UNIVERSITY OF COPENHAGEN

Extraction of Airways from Volumetric Data

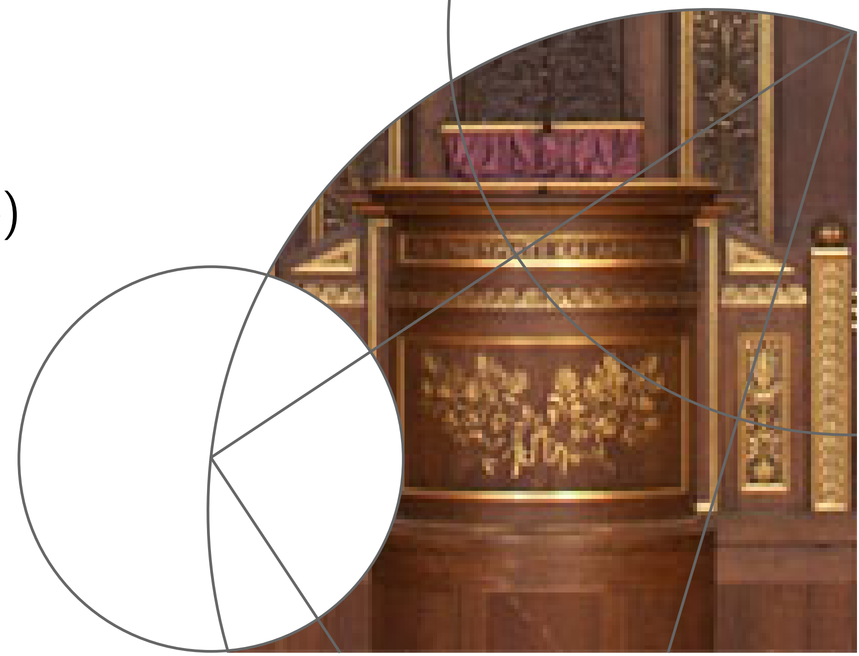
A graph refinement view

Raghavendra Selvan

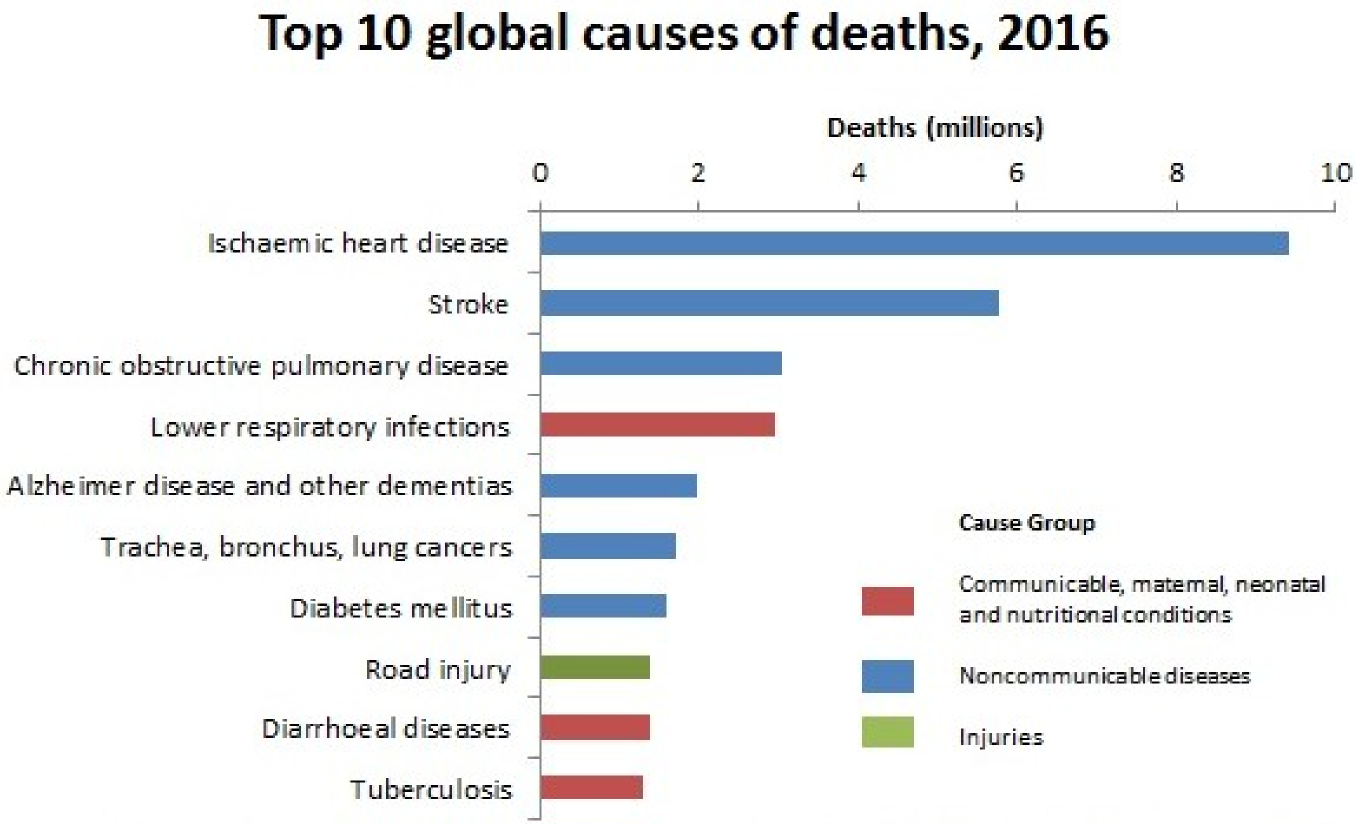
Assistant Professor
Kiehn Lab (Dept. of Neuroscience)
Machine Learning Section (Dept. of Computer Science)
University of Copenhagen

raghav@di.ku.dk
@raghavian
https://raghavian.github.io

December 16, 2020



Respiratory diseases: Major cause of morbidity & mortality



Source: Global Health Estimates 2016, World Health Organization, 2018

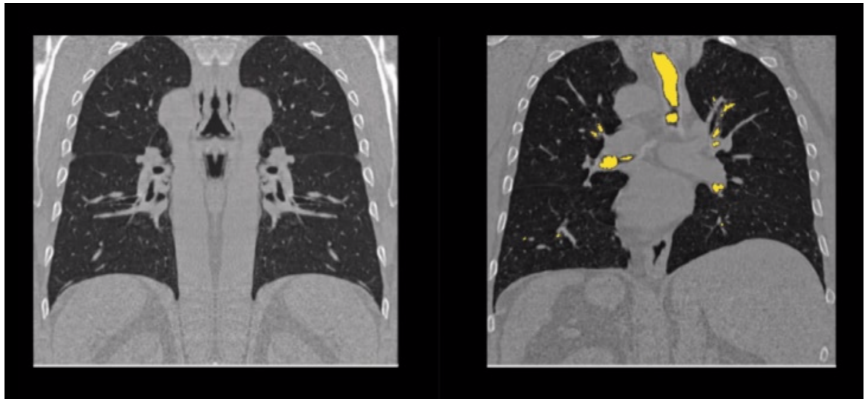
Outline

Outline

Imaging based analysis of airways & challenges

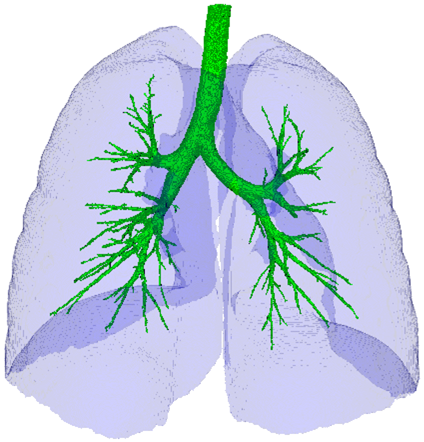
Three primary steps:

1. **Detection** of airways
2. Measurement of airway morphology
3. Deriving biomarkers



Coronal view of chest CT scan

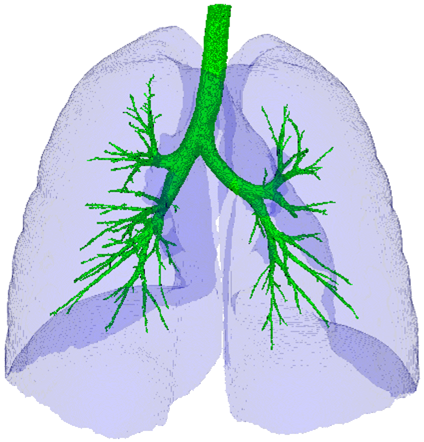
Methods exist. Majority of them are sequential



Sequential segmentation methods

Lo, P., et.al : Extraction of airways from CT (EXACT'09). IEEE Transactions on Medical Imaging, (2012)
Juarez AG, et al. Automatic airway segmentation in chest CT using convolutional neural networks. In Image Analysis for Moving Organ, Breast, and Thoracic Images 2018.

Methods exist. Majority of them are sequential

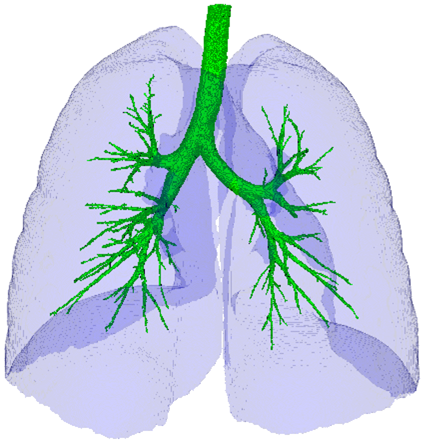


Sequential segmentation methods

- Susceptible to occlusions in data
- Small branches are challenging
- EXACT'09 Study
  - Airway extraction challenge
  - Compares 15 methods
  - 10 use region growing!
- Quite recently some U-net based attempts on patches

Lo, P., et.al : Extraction of airways from CT (EXACT'09). IEEE Transactions on Medical Imaging, (2012)
Juarez AG, et al. Automatic airway segmentation in chest CT using convolutional neural networks. In Image Analysis for Moving Organ, Breast, and Thoracic Images 2018.

Methods exist. Majority of them are sequential



Sequential segmentation methods

- Susceptible to occlusions in data
- Small branches are challenging
- EXACT'09 Study
  - Airway extraction challenge
  - Compares 15 methods
  - 10 use region growing!
- Quite recently some U-net based attempts on patches

Lo, P., et.al : Extraction of airways from CT (EXACT'09). IEEE Transactions on Medical Imaging, (2012)
Juarez AG, et al. Automatic airway segmentation in chest CT using convolutional neural networks. In Image Analysis for Moving Organ, Breast, and Thoracic Images 2018.

Objective of this work

Extraction of airways from volumetric data

With automatic methods that:

- Are exploratory
- Use more global information in local decisions

Outline

Data from Danish Lung Cancer Screening Trial (DLCST)

- $> 10,000$ Low-dose CT from 2052 subjects
- Smoker or former smoker (> 20 pack years)
- Voxels $\sim 0.75 \times 0.75 \times 1$ mm^3
- 32 scans with manual annotations for evaluation
- And additional 100 with automatic segmentations for hyperparameter tuning

Pedersen, J. H., et.al : The Danish randomized lung cancer CT screening trial – Overall design and results of the prevalence round. Journal of Thoracic Oncology, (2009)

Outline

Graph Refinement Models

Work based on

[1] Raghavendra Selvan, Thomas Kipf, Max Welling, Antonio GU Juarez, Jesper H. Pedersen, Jens Petersen, and Marleen de Bruijne. "Graph Refinement based airway extraction using Mean-Field Networks and Graph Neural Networks" Preprint/Medical Image Analysis (2018/2020).

[2] Raghavendra Selvan, Max Welling, Jesper H. Pedersen, Jens Petersen, and Marleen de Bruijne. "Mean field network based graph refinement with application to airway tree extraction." 21st Conference on Medical Image Computing & Computer Assisted Intervention (MICCAI 2018), pp. 750-758, Cham. Springer International Publishing.

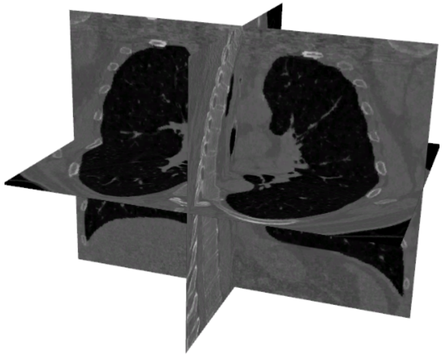
[3] Raghavendra Selvan, Thomas Kipf, Max Welling, Jesper H. Pedersen, Jens Petersen, and Marleen de Bruijne. "Extraction of Airways using Graph Neural Networks." 1st Conference on Medical Imaging with Deep Learning (MIDL 2018), Amsterdam.

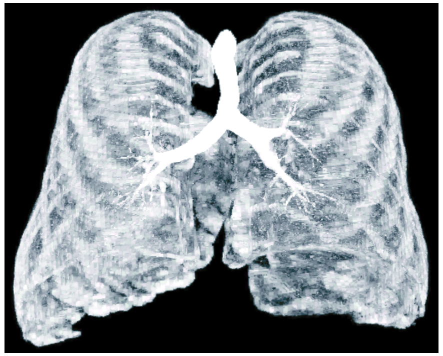
Graph Refinement Model for Airway Extraction

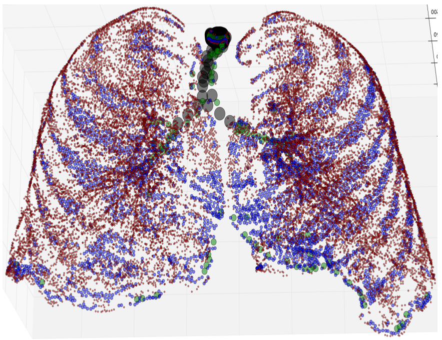
High level idea

- Assume over-connected graphs with node attributes
- Optimise global connectivity, instead of qualifying individual branches

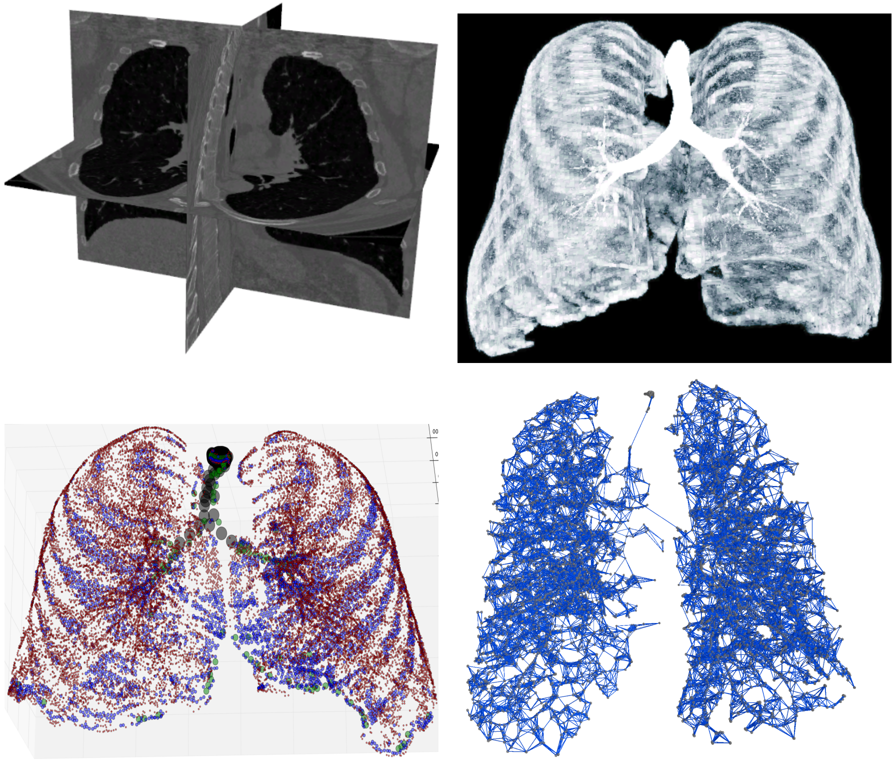
Volumetric data to Graph data



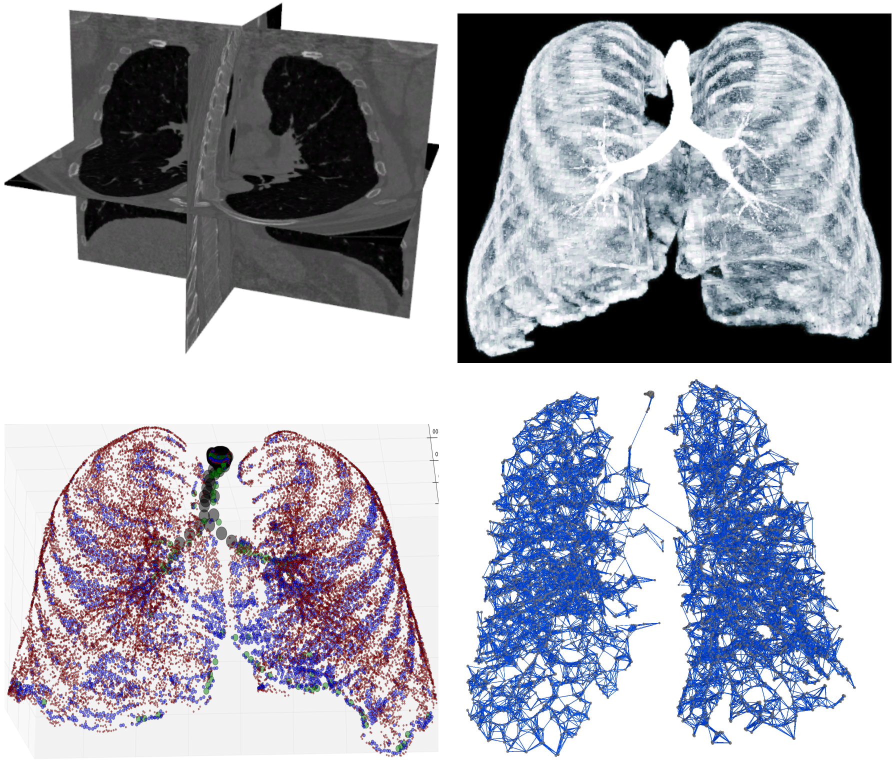




Volumetric data to Graph data



Volumetric data to Graph data



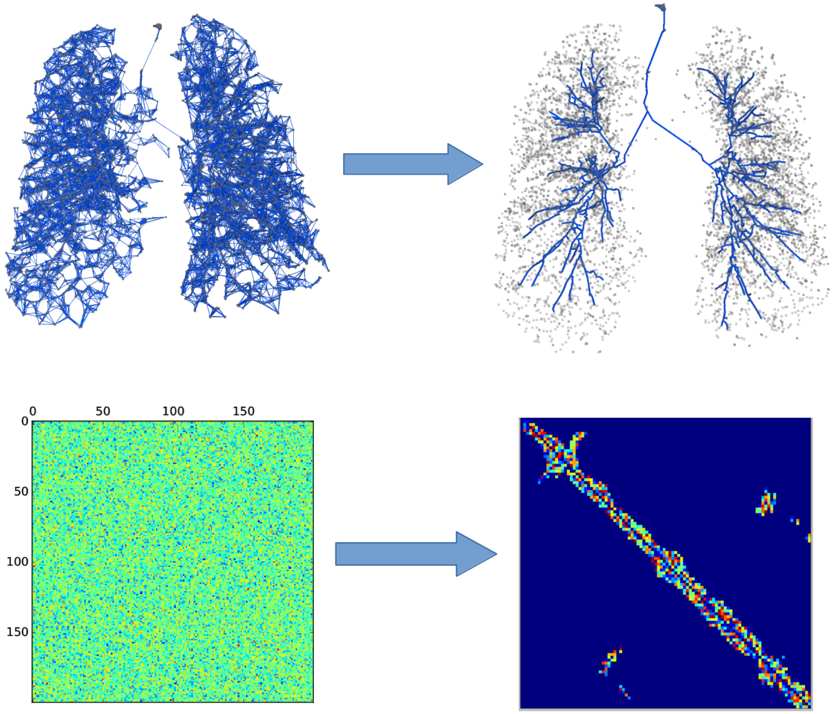
- Overconnected input graph: $\mathcal{G}_{\text{in}} : \{\mathcal{V}, \mathcal{E}_{\text{in}}\}$, with $|\mathcal{V}| = N$
- Node features: $\mathbf{X} \in \mathbb{R}^{F \times N}$
- Input adjacency: $\mathbf{A}_{\text{in}} \in \{0, 1\}^{N \times N}$

Airway extraction as Graph Refinement task

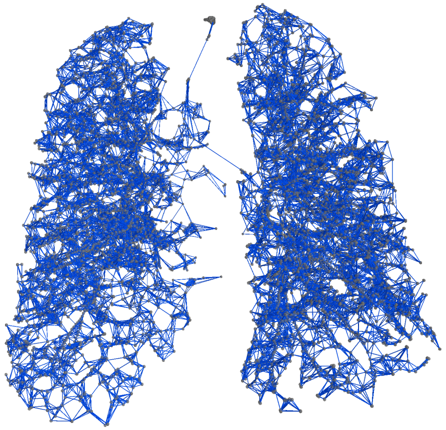
Graph Refinement Model

$$f(\cdot) : \mathcal{G}_{\text{in}} \mapsto \mathcal{G}$$

Output subgraph $\mathcal{G}$ with $\mathcal{E} \subset \mathcal{E}_{in}$; $\mathbf{A} \in \{0,1\}^{N\times N}$

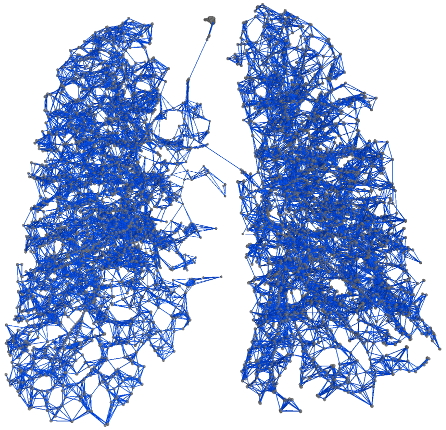


Probabilistic Graphical Model for MFN



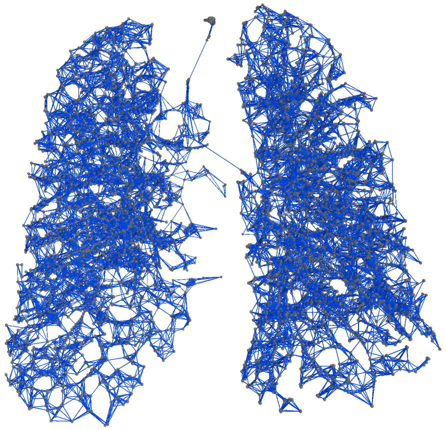
- Binary random variable $s_{ij} \in \{0, 1\}$ with prob. $\alpha_{ij} \in [0, 1]$

Probabilistic Graphical Model for MFN



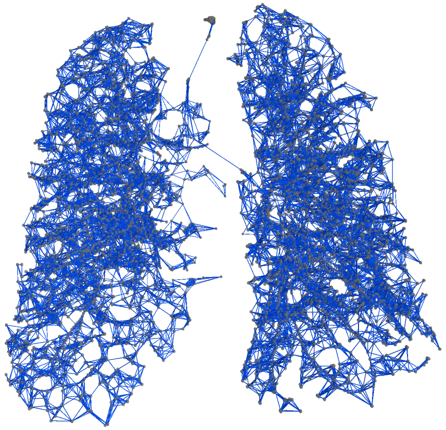
- Binary random variable $s_{ij} \in \{0, 1\}$ with prob. $\alpha_{ij} \in [0, 1]$
- For each node: $\mathbf{s}_i = \{s_{ij}\} : j = 1 \ldots N$

Probabilistic Graphical Model for MFN



- Binary random variable $s_{ij} \in \{0,1\}$ with prob. $\alpha_{ij} \in [0,1]$
- For each node: $\mathbf{s}_i = \{s_{ij}\} : j = 1 \ldots N$
- Global connectivity variable: $\mathbf{S} = [\mathbf{s}_1 \ldots \mathbf{s}_N]$
- Instances of $\mathbf{S}$ are $N \times N$ adjacency matrices

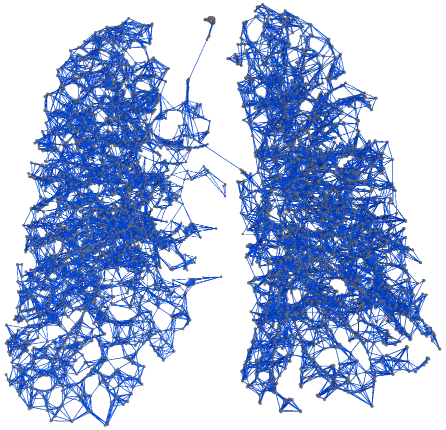
Probabilistic Graphical Model for MFN



- Binary random variable $s_{ij} \in \{0, 1\}$ with prob. $\alpha_{ij} \in [0, 1]$
- For each node: $\mathbf{s}_i = \{s_{ij}\} : j = 1 \dots N$
- Global connectivity variable: $\mathbf{S} = [\mathbf{s}_1 \dots \mathbf{s}_N]$
- Instances of $\mathbf{S}$ are $N \times N$ adjacency matrices

Posterior density of interest: $p(\mathbf{S}|\mathbf{X}, \mathbf{A}_{\text{in}})$

Probabilistic Graphical Model for MFN



- Binary random variable $s_{ij} \in \{0, 1\}$ with prob. $\alpha_{ij} \in [0, 1]$
- For each node: $\mathbf{s}_i = \{s_{ij}\} : j = 1 \dots N$
- Global connectivity variable: $\mathbf{S} = [\mathbf{s}_1 \dots \mathbf{s}_N]$
- Instances of $\mathbf{S}$ are $N \times N$ adjacency matrices

Posterior density of interest: $p(\mathbf{S}|\mathbf{X}, \mathbf{A}_{\text{in}})$

$$\begin{aligned}\ln p(\mathbf{S}|\mathbf{X}, \mathbf{A}_{\text{in}}) &\propto \ln p(\mathbf{S}, \mathbf{X}, \mathbf{A}_{\text{in}}) \\ &= \sum_{i \in \mathcal{N}} \phi_i(\mathbf{s}_i) + \sum_{(i,j) \in \mathcal{E}} \phi_{ij}(\mathbf{s}_i, \mathbf{s}_j) - \ln Z,\end{aligned}$$

Node and Pairwise Potentials for MFA

Node Potential: For each node $i \in \mathcal{V}$

$$\phi_i(\mathbf{s}_i) = \sum_{v=0}^{D} \beta_v \mathbb{I}\Big[\sum_j s_{ij} = v\Big] + \mathbf{a}^T \mathbf{x}_i \sum_j s_{ij}, \tag{5}$$

Pairwise Potential: For each edge,$(i, j) \in \mathcal{E}_{\text{in}}$

$$\phi_{ij}(\mathbf{s}_i, \mathbf{s}_j) = \lambda(1 - 2|s_{ij} - s_{ji}|) + (2 s_{ij} s_{ji} - 1)\Big[\boldsymbol{\eta}^T |\mathbf{x}_i - \mathbf{x}_j| + \boldsymbol{\nu}^T (\mathbf{x}_i \mathbf{x}_j)\Big]. \tag{6}$$

Parameters $= [\cdot]$

Node and Pairwise Potentials for MFA

Node Potential: For each node $i \in \mathcal{V}$

$$\phi_i(\mathbf{s}_i) = \sum_{v=0}^{D} \beta_v \mathbb{I}\left[\sum_j s_{ij} = v\right] + \mathbf{a}^T \mathbf{x}_i \sum_j s_{ij}, \tag{5}$$

Pairwise Potential: For each edge,$(i,j) \in \mathcal{E}_{\text{in}}$

$$\phi_{ij}(\mathbf{s}_i, \mathbf{s}_j) = \lambda(1 - 2|s_{ij} - s_{ji}|) + (2s_{ij}s_{ji} - 1)\left[\boldsymbol{\eta}^T |\mathbf{x}_i - \mathbf{x}_j| + \boldsymbol{\nu}^T (\mathbf{x}_i \mathbf{x}_j)\right]. \tag{6}$$

$$\text{Parameters} = [\beta]$$

Node and Pairwise Potentials for MFA

Node Potential: For each node $i \in \mathcal{V}$

$$\phi_i(\mathbf{s}_i) = \sum_{v=0}^{D} \beta_v \mathbb{I}\Big[\sum_j s_{ij} = v\Big] + \mathbf{a}^T \mathbf{x}_i \sum_j s_{ij}, \tag{5}$$

Pairwise Potential: For each edge,$(i, j) \in \mathcal{E}_{\text{in}}$

$$\phi_{ij}(\mathbf{s}_i, \mathbf{s}_j) = \lambda(1 - 2|s_{ij} - s_{ji}|) + (2 s_{ij} s_{ji} - 1)\Big[\boldsymbol{\eta}^T |\mathbf{x}_i - \mathbf{x}_j| + \boldsymbol{\nu}^T (\mathbf{x}_i \mathbf{x}_j)\Big]. \tag{6}$$

$$\text{Parameters} = [\boldsymbol{\beta}, \mathbf{a}]$$

Node and Pairwise Potentials for MFA

Node Potential: For each node $i \in \mathcal{V}$

$$\phi_i(\mathbf{s}_i) = \sum_{v=0}^{D} \beta_v \mathbb{I}\Big[\sum_j s_{ij} = v\Big] + \mathbf{a}^T \mathbf{x}_i \sum_j s_{ij}, \tag{5}$$

Pairwise Potential: For each edge, $(i,j) \in \mathcal{E}_{\text{in}}$

$$\phi_{ij}(\mathbf{s}_i, \mathbf{s}_j) = \lambda\big(1 - 2|s_{ij} - s_{ji}|\big) + (2 s_{ij} s_{ji} - 1)\Big[\boldsymbol{\eta}^T |\mathbf{x}_i - \mathbf{x}_j| + \boldsymbol{\nu}^T (\mathbf{x}_i \mathbf{x}_j)\Big]. \tag{6}$$

$$\text{Parameters} = [\boldsymbol{\beta}, \mathbf{a}, \lambda]$$

Node and Pairwise Potentials for MFA

Node Potential: For each node $i \in \mathcal{V}$

$$\phi_i(\mathbf{s}_i) = \sum_{v=0}^{D} \beta_v \mathbb{I}\left[\sum_j s_{ij} = v\right] + \mathbf{a}^T \mathbf{x}_i \sum_j s_{ij}, \tag{5}$$

Pairwise Potential: For each edge,$(i,j) \in \mathcal{E}_{\text{in}}$

$$\phi_{ij}(\mathbf{s}_i, \mathbf{s}_j) = \lambda\big(1 - 2|s_{ij} - s_{ji}|\big) + (2s_{ij}s_{ji} - 1)\Big[\boldsymbol{\eta}^T|\mathbf{x}_i - \mathbf{x}_j| + \boldsymbol{\nu}^T(\mathbf{x}_i\mathbf{x}_j)\Big]. \tag{6}$$

$$\text{Parameters} = [\boldsymbol{\beta}, \mathbf{a}, \lambda, \boldsymbol{\eta}, \boldsymbol{\nu}]$$

Approximate posterior density with a simpler one

Approximate posterior density with a simpler one

Mean-Field Factorisation: $q(\mathbf{S}) \in \mathcal{Q}$

$$q(\mathbf{S}) = \prod_{i=1}^{N} \prod_{j=1}^{N} q_{ij}(s_{ij}), \tag{1}$$

Implication: Node connectivities are independent.

Approximate posterior density with a simpler one

Mean-Field Factorisation: $q(\mathbf{S}) \in \mathcal{Q}$

$$q(\mathbf{S}) = \prod_{i=1}^{N} \prod_{j=1}^{N} q_{ij}(s_{ij}), \tag{1}$$

Implication: Node connectivities are independent.

Variational Inference to approximate $p(\mathbf{S}|\mathbf{X}, \mathbf{A}_{\text{in}})$

$$p(\mathbf{S}|\mathbf{X}, \mathbf{A}_{\text{in}}) \approx q(\mathbf{S}) \tag{2}$$

Approximate posterior density with a simpler one

Mean-Field Factorisation: $q(\mathbf{S}) \in \mathcal{Q}$

$$q(\mathbf{S}) = \prod_{i=1}^{N} \prod_{j=1}^{N} q_{ij}(s_{ij}), \tag{1}$$

Implication: Node connectivities are independent.

Variational Inference to approximate $p(\mathbf{S}|\mathbf{X}, \mathbf{A}_{in})$

$$p(\mathbf{S}|\mathbf{X}, \mathbf{A}_{in}) \approx q(\mathbf{S}) \tag{2}$$

Minimize KL Divergence $\equiv$ Maximize Evidence Lower Bound (ELBO)

$$\text{ELBO}(q) = -\text{KLD}(q(\mathbf{S})||p(\mathbf{S}|\mathbf{X}, \mathbf{A}_{in})\Big] + \ln Z \tag{3}$$

Maximising ELBO wrt $q_{ij}(s_{ij})$ yields MFA Iterations

MFA Iterations

$$\alpha_{kl}^{(t+1)} = q_{kl}^{(t+1)}(s_{kl} == 1)$$
$$= \frac{1}{1 + \exp^{-\gamma_{kl}}}$$

$\forall\ k = \{1 \dots N\},\ l \in \mathcal{N}_k$

$\boldsymbol{\alpha}$: Global connectivity prediction

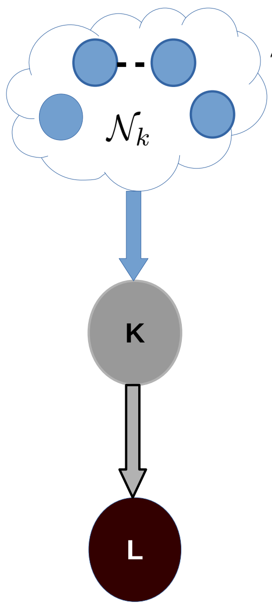
Maximising ELBO wrt $q_{ij}(s_{ij})$ yields MFA Iterations

MFA Iterations

$$\alpha_{kl}^{(t+1)} = q_{kl}^{(t+1)}(s_{kl} == 1) = \frac{1}{1 + \exp^{-\gamma_{kl}}}$$

$\forall\ k = \{1 \dots N\},\ l \in \mathcal{N}_k$

$\boldsymbol{\alpha}$: Global connectivity prediction



$$\gamma_{kl} = \prod_{j \in \mathcal{N}_k \setminus l} \left(1 - \alpha_{kj}^{(t)}\right) \Bigg\{ \sum_{m \in \mathcal{N}_k \setminus l} \frac{\alpha_{km}^{(t)}}{(1 - \alpha_{km}^{(t)})} \Bigg[(\beta_2 - \beta_1) - \beta_2 \sum_{n \in \mathcal{N}_k \setminus l,m} \frac{\alpha_{kn}^{(t)}}{(1 - \alpha_{kn}^{(t)})} \Bigg] + \left(\beta_1 - \beta_0\right) \Bigg\} + \mathbf{a}^T \mathbf{x}_i + (4\alpha_{lk}^{(t)} - 2)\lambda + 2\alpha_{lk}^{(t)} \left(\boldsymbol{\eta}^T |\mathbf{x}_i - \mathbf{x}_j| + \boldsymbol{\nu}^T (\mathbf{x}_i \mathbf{x}_j) \right)$$

$$\alpha_{kl}^{(t+1)} = \sigma(\gamma_{kl}) = \frac{1}{1 + \exp^{-\gamma_{kl}}}$$

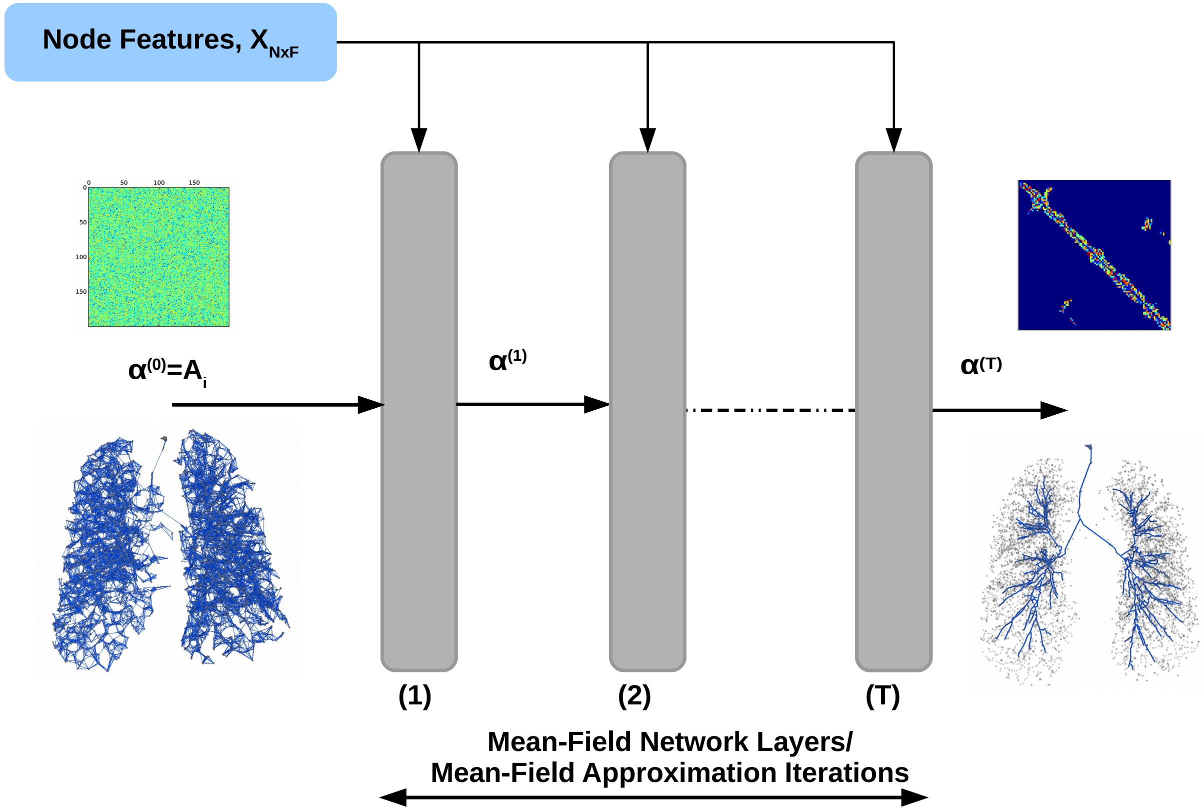
Note: MFA iterations resemble feed-forward operations in neural nets

MFA as Mean-Field Networks

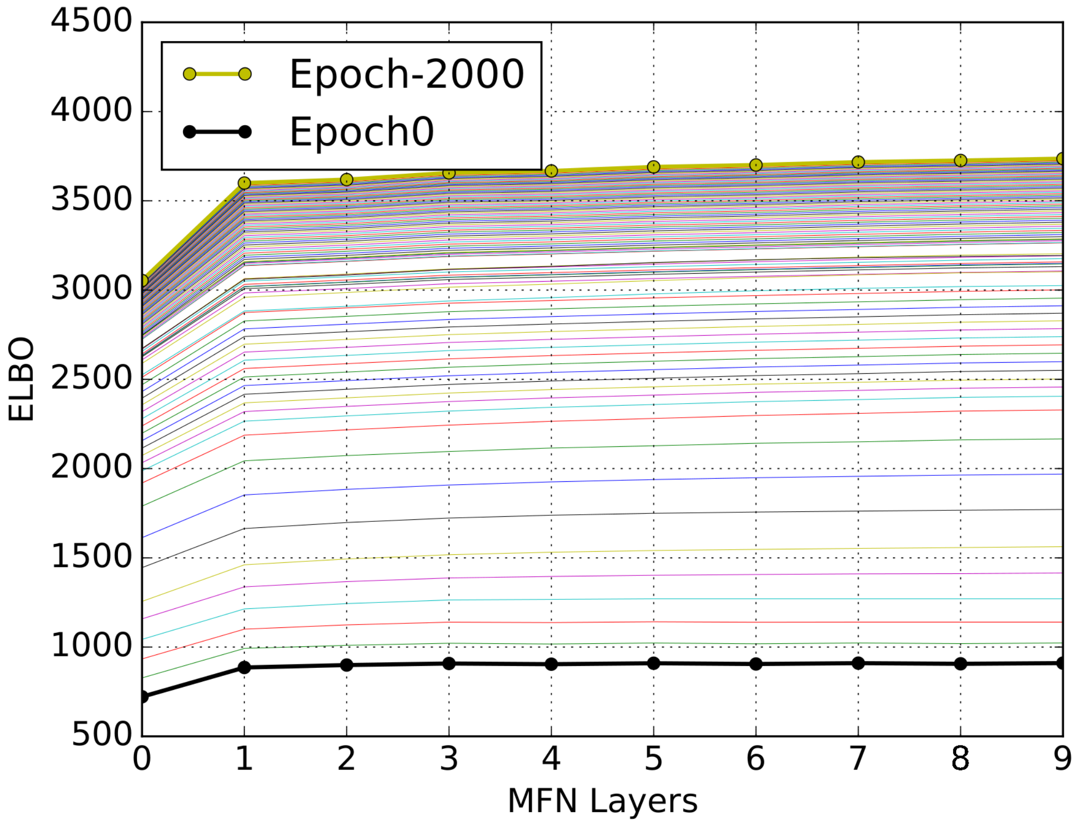
- $T-$iterations as a $T-$layered network

MFA as Mean-Field Networks

- $T-$iterations as a $T-$layered network
- Gradient descent to learn model parameters: $\mathcal{L}(\boldsymbol{\alpha}, \mathbf{A}_r)$



Increasing ELBO $\Longrightarrow$ Better approximation



Experiments

- **Baseline:** a) Region growing on probability images b) Bayesian smoothing merged with region growing for evaluation
- Pretraining dataset used to tune hyperparameters
- Eight-fold cross validation

Experiments

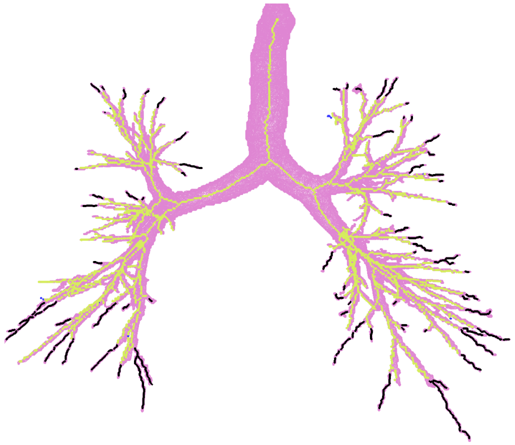
- **Baseline:** a) Region growing on probability images b) Bayesian smoothing merged with region growing for evaluation
- Pretraining dataset used to tune hyperparameters
- Eight-fold cross validation
- **Error measures**:
  - Average centerline distance: $d_{err} = (d_{FP} + d_{FN})/2$
  - $d_{FP} \equiv$ Specificity
  - $d_{FN} \equiv$ Sensitivity
  - Percentage of tree length (TL)
  - False positive rate (FPR)

Performance comparison

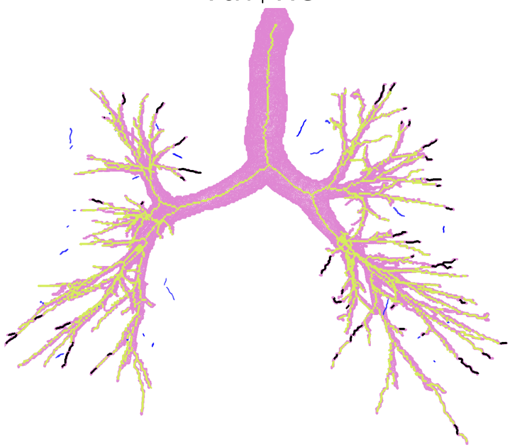
	d_{FP}(mm)	d_{FN}(mm)	d_{err} (mm)	TL(%)	FPR(%)
Vox+RG	3.624 ± 0.776	5.155 ± 0.580	4.389 ± 0.441	79.6 ± 7.2	5.0 ± 3.9
BS+RG	3.921 ± 0.612	4.218 ± 0.334	4.069 ± 0.476	82.3 ± 6.1	8.7 ± 3.4
MFN	3.599 ± 0.583	3.491 ± 0.295	3.595 ± 0.321	83.1 ± 6.7	8.6 ± 5.3

- d_{FP} $\equiv$ Specificity
- d_{FN} $\equiv$ Sensitivity
- Average centerline distance: d_{err}
- Percentage of tree length (TL)
- False positive rate (FPR)

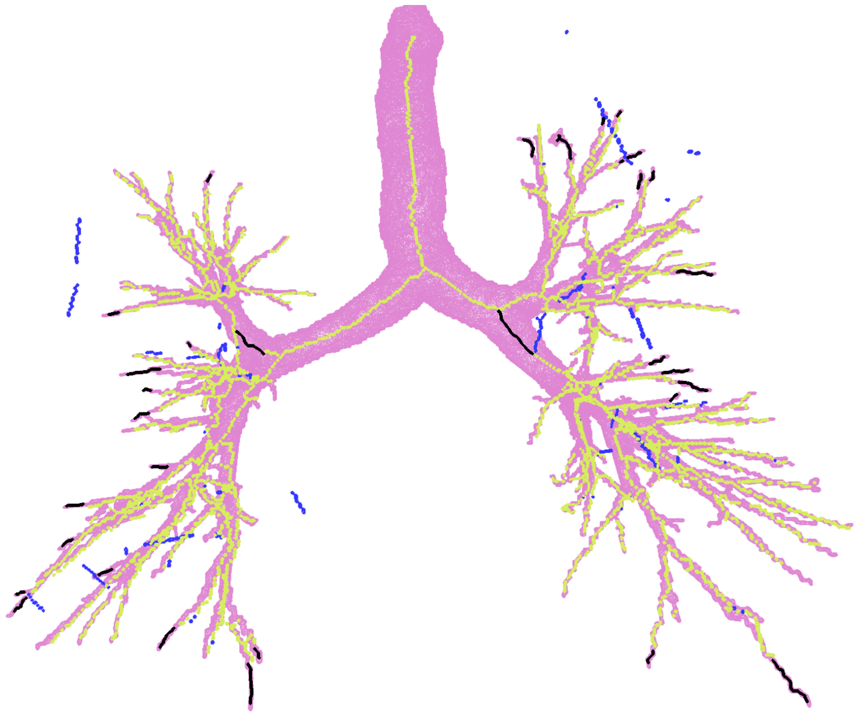
Visualisation of extracted airways



Vox+RG



BS+RG



MFN

Legend: Reference (pink), True Positive (Yellow), False Negative (Black), False Positive (Blue)

Summary

- Airway extraction as graph refinement
- Novel use of Mean-Field Approximation
- Proposed expressive node and pairwise potentials
- Mean-Field Network interpretation
- Few parameters (46 scalar weights)
- Easy to optimise using gradient descent

– Might not generalise across applications

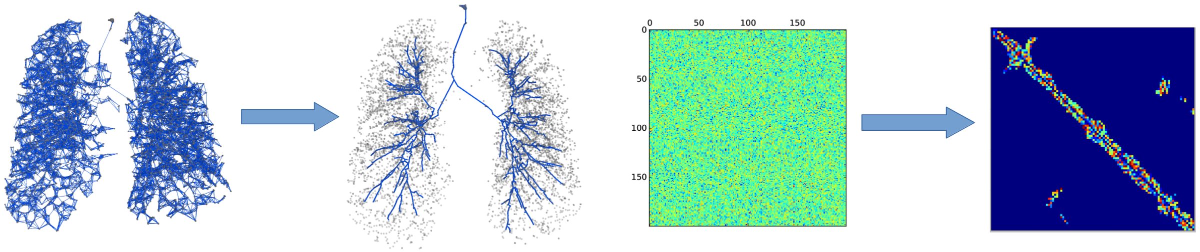
– Hand-crafting potentials is cumbersome

Graph Neural Networks

Graph Neural Networks

- Neural nets with graph input
- Step towards non-Euclidean (geometric) Deep Learning
- Generalisation of message passing algorithms
- Complex task-specific messages can be learnt
- End-to-end trainable inference systems

GNN based Graph Refinement



- Graph refinement task: $f(\cdot) : \mathcal{G}_{\text{in}} \mapsto \mathcal{G}$
- GNN based encoder-decoder pair
- Encoder comprises stacks of GNNs; Message passing between nodes
- Joint training of encoder-decoder pair to learn useful embeddings
- Simple decoder predicts graph connectivity

GNN Model for Graph Refinement

GNN Model for Graph Refinement

Consider node j with neighbours $\mathcal{N}_j$,

$$\text{Node Embedding:} \quad \mathbf{h}_j^1 = g_n(\mathbf{x}_j) \tag{8}$$

$$\text{N2E mapping:} \quad \mathbf{h}_{(i,j)}^1 = g_{n2e}([\mathbf{h}_i^1, \mathbf{h}_j^1]) \tag{9}$$

$$\text{E2E mapping:} \quad \mathbf{h}_j^2 = g_{e2n}(\sum \mathbf{h}_{(i,j)}^1]) \quad \forall i \in \mathcal{N}_j \tag{10}$$

$$\text{N2E mapping:} \quad \mathbf{h}_{(i,j)}^2 = g_{n2e}([\mathbf{h}_i^2, \mathbf{h}_j^2]) \tag{11}$$

$$\textbf{Decoder:} \quad \alpha_{ij} = \sigma(g_{dec}(\mathbf{h}_{(i,j)}^2)) \tag{12}$$

$g_{\dots}(\cdot)$ are MLPs, g_{dec} is MLP with 1 output channel

GNN Model for Graph Refinement

Consider node j with neighbours $\mathcal{N}_j$,

$$\text{Node Embedding:} \quad \mathbf{h}_j^1 = g_n(\mathbf{x}_j) \tag{8}$$

$$\text{N2E mapping:} \quad \mathbf{h}_{(i,j)}^1 = g_{n2e}([\mathbf{h}_i^1, \mathbf{h}_j^1]) \tag{9}$$

$$\text{E2E mapping:} \quad \mathbf{h}_j^2 = g_{e2n}(\sum \mathbf{h}_{(i,j)}^1]) \quad \forall i \in \mathcal{N}_j \tag{10}$$

$$\text{N2E mapping:} \quad \mathbf{h}_{(i,j)}^2 = g_{n2e}([\mathbf{h}_i^2, \mathbf{h}_j^2]) \tag{11}$$

$$\textbf{Decoder:} \quad \alpha_{ij} = \sigma(g_{dec}(\mathbf{h}_{(i,j)}^2)) \tag{12}$$

$g_{\dots}(\cdot)$ are MLPs, g_{dec} is MLP with 1 output channel

GNN Model for Graph Refinement

Consider node j with neighbours $\mathcal{N}_j$,

$$\text{Node Embedding:} \quad \mathbf{h}_j^1 = g_n(\mathbf{x}_j) \tag{8}$$

$$\text{N2E mapping:} \quad \mathbf{h}_{(i,j)}^1 = g_{n2e}([\mathbf{h}_i^1, \mathbf{h}_j^1]) \tag{9}$$

$$\text{E2N mapping:} \quad \mathbf{h}_j^2 = g_{e2n}(\sum \mathbf{h}_{(i,j)}^1]) \quad \forall i \in \mathcal{N}_j \tag{10}$$

$$\text{N2E mapping:} \quad \mathbf{h}_{(i,j)}^2 = g_{n2e}([\mathbf{h}_i^2, \mathbf{h}_j^2]) \tag{11}$$

$$\textbf{Decoder:} \quad \alpha_{ij} = \sigma(g_{dec}(\mathbf{h}_{(i,j)}^2)) \tag{12}$$

$g_{\dots}(\cdot)$ are MLPs, g_{dec} is MLP with 1 output channel

GNN Model for Graph Refinement

Consider node j with neighbours $\mathcal{N}_j$,

$$\text{Node Embedding:} \quad \mathbf{h}_j^1 = g_n(\mathbf{x}_j) \tag{8}$$

$$\text{N2E mapping:} \quad \mathbf{h}_{(i,j)}^1 = g_{n2e}([\mathbf{h}_i^1, \mathbf{h}_j^1]) \tag{9}$$

$$\text{E2N mapping:} \quad \mathbf{h}_j^2 = g_{e2n}(\sum \mathbf{h}_{(i,j)}^1]) \quad \forall i \in \mathcal{N}_j \tag{10}$$

$$\text{N2E mapping:} \quad \mathbf{h}_{(i,j)}^2 = g_{n2e}([\mathbf{h}_i^2, \mathbf{h}_j^2]) \tag{11}$$

$$\textbf{Decoder:} \quad \alpha_{ij} = \sigma(g_{dec}(\mathbf{h}_{(i,j)}^2)) \tag{12}$$

$g_{\dots}(\cdot)$ are MLPs, g_{dec} is MLP with 1 output channel

GNN Model for Graph Refinement

Consider node j with neighbours $\mathcal{N}_j$,

$$\text{Node Embedding:} \quad \mathbf{h}_j^1 = g_n(\mathbf{x}_j) \tag{8}$$

$$\text{N2E mapping:} \quad \mathbf{h}_{(i,j)}^1 = g_{n2e}([\mathbf{h}_i^1, \mathbf{h}_j^1]) \tag{9}$$

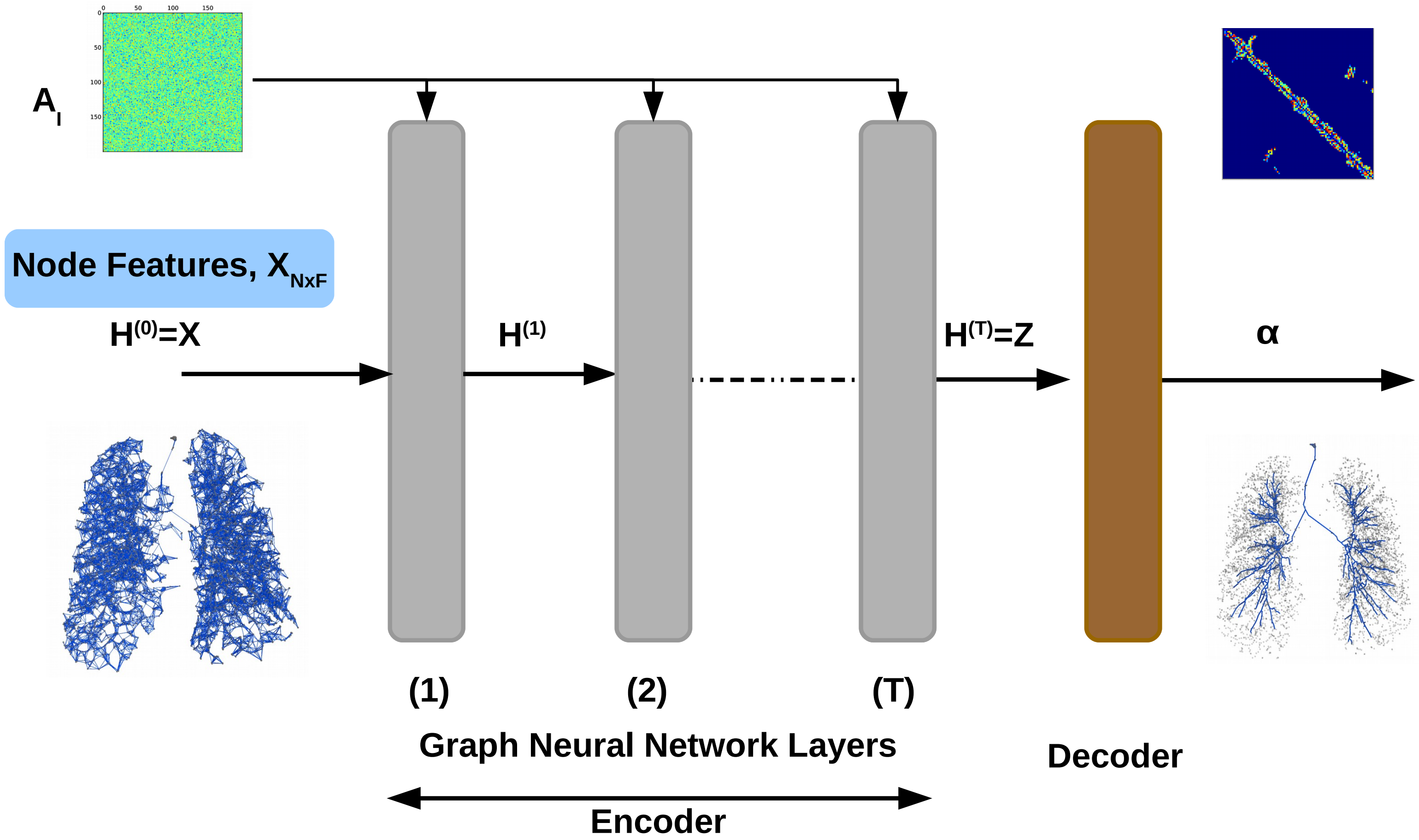
$$\text{E2N mapping:} \quad \mathbf{h}_j^2 = g_{e2n}(\sum \mathbf{h}_{(i,j)}^1]) \quad \forall i \in \mathcal{N}_j \tag{10}$$

$$\text{N2E mapping:} \quad \mathbf{h}_{(i,j)}^2 = g_{n2e}([\mathbf{h}_i^2, \mathbf{h}_j^2]) \tag{11}$$

$$\textbf{Decoder:} \quad \alpha_{ij} = \sigma(g_{dec}(\mathbf{h}_{(i,j)}^2)) \tag{12}$$

$g_{...}(\cdot)$ are MLPs, g_{dec} is MLP with 1 output channel

Summarising GNN Model



Experiments

- Same set-up as with MFNs
- Pretraining dataset used to tune hyperparameters
- Eight fold cross validation

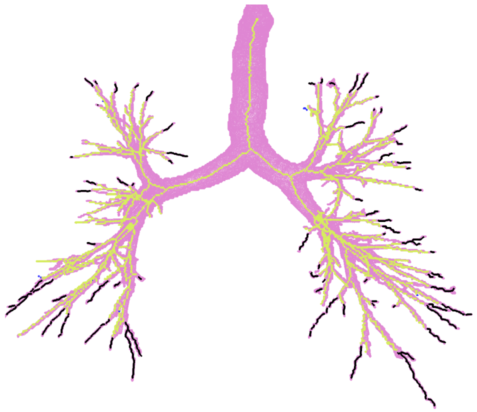
Performance comparison

Table 1
Performance comparison of five methods: Region growing on probability images (Vox+RG), Bayesian smoothing merged with Vox+RG (BS+RG), UNet, MFN and GNN models. Dice similarity, centerline distances (d_{FP}, d_{FN}, d_{err}), fraction of tree length detected (TL) and false positive rate (FPR) are reported based on 8−fold cross validation. Significant improvements when compared to other methods are shown in boldface. Additionally, we also report the running time to train each of the models in a single fold. Note that the MFN and GNN models require additional preprocessing that is performed only once when preparing the graphs..

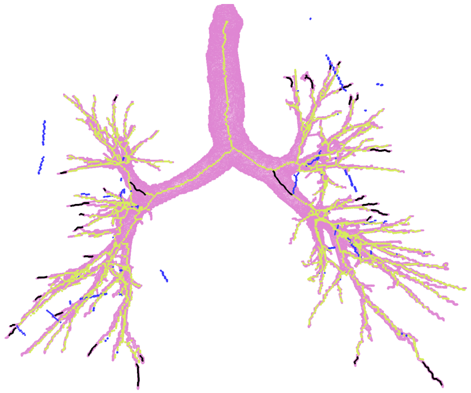
	Dice(%)	d_{FP}(mm)	d_{FN}(mm)	d_{err} (mm)	TL(%)	FPR(%)	Time (m)
Vox+RG	–	2.937 ± 1.005	6.762 ± 2.1042	4.847 ± 2.527	73.2 ± 9.9	4.9 ± 3.9	90
BS+RG	–	2.827 ± 1.266	4.601 ± 2.002	3.714 ± 1.896	73.6 ± 6.1	7.9 ± 6.1	105
UNet	–	3.540 ± 1.316	3.525 ± 1.201	3.532 ± 1.259	75.6 ± 8.7	6.5 ± 3.3	5700
MFN	86.5 ± 2.5	3.608 ± 1.360	3.116 ± 0.632	3.362 ± 1.297	74.5 ± 6.7	8.6 ± 5.4	60 + 35
GNN	84.8 ± 3.3	**2.216 ± 0.464**	**2.878 ± 0.505**	**2.547 ± 0.587**	**81.9 ± 7.3**	7.8 ± 4.6	60 + 12

- $d_{FP} \equiv$ Specificity
- $d_{FN} \equiv$ Sensitivity
- Average centerline distance: d_{err}
- Percentage of tree length (TL)
- False positive rate (FPR)

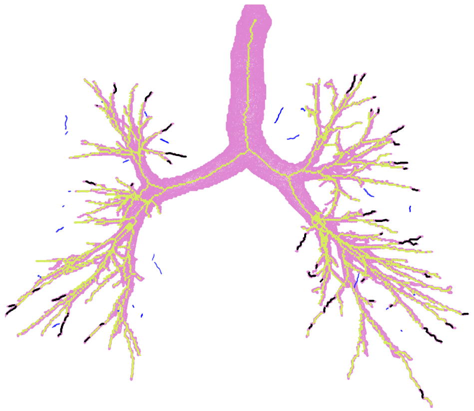
Visualisation of extracted airways



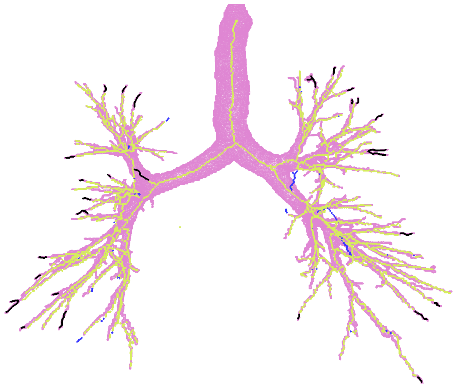
Vox+RG



MFN



BS+RG



GNN

Legend: Reference (pink), True Positive (Yellow), False Negative (Black), False Positive (Blue)

Summary

- GNN based supervised graph refinement
- Unique, inductive graph application of GNNs
- Edge embeddings used for prediction
- Competitive results with limited data
- Generalisations of MFNs
  - Disconnected trees
  - Relies on quality labelled training data

Outline

Conclusions from the study

- Exploratory methods can extract more branches
- Graph based representations are less computationally intensive
- Using global information in local decisions is helpful
- Incorporating prior knowledge is valuable
- MFNs as structured neural networks
- GNNs as generalisations of message passing algorithms
- Bias-variance trade-off between MFNs and GNNs

Acknowledgements

- Max Welling, UvA
- Thomas Kipf, UvA
- BIGR group, Erasmus MC

Independent Research Fund Denmark (DFF)

Netherlands Organisation for Scientific Research

Questions?

Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models

Lasse F. Wolff Anthony* [1] Benjamin Kanding* [1] Raghavendra Selvan [1]

https://github.com/lfwa/carbontracker/

raghav@di.ku.dk