{k} = \{ s_a \}$$

- Case 2: $d(s_a, C_{close}(s_a, P_i)) \leq d_{min}(P_i)$
  - s_a is merged into $C_{close}(s_a, P_i)$: $C_{close}(s_a, P_i) = C_{close}(s_a, P_i) \cup \{s_a\}$
  - $<\mathcal{P}', \mathcal{T}', \gamma_3>$ where γ_3 is the constraint as:

$$\gamma_3 : \forall\, j \in [i, .., m], \exists C'^{j}_{k} \in P'_j : (C_{close}(s_a, P_i) \cup \{s_a\}) \subseteq C'^{j}_{k}$$

Neighbor checking



- Given a distance measure $d(s_a,s_b)$ on whole tractography T_B

  $$neighbor(T,\theta) = \{s \mid d(s,s_t) \le \theta,\ \forall s_t \in T,\ s \in T_B,\ s \ne s_t\}$$

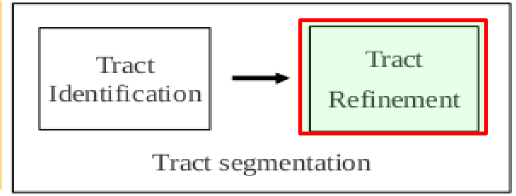
- Add streamline $s_a \in neighbor(T,\theta)$ to T ???
  - The current viewing partition is P_i
  - $C_{close}(s_a,P_i)$ is the cluster in P_i closest to s_a

    $$C_{close}(s_a,P_i) = \text{argmin}\ (d(s_a,C^i_k)),\ \forall C^i_k \in P_i$$

  - $d_{min}(P_i)$ is the min distance between two clusters of P_i

    $$d_{min}(P_i) = \text{argmin}\ (d(C^i_k,C^i_l)), \forall C^i_l,C^i_k \in P_i,\ k \ne l$$

Neighbor checking – *add streamline*



- Case 1: $d(s_a, C_{close}(s_a, P_i)) > d_{min}(P_i)$
  - s_a becomes a new cluster: $P'_i = P_i \cup \{s_a\}$
  - $<\mathcal{P}', \mathcal{T}', \gamma_2>$ where γ_2 is the constraint as:

$$\gamma_2 : \forall j \in [1, .., i], \exists C'^j_k \in P'_j : C'^j_k = \{ s_a \}$$

- Case 2: $d(s_a, C_{close}(s_a, P_i)) \leq d_{min}(P_i)$
  - s_a is merged into $C_{close}(s_a, P_i)$: $C_{close}(s_a, P_i) = C_{close}(s_a, P_i) \cup \{s_a\}$
  - $<\mathcal{P}', \mathcal{T}', \gamma_3>$ where γ_3 is the constraint as:

$$\gamma_3 : \forall j \in [i, .., m], \exists C'^j_k \in P'_j : (C_{close}(s_a, P_i) \cup \{s_a\}) \subseteq C'^j_k$$

Interactive clustering: add new object

- Current viewing abstraction level is *ith*
- $C(x, P_i)$: cluster in parition P_i having object x

- All upper partitions

$$\gamma_1' : \forall j \in [i, .., m], \ \forall x \in \mathcal{X}:$$

$$C(x,P'_i) = C(x_{add},P'_i) \rightarrow C(x,P'_j) = C(x_{add},P'_j)$$

- All lower partitions

$$\gamma_2' : \forall k \in [1, .., i-1], \ \forall x \in \mathcal{X}:$$

$$C(x,P'_i) \neq C(x_{add},P'_i) \rightarrow C(x,P'_k) \neq C(x_{add},P'_k)$$

Dissimilarity approximation for tractography

- Given a tractography of n streamlines $\mathcal{T} = \{s_1, \ldots, s_n\}$
- Chose a set of prototype: $\Pi = \{t_1, \ldots, t_d\}$, $\forall i \; t_i \in \mathcal{T}$, $d \leq n$
- Dissimilarity representation:

$$\phi : \mathcal{T} \rightarrow \mathbb{R}^d$$

$$\phi_\Pi^d(s) = [d(s,t_1), \ldots, d(s,t_d)]$$

Good representation

$$\rho = \frac{\mathrm{Cov}(d(s,s'), \Delta_\Pi^d(s,s'))}{\sigma_{d(s,s')}\sigma_{\Delta_\Pi^d(s,s')}}$$

- ρ far from zero and close to 1
- $\Delta_\Pi^d(s,s') = ||\phi_\Pi^d(s) - \phi_\Pi^d(s')||_2$

Prototype policy

- Random
- FFT (farthest first travel)
- SFF (subset farther first)

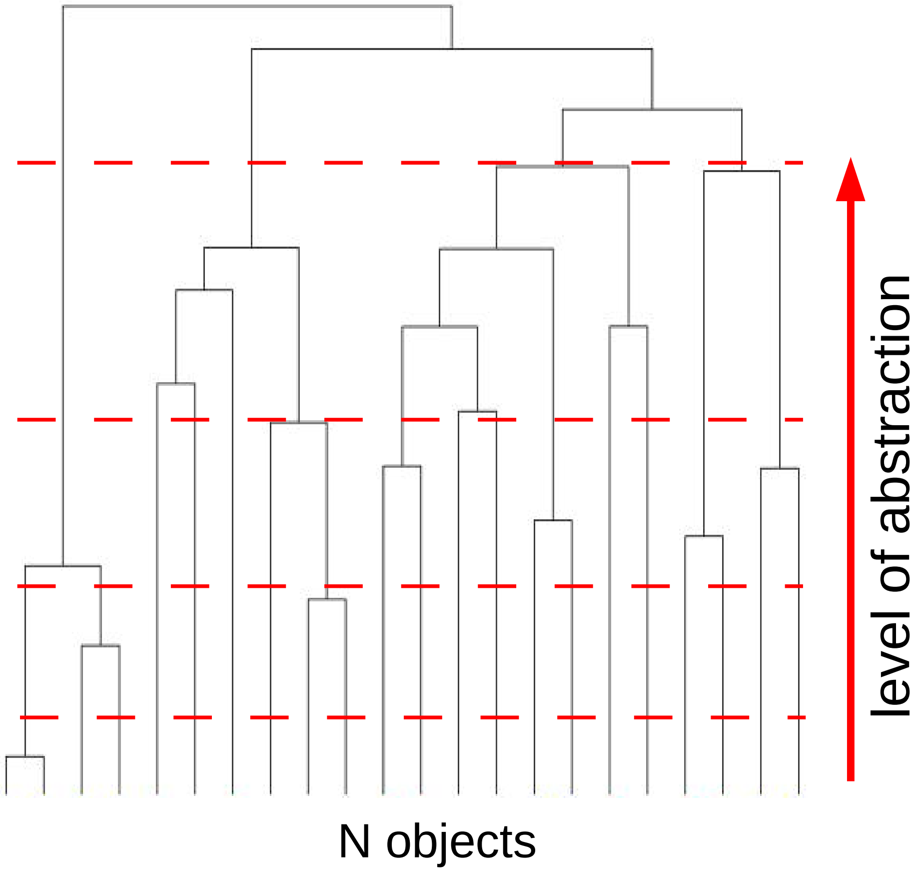
Hierarchical clustering

Johnson, 1967

Find m partitions of $\mathcal{X} : \mathcal{P} = \{P_1, \ldots, P_m\}$, $P_i \preceq P_{i+1}$, $i \in [1, m-1]$

Algorithm

1. Assign each x_i to one cluster
2. Merge two closest clusters
3. Compute distances
4. Repeat until all in one cluster



Interactive clustering: *add object*

Widyantoro et al., 2002
Gurrutxaga et al., 2009

Denote δ_{max} as the maximum distance in cluster C_i

$$\delta_{max}(C_i) = \max \{d(x,x')\}, \forall x,x' \in C_i, x \neq x'$$

1. Find the closet cluster of x_{add} in P_1: $C_{close}(x_{add},P_1)$

2. Start from the direct parent of $C_{close}(x_{add},P_1)$:

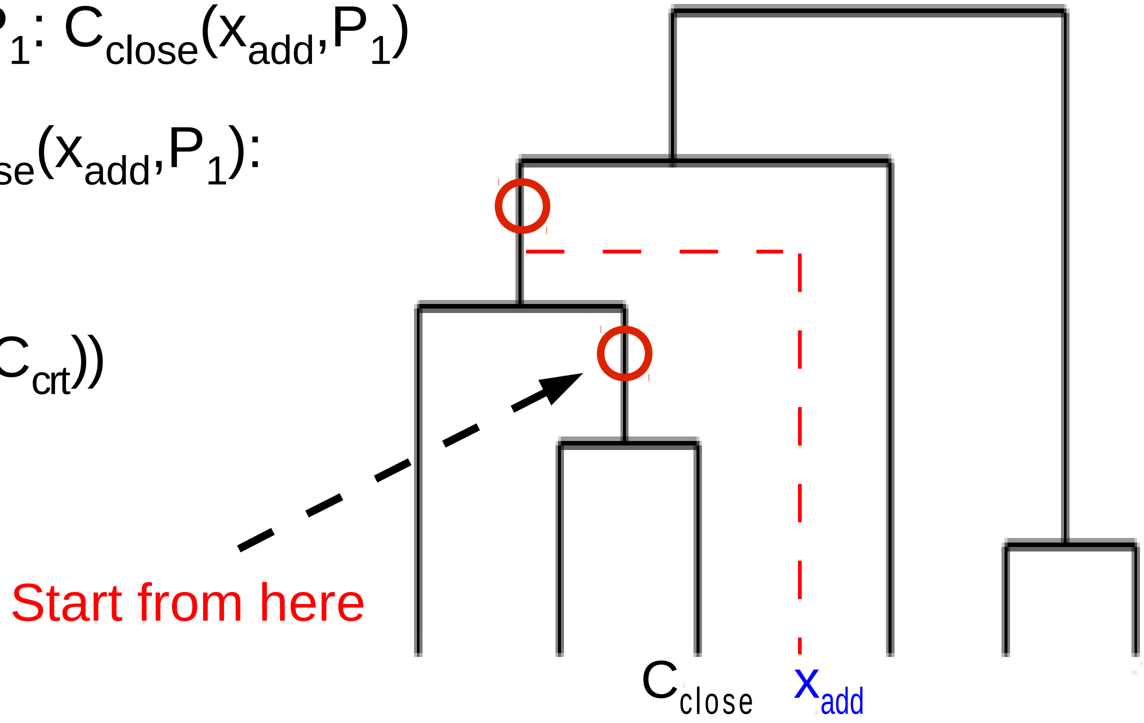
   $C_{crt} = C_{close}(x_{add},P_1).parent$

3. If $(C_{crt} = \emptyset) \vee (d(x_{add}, C_{crt}) < \delta_{max}(C_{crt}))$

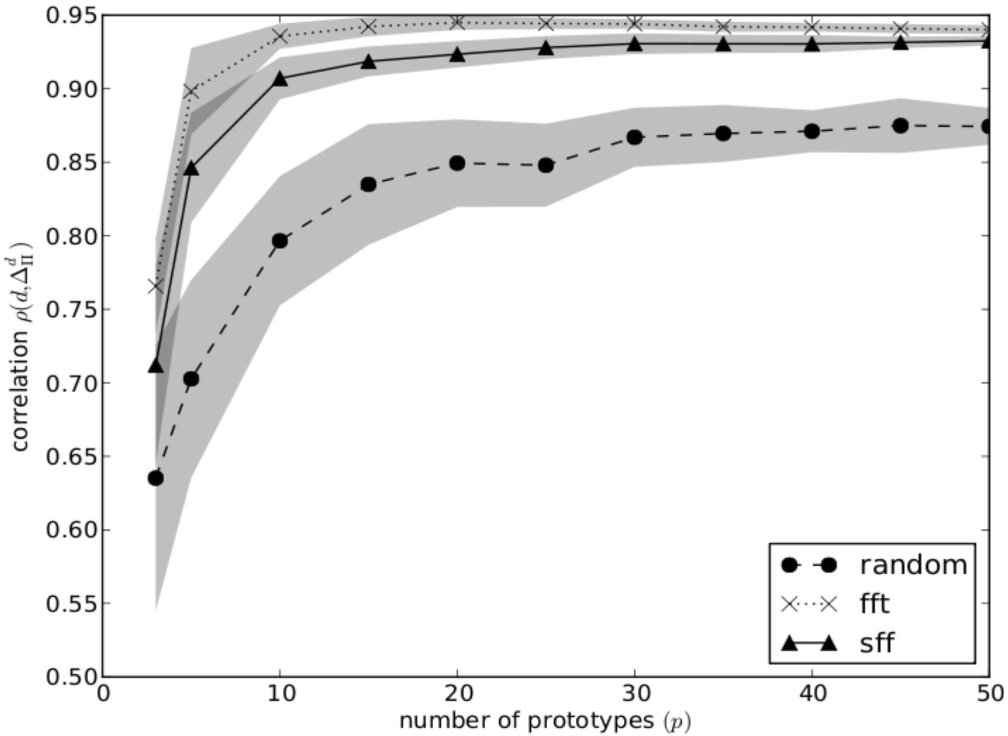
   3.1. Merge: $C_{crt} = C_{crt} \cup \{s_a\}$

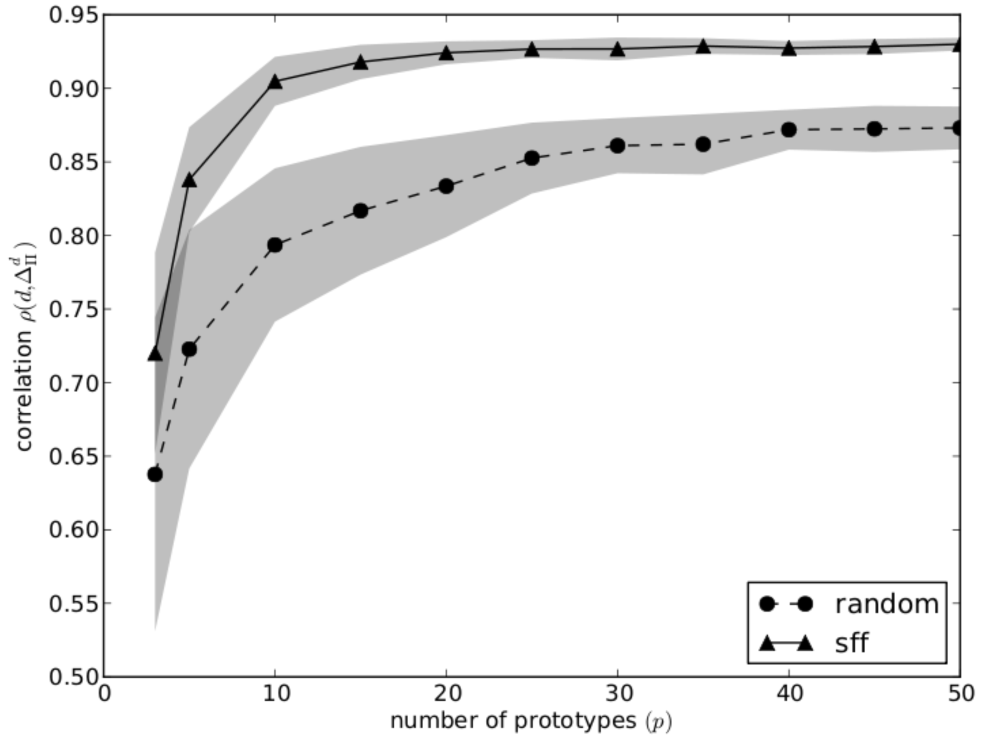
   3.2. Stop

4. $C_{crt} = C_{crt}.parent$, goto step 3



(dis)Similarity approximation for tractography





FFT: < 1secs, SFF: 2 secs (one iteration)

Tractography of 10^3 streamlines

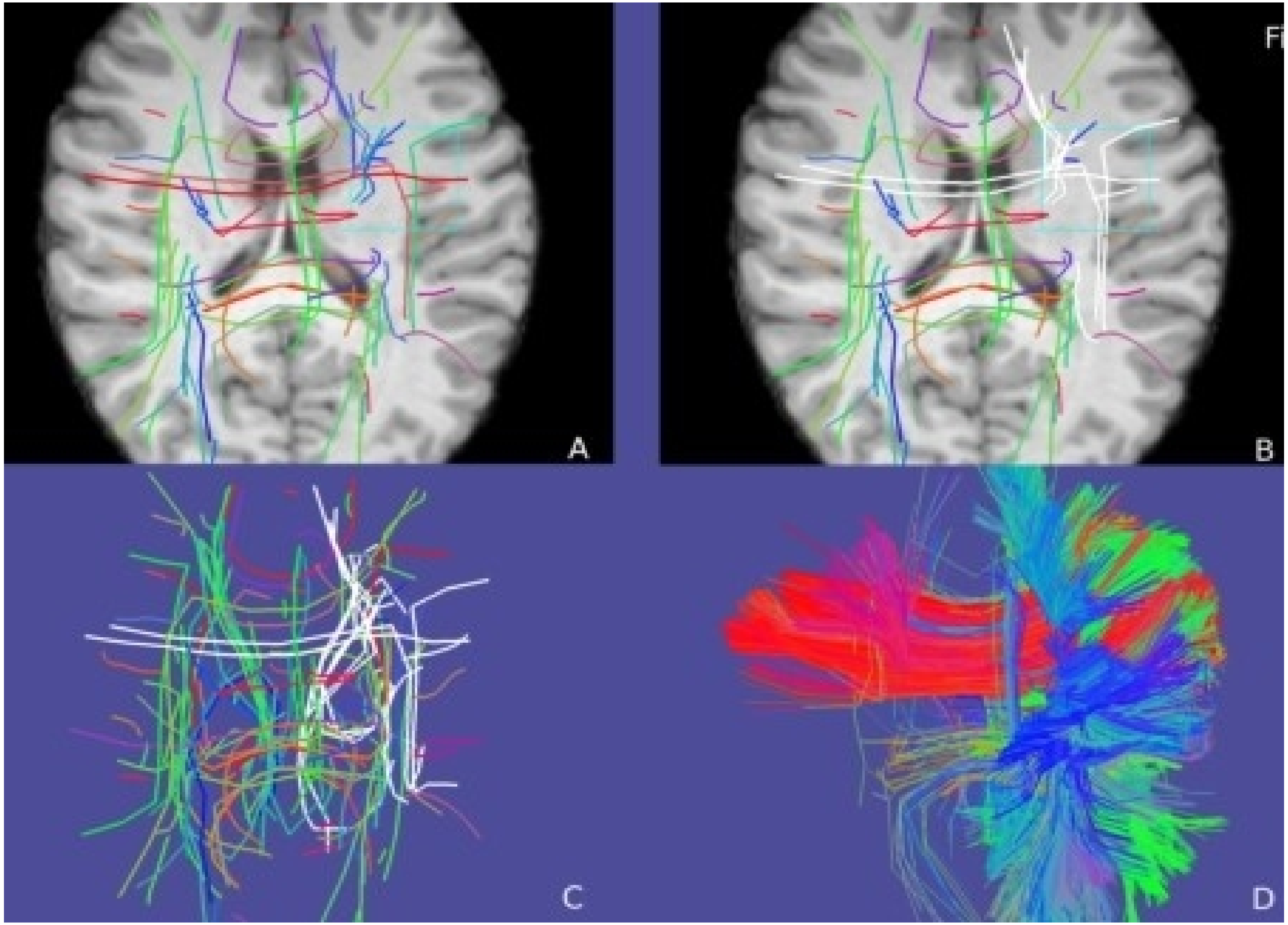
FFT: 15mins, SFF: 2 secs (one iteration)

Tractography of $3x10^5$ streamlines

E. Olivetti, **T. B. Nguyen**, E. Garyfallidis, *The Approximation of the Dissimilarity Projection*, PRNI 2012.

Spaghetti

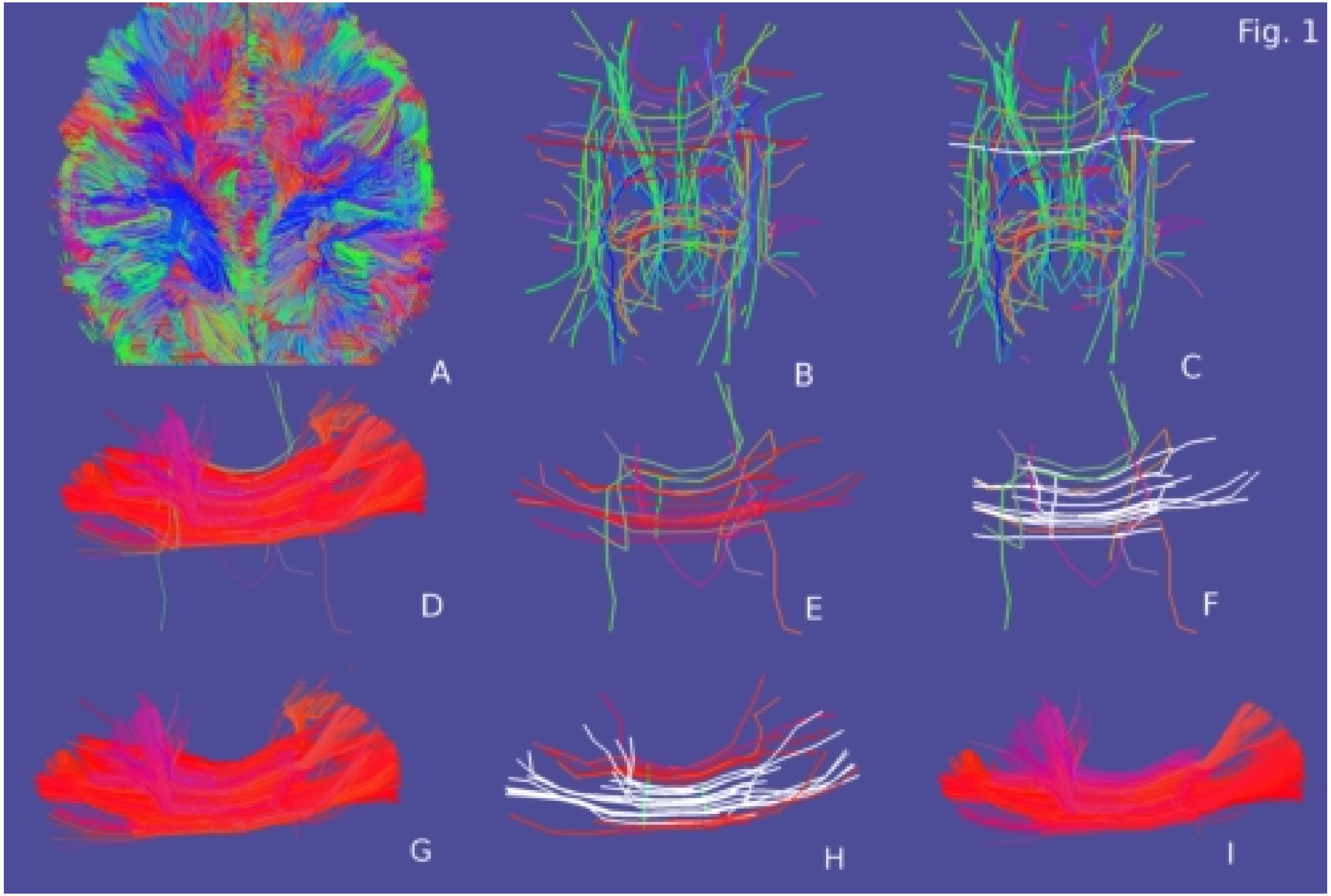
- An interactive visualization tool for segmentation tractography
- Rotate, hide, select, explore, zoom, "cluster", etc



E. Garyfallidis, S. Gerhard, P. Avesani, **T. B. Nguyen**, V. Tsiaras, I. N. Smith, and E. Olivetti, *A software application for real-time, clustering-based exploration of tractographies*, OHBM 2012.

Spaghetti

- Refinement, not support the tract candidate step
- "cluster" – recluster, not change partition



E. Garyfallidis, S. Gerhard, P. Avesani, **T. B. Nguyen**, V. Tsiaras, I. N. Smith, and E. Olivetti, *A software application for real-time, clustering-based exploration of tractographies*, OHBM 2012.

QuickBundles

- Trajectory clustering algorithm
- Inspired from BIRCH
- Linear time O(N)
- Online
- Low Memory Usage
- Simplifies Tractographies
- Every track belongs to one cluster only
-Useful for a huge number of applications

QB creates an online list of cluster nodes. The cluster node is defined as $c = \{I, \mathbf{h}, n\}$ where I is the list of the integer indices of the tracks in that cluster, $\mathbf{h}$ is an $p \times 3$ matrix which the most important feature of the cluster and n is the number of tracks on that cluster. $\mathbf{h}$ is a matrix which can be updated online when a track is added on a cluster and is equal to

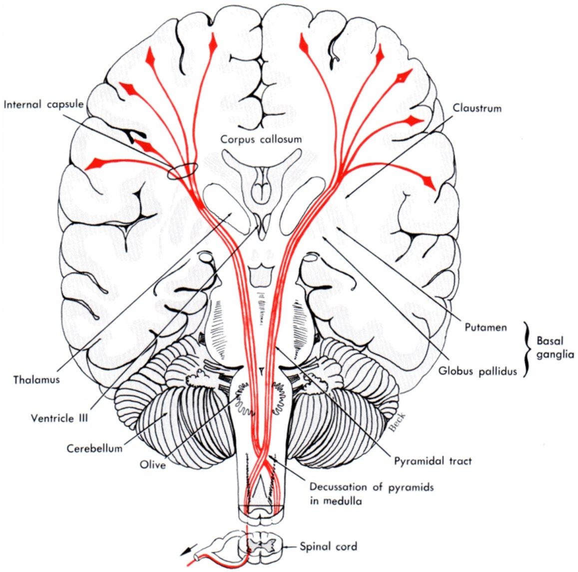
$$\mathbf{h} = \sum_{i=1}^{n} s_i$$

where s_i is the $p \times 3$ matrix representing track i, Σ represents here matrix addition along the second axis and n is the number of tracks in the cluster.

Case study: ALS disease

- Aim: CST difference between healthy and ALS (Amyotrophic Lateral Sclerosis) diseased brains
- Based on tractography approach

ALS disease



Cortical spinal tract (CST)

CST segmentation using Spaghetti

Case study: ALS disease

- Dataset: 12 subjects + 12 controls
- Features: streamline number, length, volume, density, FA (fractional anisotrophy), MD (mean diffusion)

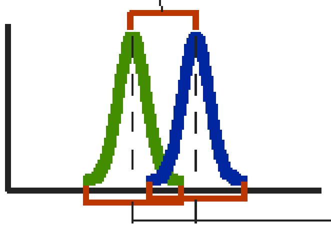
Features	Left		Right	
	t-score	p-value	t-score	p-value
Number	-0.9002843	0.3777187	**-4.1417273**	**0.0004268**
Mean length	-1.0671294	0.2974843	0.1987283	0.8443004
Volumn	0.6123652	0.5465741	**-3.2843068**	**0.0033857**
Density	-1.5767140	0.1291333	**-3.5359672**	**0.0018566**
Mean FA	-0.9798571	0.3378099	-0.1985348	0.8444499
Mean MD	-0.0132796	0.9895244	-0.0856670	0.9325060

t-test

t-score

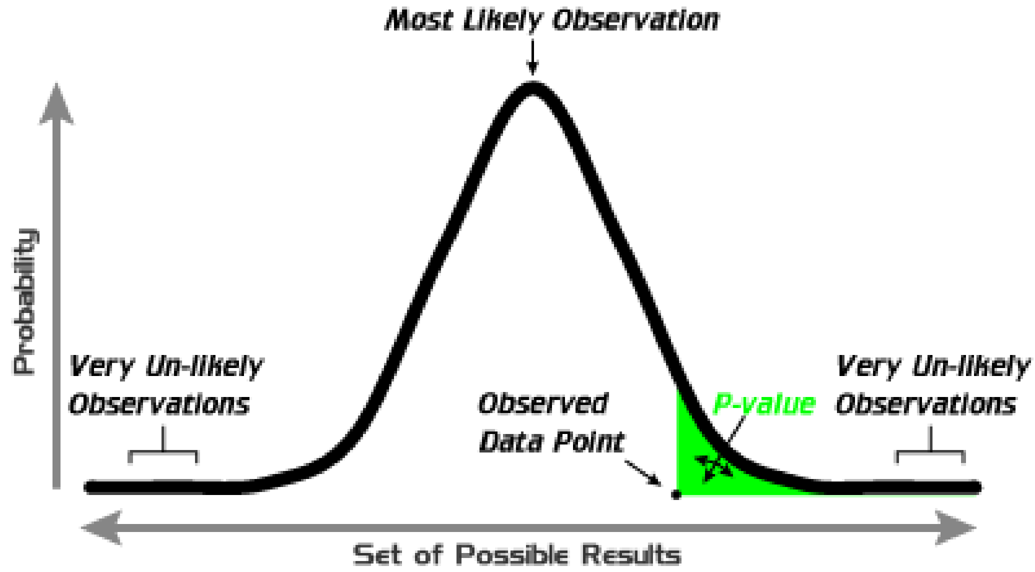
p-value

$$\frac{\text{signal}}{\text{noise}} = \frac{\text{difference between group means}}{\text{variability of groups}} = \frac{\bar{X}_T - \bar{X}_C}{SE(\bar{X}_T - \bar{X}_C)} = \text{t-value}$$



$$SE(\bar{X}_T - \bar{X}_C) = \sqrt{\frac{var_T}{n_T} + \frac{var_C}{n_C}}$$

$$t = \frac{\bar{X}_T - \bar{X}_C}{\sqrt{\frac{var_T}{n_T} + \frac{var_C}{n_C}}}$$



A p-value (shaded green area) is the probability of an observed (or more extreme) result arising by chance

p-value is the probability of obtaining a test statistic as the one that was actually observed, given the null hypothesis is true

Incremental clustering – SoA

Two strategies:

- Incremental clustering algorithm *(cat.1)*
  - directly handle dynamic changes *(hard)*
  - find place and do update on the dendrogram
  - NP problem
- Incremental data summarization *(cat.2)*
  - compress the data incrementally
  - apply standard clustering algorithm subsequently
  - exploit a wide range of current clustering algorithms
  - application driven

Incremental clustering

- Fisher et al., 1987 *(cat.1)*

Knowledge Acquisition Via Incremental Conceptual Clustering

  - COBWEB: an incremental hierarchical clustering
  - Node: probabilistic concept that summarizes the attribute-value distributions of all objects under the node
  - Top down strategy
  - At each node: chose action get the best clustering quality.
  - Action: Insert, Create, Merge, Split
  - Quality: CU (Category Utility)
  - Only consider normal distribution
  - Complexity: O(N * AVB2 logK)
    where B : branching factor(2-5), A : attributes, V : average number of values), K : number of classes/node
  - Each node: pairs (attribute, value)

Incremental clustering

- Ribert et al., 1999 *(cat.1)*

  An Incremental Hierarchical Clustering

  - Complexity: $O(N^3)$
  - ROI – region of influence
  - Obtain the same tree as the HAC on the union of the initial dataset and the new case
  - Considerably reduce memory usage

- Widyantoro et al., 2002 *(cat.1)*

  An incremental approach to building a cluster hierarchy

  - O(N log N) (not inluce "update MST") → O (N^2 log N)
  - Find place: closest – MST (minimum spanning tree)
  - Restructure: based on "density" property
  - Little sensative to input ordering

Incremental clustering *(con't)*

An incremental hierarchical data clustering algorithm based on gravity theory

- Chen et al., 2002 *(cat.1 + cat.2)*
  - GRACE: build a dendrogram, $O(N^2)$
    (gravity-based agglomerative hierarchical clustering)
  - From dendrogram, generate the tentative dendrogram
  - Insertion: $O(N)$ (or $O(N^2)$ in the worst case)
    1. Belong to leaf node → insert *(cat. 1)*
    2. No: in outlier buffer → exceed threshold, reapply GRACE *(cat.2)*
  - No deletion
  - Not sensative to input ordering
  - $O(N^3)$ (or $O(N^4)$ in the worst case)

Incremental clustering

Incremental and effective data summarization for dynamic hierarchical clustering

- Nassar et al., 2004 *(cat.2)*
  - Domain: KDD (knowledge discorvery in database)
  - Create dendrogram:
    1. Random s seed points
    2. Compress data: create s "data bubbles”
       B = (rep, n, extent, nnDist)
    3. Run hierarchical clustering on s ”bubles”
  - Update: measure 'bubble' quality
    → reassign points among ”bubbles”
  - Rerun clustering algorithm when the number of seed changes
  - O (s * N)

Incremental clustering *(con't)*

SIHC: A Stable Incremental Hierarchical Clustering Algorithm

- Gurrutxaga et al., 2009 *(cat. 1)*
  - Complexity: $O(N^2 \log N)$
  - Top-down strategy
  - Each node: height
  - Place: height(current node) < distance(new object, current node)
    1. Insert into current node
    2. Else: update height and travese the nearest child node
  - No deletions
  - Little sensative to input ordering
  - Keep the original structure

Incremental clustering *(con't)*

Efficient clustering approach using incremental and hierarchical clustering methods

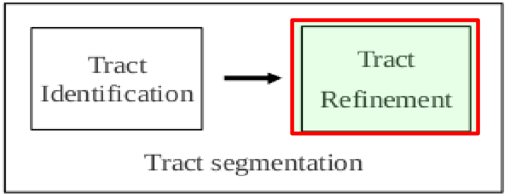
- Srinivas et al., 2010 *(not incremental)*
  - Two phases: LC and HAC
  - LC (leader clustering): create m sub-clusters
    - ✔ Complexity: best – O(N) or worst - $O(N^2)$
    - ✔ Similar to QB (QuickBundle)
  - HAC (hierarchical allogmerative clustering)
    - ✔ Complexity : $O(m^2 \log m)$
    - ✔ Bottom-up strategy
  - Similarity measurement: cohesion or joinability of two clusters

Incremental clustering *(con't)*

Document update summarization using incremental hierarchical clustering

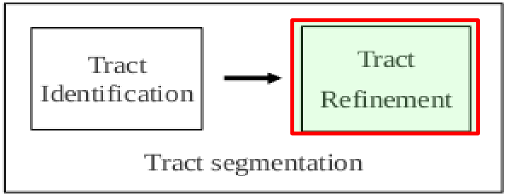
- Wang et al., 2010 *(cat.1)*
  - Two phases: summarization and clustering
  - Summarization: create sentence matrix of input document
    - Sentence: a set of pairs (term, frequency)
  - Clustering: COBWEB
    - Not normal distribution, but Katz distribution *(word occurence)*
  - Similar to original COBWEB, only use different distribution

Interactive clustering – challenges



- Intuitive display of both the hierarchy and the instances at the same time
- Maintain the history of operations
- Provide a stable structure of the dendrogram
  - Not drammatically change the previous one
  - User can assimilate the changes
- At an acceptable computational cost

Interactive clustering – close solutions



- Widyantoro et al., 2002 *(cat.1)*

  An incremental approach to building a cluster hierarchy

  O(N log N) v.s build MST (O(N^2 log N)) + not stable structure

- Gurrutxaga et al., 2009 *(cat. 1)*

  SIHC: A Stable Incremental Hierarchical Clustering Algorithm

  Keep a stable structure v.s O(N^2 log N)

- Nassar et al., 2004 *(cat.2)*

  Incremental and effective data summarization for dynamic hierarchical clustering

  O (m * N) v.s data summarization

Incremental clustering *(con't)*

Incremental hierarchical clustering of text documents (good)

- Sahoo et al., 2006 *(cat. 1)*
  - ...., $O(N^2)$
    (gravity-based agglomerative hierarchical clustering)
  - From dendrogram, generate tentative dendrogram
  - Insertion: $O(N)$ (or $O(N^2)$ in the worst case)
    i. Belong to leaf node → insert *(cat. 1)*
    ii. No: put in outlier buffer → exceed threshold, reapply GRACE *(cat. 2)*
  -
  - Not sensative to input ordering