Article:

Bayesian Polytrees with Learned Deep Features for Multi-Class Cell Segmentation

Hamid Fehri, *Student Member, IEEE*, Ali Gooya, *Member, IEEE*, Yuanjun Lu, Erik Meijering, *Fellow, IEEE*,
Simon A. Johnston, Alejandro F. Frangi *Fellow, IEEE*

Abstract—The recognition of different cell compartments, types of cells, and their interactions is a critical aspect of quantitative cell biology. However, automating this problem has proven to be non-trivial, and requires solving multi-class image segmentation tasks that are challenging owing to the high similarity of objects from different classes and irregularly shaped structures. To alleviate this, graphical models are useful due to their ability to make use of prior knowledge and model inter-class dependencies. Directed acyclic graphs, such as trees have been widely used to model top-down statistical dependencies as a prior for improved image segmentation. However, using trees, a few inter-class constraints can be captured. To overcome this limitation, we propose polytree graphical models that capture label proximity relations more naturally compared to tree based approaches. A novel recursive mechanism based on two-pass message passing was developed to efficiently calculate closed-form posteriors of graph nodes on polytrees. The algorithm is evaluated on simulated data and on two publicly available fluorescence microscopy datasets, outperforming directed trees and a state-of-the-art convolutional neural network architecture, namely SegNet. Two types of features were used to explore the role of features in segmentation: 1) scale-space differential invariants, 2) deep representations extracted by SegNet. Polytrees are shown to outperform directed trees in predicting segmentation error, by highlighting areas in the segmented image that do not comply with prior knowledge. This paves the way to uncertainty measures on the resulting segmentation and guides subsequent segmentation refinement.

Index Terms—hierarchical graphs, cell and nucleus segmentation, multi-class segmentation, segmentation error prediction

I. Introduction

Accurate and efficient image segmentation of complex spatial object arrangements composed of multiple constituting structures (or classes) is challenging yet paramount for biological discoveries underpinned by quantitative imaging. For example, the identification of different cells within tissue [1] or organelles within cells [2], the sub-cellular localization of proteins [3], the interactions of different cell types in organ development [4], or the immune response during infection [5], are just a few relevant problems in biology. To assess the morphological and behavioral characteristics of these cells (some having unknown causes [6]), quantitative metrics are devised, which require image segmentation as an unavoidable first step. Additionally, histology images are increasingly used for disease diagnosis and grading. Quantitative analysis of these images through the developed metrics (e.g. for abnormal nuclei as a potential indicator of cancer) helps pathologists by providing a supporting diagnosis and disease progress evaluation [7], [8]. Still, at a finer resolution, the biology of cell nucleus, i.e. the organization of the genome and the proteins, has a functional relevance with the biological cell processes, and their mis-localization (hence segmentation) can be a valuable indicator for many pathologies [9], [10], [11]. Given that all the above mentioned examples are multi-class segmentation problems, automatic methods are of high significance due to their labor-intensity, and inter- and intra-observer variability of manual analysis, especially for large datasets. However, common features of these images, such as defused or overlapping boundaries, irregular shapes and high deformability of objects, limited resolution and quality in biological images, may contribute to the poor segmentation performance of automatic methods.

Incorporation of prior knowledge can play an important role in aiding and improving segmentation. Inter-object dependencies have been used in the segmentation of interacting objects [12] and intra-object spatial relationships were shown to enhance the segmentation of cell organelles [13]. Other examples in cell segmentation include using priors to consider the relative topology of cells and nuclei [14], [15], and to impose area and size constraints on segmented regions [16], to achieve a better segmentation. In brain tissue analysis, appearance and spatial priors have been used to improve tumor localization [17], generalization to unseen images [18], and lesion recognition as atypical brain tissues [19].

Graphical models enable modeling associative relations between random variables [20]. These probabilistic models can improve segmentation by imposing constraints emerged from the prior knowledge [21], [22]. The key aspect of graphical models is that the label of each node is determined based on both its own attributes and attributes of other nodes connected

H. Fehri is with the Center for Computational Imaging Simulation Technologies in Biomedicine (CISTIB), The University of Sheffield, Sheffield, United Kingdom. He is also with the Bateson Center, Firth Court, University of Sheffield and the Department of Infection, Immunity and Cardiovascular Disease, Medical School, University of Sheffield. (e-mail: h.fehri@sheffield.ac.uk)

A. Gooya, Y. Lu and A. F. Frangi are with the Center for Computational Imaging Simulation Technologies in Biomedicine (CISTIB), The University of Sheffield, Sheffield, United Kingdom. (email: a.gooya@sheffield.ac.uk; yuanjun.lu@sheffield.ac.uk; a.frangi@sheffield.ac.uk)

E. Meijering is with the Biomedical Imaging Group Rotterdam, Erasmus University Medical Center, 3015 GE Rotterdam, The Netherlands. (email: meijering@imagescience.org)

S. A. Johnston is with the Bateson Center, Firth Court, University of Sheffield and the Department of Infection, Immunity and Cardiovascular Disease, Medical School, University of Sheffield. (e-mail: s.a.johnston@sheffield.ac.uk)

SAJ and AFF are joint senior authors.

through graph edges. This way, not only all the information is incorporated in inferring the labels, but also label configuration constraints can be effectively applied during inference. For instance, Chen *et al.* [23] employed graphical models to incorporate nuclear positions with boundary information for yeast cell segmentation. In a rather different application, segmentation of retinal images, graphical models have been used for combining appearance models with shape priors [24].

We propose polytree graphical models for implementation of local label configurations for multi-class segmentation problems. Polytrees are a type of Bayesian Networks (BNs) whose nodes can have more than one parent. Compared to other well-known BNs based on trees [25], [26], [27], [28], where each node only has one parent, polytrees can capture more complex configurations and constraints. This higher flexibility of polytrees also inhibits certain unfeasible label cliques on the graph that trees are unequipped to exclude, in spite of their contravening prior knowledge. The performance of the proposed method was compared to that of the directed trees and a convolutional neural network to assess the modeling and error prediction efficiencies.

II. RELATED WORK

Two types of graphical models have been used for image segmentation, namely Markov Random Fields (MRFs) and Bayesian Networks (BNs). MRFs have weighted edges indicating dependencies between variables and are used for capturing correlations between random variables. Directed edges in BNs indicate causal relationships between random variables [29]. In this paper, we focus on BNs and enforce the constraints using conditional probabilities that appear in the joint probability distribution. To find the optimal labels of the graph, different inference algorithms have been proposed. Two-pass inference algorithms were initially proposed for chain-based models, which calculate exact probabilities for node labels [30]. Extension of this forward-backward algorithm, known as belief propagation [31], [32], resulted in exact solutions for two main types of Directed Acyclic Graphs (DAGs): *trees* and *polytrees*. Directed trees are BNs with only one route between each pair of nodes in the graph (i.e. singly connected [20]), with each node, except the root node, having exactly one parent node. Polytrees, however, are singly connected BNs where each node can have more than one parent node. Existing solutions for these two DAGs factorize the posterior of each node into two factors: a marginal posterior given a subset of observations, and a subgraph data likelihood given the label of the node [32]. Despite their being exact and non-iterative, the dependency on the likelihood function in these factorizations makes the numerical implementation impractical [33]. This is because probability values become very small at some nodes, where the likelihoods involve a large number of data components, hence causing arithmetic underflow.

To address the implementation problem of the proposed algorithms for inference, Laferte *et al.* [33] designed a recursive framework that calculates exact posteriors of nodes on a regular quadtree, based on posteriors of its neighboring nodes. Feng *et al.* [34] used Tree-Structured Belief Networks
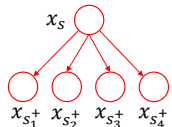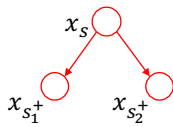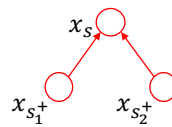
(TSBNs) as a prior model combined with a neural network for local prediction of class labels. TSBNs suffer from block artifacts [35] resulting from the descendants of a node s on a tree being conditionally independent, given the state of s. More complex graph structures, such as overlapping trees [36] whose nodes do not point to distinct areas of the image, and two-dimensional trees [37] have been proposed to reduce this effect at the expense of higher computational costs. Alternatively, a group of authors proposed trees with dynamic structures fitting the image contents (e.g. [38], [39]) where the labels and the graph structure are inferred. Priors have also been incorporated into trees using Hierarchically-Structured Interacting Segments (HINTS) [12], where the nodes represent interacting segments in the image. Iterative algorithms were proposed for approximating the optimal label configurations for binary [40] and multi-label [41] cases. However, the proposed optimization algorithms do not always converge and may require modifying the graph structure or relaxing the constraints for convergence.

To address the limitations of trees, which mainly stem from the independence of same-level nodes [34], we propose polytrees for multi-class image segmentation. Compared to trees, polytrees can eliminate a wider range of unfeasible label configurations, by modeling both inter- and intra-level dependencies. Similar to the work of Laferte *et al.* [33], we derive a two-pass inference algorithm on the polytree for exact calculation of posterior probabilities on the graph. The proposed polytree based method is evaluated by segmenting objects from multiple classes in real microscopy images. We show it outperforms methods based on the state-of-the-art convolutional neural networks, viz. SegNet [42], and directed trees.

The proposed model is also evaluated on its ability to predict segmentation error. Areas of the segmented image that do not comply with the imposed priors are nominated and their similarity to the actual segmentation error is measured. Polytrees are shown to outperform trees in finding the wrongly segmented areas.

Our polytree based segmentation method is fundamentally different from the method proposed by Laferte *et al.* [33] and entails important extensions. Our hierarchical graph structure (both in polytrees and trees) is made based on an initial super-pixelation step [43], and subsequently merges the most similar superpixels (graph nodes) until the highest level. The graph structure is asymmetric and irregular. This property allows capturing more natural cell boundaries for a more complex implementation. Conversely, Laferte *et al.* use symmetric and regular quadtrees, where the nodes are represented by square regions. The shapes of the nodes do not match the actual morphologies of the cells, rendering the method unsuitable for comparison. Inference-wise, our method uses features extracted by convolutional neural networks (CNN) (details explained in section IV-E) and is applied to supervised multi-class image segmentation, while Laferte *et al.* use pre-defined intensity and texture features for an EM-based unsupervised image classification. See Table I for a summary of fine differences between the three mentioned methods.

TABLE I
SUMMARY OF KEY DIFFERENCES BETWEEN LAFERTE *et al.* METHOD AND THE PROPOSED TREE AND POLYTREE

Method	Laferte *et al.*	Proposed tree	Proposed polytree
Number of descendants	4	2	2
Hierarchical structure	Regular	Irregular	Irregular
Features	Intensity and texture	CNN	Intensity features/ CNN
Application	Unsupervised segmentation	Supervised segmentation	Supervised segmentation
Sample constructing element	x_s $x_{s_1^+}$ $x_{s_2^+}$ $x_{s_3^+}$ $x_{s_4^+}$	x_s $x_{s_1^+}$ $x_{s_2^+}$	x_s $x_{s_1^+}$ $x_{s_2^+}$

III. METHOD

We present here our proposed graphical model for image segmentation. First, a polytree is generated for the image by grouping similar pixels and regarding them as nodes in the graph. Next, the parameters of the likelihood functions are trained and labels of the nodes are inferred. Finally, the segmented image is constructed based on the optimal labels on the graph.

A. Graph generation

Initially, the graph is generated by grouping pixels into locally coherent areas (superpixels), each representing a single root node (Fig. 1). We use the SEEDS algorithm [43], which refines an initial grid of identically block shaped superpixels into more coherent ones. The two most similar superpixels are then recursively merged to generate higher-level nodes in the graph hierarchy, in a similar manner to generating a merge-tree [44].

For each superpixel at the finest level, one (root) node in the lowest level of the graph is created (see Fig. 1). Every two nodes achieving highest scores according to a similarity metric are then merged to create a new super node. The new super node is the union of image regions attached to its two lower-level descendant nodes. We define the similarity metric as a superposition of distances using spatial and intensity features of the superpixels. A vector $\boldsymbol{\beta} = [\boldsymbol{\beta}_s; \boldsymbol{\beta}_i]$ is introduced to adjust contributions of each feature in the similarity metric. An adaptive scheme is designed for setting $\boldsymbol{\beta}$, which helps in the generation of more meaningful nodes in the graph. Nodes in the lower levels of the graph represent subregions of objects, rather than their full areas. For these nodes, we set $\boldsymbol{\beta}$ such that $\boldsymbol{\beta}_s$ consists of greater values compared to $\boldsymbol{\beta}_i$. This makes merging neighboring nodes that correspond to parts of the same object (i.e. a cell or a nucleus in our case) more probable. In the higher levels, however, values of $\boldsymbol{\beta}_i$ are set to be greater than those of $\boldsymbol{\beta}_s$ to facilitate the merging of regions belonging to the same class, although they might not be neighbors. Assuming $\boldsymbol{\beta}_i = \beta_i \mathbf{1}$ and $\boldsymbol{\beta}_s = \beta_s \mathbf{1}$ and setting $\beta_i = 1$ for simplicity, $\boldsymbol{\beta}$ is determined by a cross validation merely on β_s.

After each merging step, the new node and all the other *orphan* nodes, are assessed with the similarity metric to recognize candidate nodes for merging next. Region merging is continued until only two orphan nodes remain in the graph,
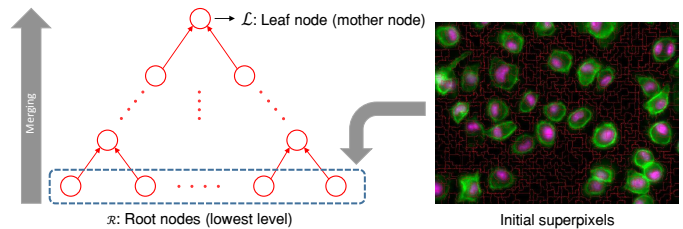

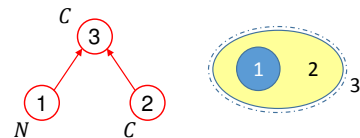Fig. 1. Generating a polytree from an oversegmented input image.

Fig. 2. A symbolic process of node merging for a synthetic cell (C) with a nucleus (N) resulting a polytree constructing element.
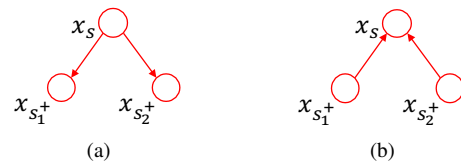

Fig. 3. Edge directions on cliques in (a) directed tree and (b) polytree graphical models.

which are eventually merged to create the leaf node that corresponds to the whole image (Fig. 1). Since two nodes are merged at each step of the graph evolution, the resulting structure is a binary graph; i.e. each non-root node has two descendant nodes directly connected to it. We call this three-wise structure a *clique* and denote it by $parent1 - child - parent2$.

Figure 2 shows a symbolic process of merging for a cell (C) with a nucleus (N). Here, nodes 1 and 2 align with blue and yellow areas in the synthetic cell, respectively. If these two nodes are chosen to be merged based on their value in the similarity metric, node 3 is generated, which corresponds to the union of blue and yellow areas annotated by the dashed ellipse. This clique is represented by $1 - 3 - 2$.

B. Graph definition

The generated graph is a hierarchical structure modeling the interrelations between areas corresponding to different classes. Nodes in the lower levels correspond to smaller superpixels, such as sub-areas of cells, and are therefore more homogeneous. Higher-level nodes correspond to one or multiple objects that can be of different classes. The hierarchical structure allows merging smaller areas from the same class (in the lower levels), and embedding of objects within larger regions with different classes (in the higher levels) according to certain merging rules. These rules are introduced in the model by defining and applying priors on label configurations. Denoting the graph by $\mathcal{G}$, the latent variable for node s is labeled as x_s, where $x_s \in \mathbb{X}$ ($\mathbb{X}$ being the set of all latent variable nodes), and takes a discrete value from the label set Λ. The observation nodes $\boldsymbol{y}_s$ are attached to each x_s node in the graph and contain feature vectors extracted from its corresponding superpixel in the image. With these definitions, segmenting the image equals inferring labels x_s given the observations $\boldsymbol{y}_s \in \mathbb{Y}$ ($\mathbb{Y}$ being the set of all observation nodes), where the label configurations comply with the prior imposed on the model.

C. Imposing priors on the graph

Applying inclusion-based prior knowledge is the main advantage of using hierarchical graphs and is a way to constrain the solution to plausible results. In a directed graphical model, prior knowledge can be modeled through setting specific forms of the conditional probabilities that implement causality according to the directions of the edges. These probabilities act as the prior factor in the Bayesian factorization of the posterior.

In directed trees, the joint probability consists of one-to-one priors that can only model across-level dependencies. For instance, in the constructing element of a dyadic tree depicted in Fig. 3a (excluding the observation nodes temporarily for simplicity) the joint probability is written as $p(\mathbb{X}) = p(x_{s_1^+}|x_s)p(x_{s_2^+}|x_s)p(x_s)$, where $p(x_{s_1^+}|x_s)$ and $p(x_{s_2^+}|x_s)$ are the one-to-one priors. In polytrees however, the joint probability has multiple-to-one priors modeling both across-level and same-level dependencies. The joint probability for the sample polytree structure of Fig. 3b factorizes as $p(\mathbb{X}) = p(x_s|x_{s_1^+}, x_{s_2^+})p(x_{s_1^+})p(x_{s_2^+})$, in which the factor $p(x_s|x_{s_1^+}, x_{s_2^+})$ is the prior. To show how this can influence the modeling ability of the hierarchy, imagine the label set consists of two classes: $\Lambda = \{A, B\}$. Also, assume $B-A-A$ is a feasible and $B-A-B$ is an unfeasible configuration. Using trees, $B-A-A$ is allowed by setting probabilities $p(x_{s_i^+} = B|x_s = A)$ and $p(x_{s_i^+} = A|x_s = A)$ to non-zero values. However, enforcing the former constraint also makes $B-A-B$ cliques feasible, even though they are to be prevented by the model. But thanks to the more complex priors in the polytree, setting $p(x_s = A|x_{s_1^+} = B, x_{s_2^+} = A)$ to non-zero values and setting $p(x_s = A|x_{s_1^+} = B, x_{s_2^+} = B)$ to zero satisfies both of the constraints with no conflicts. This simple example shows the advantage of polytrees over directed trees in modeling more complex problems, by using a larger number of parameters.

In this paper, we use the generated polytree (details explained in section III-A) to segment the image by inferring the optimal labels for latent variable nodes. Each node at the lowest graph level (finest image resolution) is a root (in contrast to the single root node in directed trees) and there is only one leaf node (see Fig. 1).

Figure 4 shows the tables of priors and possible label configurations, when three classes of background (B), cell (C) and nucleus (N) exist in the image, which is useful for the problem of segmenting cells and nuclei in the images.

D. Label inference

Let $\mathbb{X} = \{x_s\}$ and $\mathbb{Y} = \{\boldsymbol{y}_s\}$ denote sets of labels (latent variables) and the corresponding observed features at nodes, respectively, $\mathcal{G}$ denote the set of nodes and edges and $x_s \in \Lambda$, where Λ is the set of all possible labels. For an internal node (neither in the lowest level nor the leaf node) s in the graph, s^-, s^+ and s' denote nodes in higher, lower and same layers, respectively (Fig. 5a).

We now derive equations governing the posterior probabilities of graph nodes. Given the observed data $\mathbb{Y}$, finding the best segmentation equals the best configuration of labels $\mathbb{X}$ for the graph. Bayesian inference associates the most probable label from the set of possible labels Λ, given all observations:

$$\forall s \in \mathcal{G}, \hat{x}_s = \arg \max_{x_s \in \Lambda} p(x_s|\mathbb{Y}) \quad (1)$$

A new set of equations is derived to calculate the closed-form posterior probabilities at each node in the polytree. The inference algorithm calculates the posteriors of the nodes in two passes. These two consist of a pass from the leaf to the roots, (*top-down* pass), and another from the roots to the leaf (*bottom-up* pass).

The probability of a node's label x_s, given all data $\mathbb{Y}$, is computed by marginalizing the probability of the clique over two parent nodes s_1^+ and s_2^+ given $\mathbb{Y}$, and the *joint posterior* is given by

$$p(x_s|\mathbb{Y}) = \sum_{x_{s_1^+}, x_{s_2^+}} p(x_s, x_{s_1^+}, x_{s_2^+}|\mathbb{Y}) \quad (2)$$

Three-wise constraints on cliques appear in the posterior calculation. To factorize the joint probability, we need a mechanism to identify the dependency of the nodes in the graph.

D-separation: Consider three sets of nodes A, B and C in a directed acyclic graph. We want to verify the conditional dependency of A and B, given C. D-separation (directional separation) rule [31] can determine this based on the paths that exist between A and B on the graph. Each path connecting A and B is blocked if it involves a node s for which either: a) arrows meet head-to-tail or tail-to-tail at node s and $s \in C$ (Fig. 6a), or b) arrows meet head-to-head at node s and neither the node nor any of its descendants are in the set C (Fig. 6b). If all paths between A and B are blocked, they are conditionally independent, given C (A and B are d-separated by C and $A \perp\!\!\!\perp B|C$).
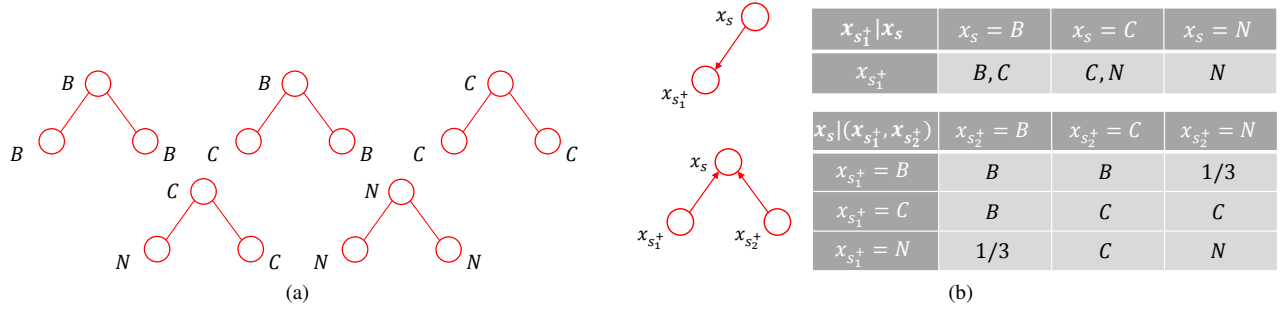
Fig. 4. The prior knowledge used for the three-class problem of cell and nucleus segmentation. Panel (a) shows the plausible label-configurations based on the inclusion of nuclei by cells and cells by the background. Panel (b) shows equivalent probabilistic conditionals when directed trees or polytrees are used for modeling the image. When no child label x_s is plausible for a pair of parent labels $x_{s_1^+}$ and $x_{s_2^+}$ in a polytree clique, a uniform prior $1/3$ is considered.

Using the d-separation rule, the joint posterior in Eq. 2 is expanded as

$$p(x_s, x_{s_1^+}, x_{s_2^+}|\mathbb{Y}) = p(x_s|x_{s_1^+}, x_{s_2^+}, \mathbb{Y})p(x_{s_1^+}, x_{s_2^+}|\mathbb{Y})$$
$$= p(x_s|x_{s_1^+}, x_{s_2^+}, \mathcal{Y}_{a(s)}) \quad (3)$$
$$p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)}, \mathcal{Y}_{d(s_1^+,s_2^+)}),$$

where $\mathcal{Y}_{a(.)}$ and $\mathcal{Y}_{d(.)}$ refer to the sets of observation nodes of the ascendant and descendant nodes, respectively (Fig. 5b). For each node s (or a set of nodes $\mathcal{S}$), ascendant nodes refer to the set of all nodes that are connected to s ($\mathcal{S}$) through edges with inward directions. Similarly, descendant nodes include the nodes connected to node s ($\mathcal{S}$) through outward oriented graph edges. The union of ascendant and descendant observation nodes constructs the set of all observations. See Fig. 5b for a graphical explanation.

We first elaborate on the factor $p(x_s|x_{s_1^+}, x_{s_2^+}, \mathcal{Y}_{a(s)})$ on the right-hand side of Eq. 3. This factor enforces posteriors of
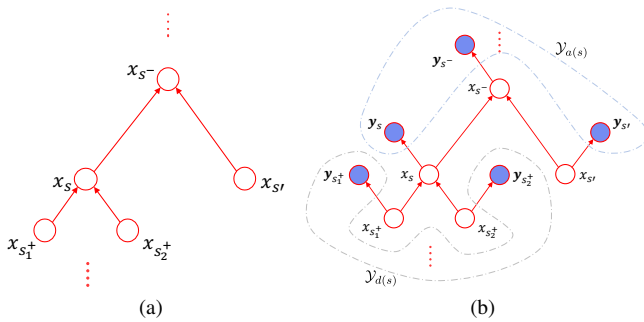

Fig. 5. Distribution of latent and observation nodes in the graph. The notation for nodes connected to an internal node s of the graph is shown in (a). In (b), the graphical representation of ascendant, $\mathcal{Y}_{a(s)}$, and descendant, $\mathcal{Y}_{d(s)}$, observation nodes is depicted.
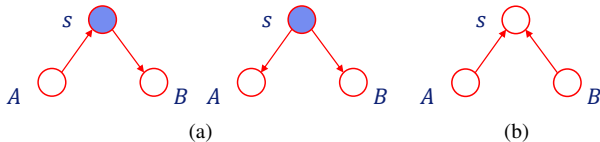

Fig. 6. D-separation rule. Nodes A and B are conditionally independent given C, when graph edges meet head-to-tail or tail-to-tail and $s \in C$ (a), or when graph edges meet head-to-head and $s \notin C$ (b).

unfeasible configurations to zero, as it is a product of the joint probability of a child node and its two parent nodes.

$$p(x_s|x_{s_1^+}, x_{s_2^+}, \mathcal{Y}_{a(s)}) = \frac{p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})}{\sum_{x_s'} p(x_s', x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})} \quad (4)$$

Using the d-separation rule, the numerator becomes:

$$p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)}) = p(x_{s_1^+}, x_{s_2^+}|x_s)p(x_s|\mathcal{Y}_{a(s)})$$
$$= \frac{p(x_s, x_{s_1^+}, x_{s_2^+})}{p(x_s)}p(x_s|\mathcal{Y}_{a(s)}). \quad (5)$$

The factor $p(x_s, x_{s_1^+}, x_{s_2^+})$ in Eq. 5 controls the occurrence of feasible and unfeasible configurations on the graph, by setting nonzero and zero values, respectively. The factor $p(x_s|\mathcal{Y}_{a(s)})$ in Eq. 5 is the posterior of node s given the observations of all its ascendant nodes and its own observations. This *top-down posterior* is expanded as:

$$p(x_s|\mathcal{Y}_{a(s)}) \propto \sum_{x_{s^-}, x_{s'}} p(\mathbf{y}_s|x_s)p(\mathbf{y}_{s'}|x_{s'})p(x_{s'}|\mathcal{Y}_{d(s')})$$
$$\frac{p(x_s, x_{s'}, x_{s^-})}{p(x_{s^-})p(x_{s'})}p(x_{s^-}|\mathcal{Y}_{a(s^-)}). \quad (6)$$

Equation 6 indicates that having calculated the likelihood probabilities $p(\mathbf{y}_s|x_s)$, $p(\mathbf{y}_{s'}|x_{s'})$, and the posterior $p(x_{s'}|\mathcal{Y}_{d(s')})$, the top-down posterior of node s is calculated based on top-down posterior of the node s^-. This implies that a top-down recursion calculates the top-down posteriors for all nodes.

The factor $p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)}, \mathcal{Y}_{d(s_1^+,s_2^+)})$ on the right-hand side of Eq. 3 is factorized by several applications of d-separation rule. This factorization separates parts calculated from ascendant and descendant nodes as follows.

$$p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)}, \mathcal{Y}_{d(s_1^+,s_2^+)})$$
$$\propto p(\mathcal{Y}_{a(s_1^+,s_2^+)}, \mathcal{Y}_{d(s_1^+,s_2^+)}|x_{s_1^+}, x_{s_2^+})p(x_{s_1^+}, x_{s_2^+})$$
$$= p(\mathcal{Y}_{a(s_1^+,s_2^+)}|x_{s_1^+}, x_{s_2^+})p(\mathcal{Y}_{d(s_1^+,s_2^+)}|x_{s_1^+}, x_{s_2^+})p(x_{s_1^+}, x_{s_2^+})$$
$$= p(\mathcal{Y}_{a(s_1^+,s_2^+)}|x_{s_1^+}, x_{s_2^+})p(\mathcal{Y}_{d(s_1^+)}|x_{s_1^+})p(\mathcal{Y}_{d(s_2^+)}|x_{s_2^+})p(x_{s_1^+}, x_{s_2^+})$$
$$\propto p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)})\frac{p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})}{p(x_{s_1^+})}\frac{p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})}{p(x_{s_2^+})} \quad (7)$$

Similar to Eq. 6, $p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)})$ on the right-hand side of Eq. 7 is calculated through a top-down recursion as below.

$$p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+,s_2^+)}) \propto \sum_{x_s} p(\mathbf{y}_{s_1^+}|x_{s_1^+})p(\mathbf{y}_{s_2^+}|x_{s_2^+})$$
$$p(x_{s_1^+}, x_{s_2^+}|x_s)p(x_s|\mathcal{Y}_{a(s)}) \quad (8)$$

The factors $p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})$ and $p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})$ in Eq. 7 are called *bottom-up posteriors* as they are calculated based on posteriors of their descendant nodes. For each node s in the graph, the bottom-up posterior is written as

$$p(x_s|\mathcal{Y}_{d(s)}) \propto \sum_{x_{s_1^+}, x_{s_2^+}} p(\mathbf{y}_{s_1^+}|x_{s_1^+})p(\mathbf{y}_{s_2^+}|x_{s_2^+}) \quad (9)$$
$$p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})p(x_s|x_{s_1^+}, x_{s_2^+}).$$

Derivations of Eq. 6, 8 and 9 are included in Appendix A.

Making use of Eq. 3, 4, 5 and 7, the node's posterior in Eq. 2, given all the observations, is written as follows.

$$p(x_s|\mathbb{Y}) \propto \sum_{x_{s_1^+}, x_{s_2^+}} \frac{p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})}{\sum_{x_s'} p(x_s', x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})} \quad (10)$$
$$p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+, s_2^+)}) \frac{p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})}{p(x_{s_1^+})} \frac{p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})}{p(x_{s_2^+})}$$

Equation 10 calculates the posterior at each node s using three marginal posteriors $p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)})$, $p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+, s_2^+)})$ and $p(x_s|\mathcal{Y}_{d(s)})$, in Eq. 5, 8 and 9. Each term is calculated through either a top-down or a bottom-up recursion. The inference is summarized in Algorithm 1. Note that $\mathcal{R}$ and $\mathcal{L}$ denote the set of root nodes and the leaf node in the graph, respectively.

IV. Experiments and results

A. General experimental design

We evaluated the proposed inference algorithm by classifying synthetic data generated using ancestral sampling. Next, two fluorescent microscopy image datasets were used for evaluating our multi-class segmentation method. The method was compared to SegNet [42], as an instance of a powerful deep convolutional neural network introduced for multi-label image segmentation. We also compared segmentation using both trees and polytrees on the real image datasets to explore how changing the direction of edges and therefore the use of two-wise priors instead of three-wise priors affect the results. For inferring posteriors on trees, we adapted Laferte *et al.* [33] formulation into the graphs generated in this work.

B. Validation of the inference algorithm: ancestral sampling

To assess the performance of the inference algorithm, regardless of the image processing tools employed, we considered the classification of synthetic data generated by ancestral sampling technique [29]. We draw samples for x_s variables to represent ground truth data. Based on this, the $\mathbf{y}_s$ variables are then drawn according to the presumed class conditional distributions. Next, ignoring the reference x_s variables of the first step, new values are inferred for x_s from the observed $\mathbf{y}_s$ variables only. We then compare the inferred x_s variables to the ground truth and experimentally validate the viability of our inference algorithm.

To draw samples $\hat{x}_1, \hat{x}_2, ..., \hat{x}_N$ from the joint distribution $p(\mathbb{X}, \mathbb{Y})$, we first sample from the probability distribution $p(x_s)\big|_{s \in \mathcal{R}}$ for all root nodes. Visiting each internal node in an upward recursion, we sample from the conditional distribution

---

Algorithm 1 Label inference on polytrees

$\square$ **Preliminary pass**. This initial upward recursion computes prior marginals for each node. The parameters $p(x_s|x_{s_1^+}, x_{s_2^+})$ are set based on problem the model represents.
for all $s \in \mathcal{R}$ **do**
  $p(x_s) = \frac{1}{|\Lambda|}$
end for
for all $s \notin \mathcal{R}$ **do**
  $p(x_s) = \sum_{x_{s_1^+}, x_{s_2^+}} p(x_s|x_{s_1^+}, x_{s_2^+})p(x_{s_1^+})p(x_{s_2^+})$
  $p(x_{s_1^+}, x_{s_2^+}|x_s) = \frac{p(x_s|x_{s_1^+}, x_{s_2^+})p(x_{s_1^+})p(x_{s_2^+})}{p(x_s)}$
end for
$\triangle$ **Bottom-up pass**. Upward recursion for calculating bottom-up posteriors of nodes.
for all $s \in \mathcal{R}$ **do**
  $p(x_s|\mathcal{Y}_{d(s)}) = p(x_s)$
end for
for all $s \notin \mathcal{R}$ **do**
  $p(x_s|\mathcal{Y}_{d(s)}) \propto \sum_{x_{s_1^+}, x_{s_2^+}} p(\mathbf{y}_{s_1^+}|x_{s_1^+})p(\mathbf{y}_{s_2^+}|x_{s_2^+})$
  $p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})p(x_s|x_{s_1^+}, x_{s_2^+})$
end for
∇ **Top-down pass**. Downward recursion for calculating top-down posteriors and calculation of complete posteriors from marginal posteriors.
if $s = \mathcal{L}$ **then**
  $p(x_s|\mathcal{Y}_{a(s)}) = p(x_s|\mathbf{y}_s) \propto p(\mathbf{y}_s|x_s)p(x_s)$
end if
for all $s \neq \mathcal{L}$ **do**
  $p(x_s|\mathcal{Y}_{a(s)}) \propto \sum_{x_{s^-}, x_{s'}} p(\mathbf{y}_s|x_s)p(\mathbf{y}_{s'}|x_{s'})p(x_{s'}|\mathcal{Y}_{d(s')})$
  $\frac{p(x_s, x_{s'}|x_{s^-})}{p(x_{s'})}p(x_{s^-}|\mathcal{Y}_{a(s^-)})$
  $p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s)}) = p(x_{s_1^+}, x_{s_2^+}|x_s)p(x_s|\mathcal{Y}_{a(s)})$
  $p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+, s_2^+)})$
  $\propto \sum_{x_s} p(\mathbf{y}_{s_1^+}|x_{s_1^+})p(\mathbf{y}_{s_2^+}|x_{s_2^+})p(x_{s_1^+}, x_{s_2^+}|x_s)p(x_s|\mathcal{Y}_{a(s)})$
end for

$p(x_s|x_{s_1^+}, x_{s_2^+})$, where the parent labels $\hat{x}_{s_1^+}$ and $\hat{x}_{s_2^+}$ have been sampled in previous steps. Once we have sampled from the leaf node of the graph, $\hat{x}_N$, we will have obtained a sample from the joint distribution $p(\mathbb{X}, \mathbb{Y})$.

In this section only, we considered two classes for x_s for simplicity, and selected y_s from the continuous range of $[0, 1]$. Class conditional likelihood functions, $p(y_s|x_s)$ were beta distributions. For different numbers of root nodes ranging from 10 to 100000 (i.e. 19 to 199999 nodes in total as the graph is binary), graphs with random structures were generated and labels were inferred. Figures 7a and 7b show beta distributions for different selectivities. Figure 7c depicts the percentage of the wrongly inferred labels for different graph sizes and the corresponding beta distributions. This experiment shows that even with significant overlaps between the likelihoods of two classes, where $a > 0.8$, the inference error is stable and small (i.e., less than 10%). Therefore, this experiment verifies the correctness of the developed derivations and also indicates that inference accuracy increases with the selectivity of the likelihood functions.
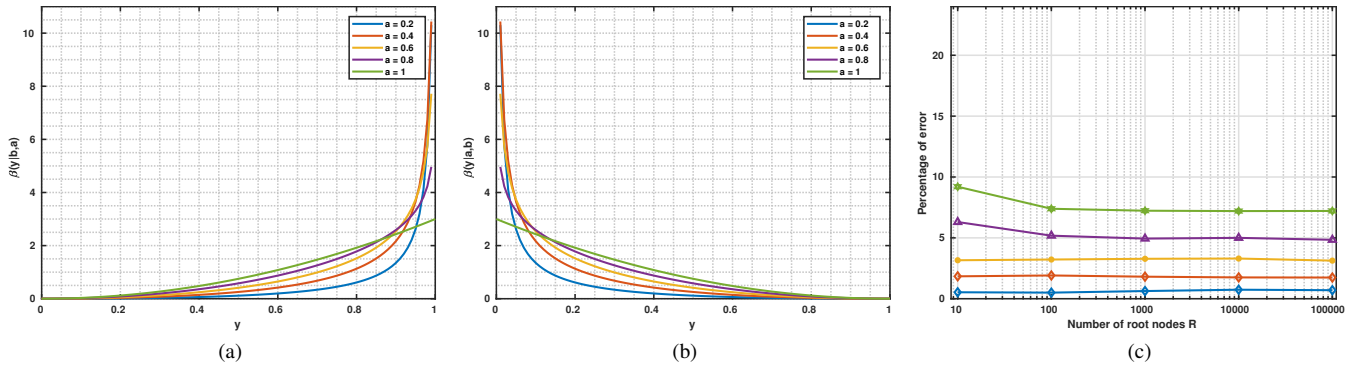
Fig. 7. Panels (a) and (b) show beta distributions used as class conditional likelihood functions in ancestral sampling. Value of b was fixed and curves correspond to values of a from 0.2 to 1, respectively, with an increasing overlap on the likelihoods (thus potential classification errors). In (c), the percentages of wrongly inferred labels using ancestral sampling are shown.

C. Validation on multi-class image segmentation

The proposed algorithm was applied to the problem of supervised multi-class image segmentation, and to evaluate the role of exploiting prior knowledge in segmentation. Two real image datasets were chosen from the publicly available datasets on Broad Bioimage Benchmark Collection that contain two-channel fluorescence microscopy images with cells and nuclei, namely BBBC020 and BBBC007 datasets [45]. In these cases, between-class relationships can help to improve the segmentation results, as only a certain set of label configurations are plausible. To compare the performance of the method to the state-of-the-art, SegNet [42] was employed for the segmentation of images.

BBBC020 contains 20 two-channel *in vitro* microscopy images of murine bone marrow macrophages, and BBBC007 has 16 two-channel *in vitro* microscopy images of drosophila Kc167 cells. Manual annotations are available for both datasets. These two datasets have the same type of images and define similar multi-class segmentation problems of cells and nuclei. The BBBC007 dataset has noisier images and a larger number of overlapping cells, however, they have more regular shapes. See Fig. 8 for samples from the two datasets.

To explore the role of features used for inference, we used two types of features: 1) scale-space first and second order differential invariants [46], 2) deep representations extracted by SegNet. In the following, details of experiments with the two feature sets are explained and results are compared to SegNet. The accuracy of the segmentation was measured by calculating confusion matrices and the Dice similarity coefficients (DSC) [47] computed by comparing the segmentation results with the ground truth.

D. Polytree with scale-space differential invariant features

In this experiment, features were chosen to be intensity value, the absolute value of the gradient, and determinants and traces of the Hessian matrix at 7 scales, for each microscopy channel. A total of 32 features were initially calculated for each image, out of which the most relevant features were selected using Fisher discriminant score [29]. Fisher scores,

W_d, are weights with higher values for features that have higher discrimination abilities and are calculated as follows.

$$W_d = \frac{\sum_{c=1}^{N}(m_d - m_{d,c})^2}{\sum_{c=1}^{N} s_{d,c}^2} \frac{N}{N-1} \quad (11)$$

Where d is the index of the feature, N is the total number of classes, m_d is the mean of dth feature over the training images. $m_{d,c}$ and $s_{d,c}$ denote mean and standard deviation of dth feature within samples of cth class, respectively.

For each dataset, four images were used for feature selection through ranking features based on their Fisher scores. The rest of the images were used for cross validation, i.e. 4- and 6-fold cross validations were applied on the 16 and 12 remaining images in BBBC020 and BBBC007 datasets, respectively. The four images used for Fisher score calculation were always included in the training sets during cross validation.

Once the Fisher scores were calculated, features were ranked for each class separately, and the first D of them were selected for classification. Gaussian distributions were used for class conditional likelihood functions with a layer-dependent variance that allows higher within-class variances for nodes in the higher levels of the graph. Parameters of the method, including β_s (explained in section III-A), number of intensity features used for graph generation (F) and inference (D), and values of mean (μ_c) and covariance matrix (Σ_c) for each class c were optimized through cross validation for each of the two datasets. Figure 9 shows the block diagram of polytree based segmentation using scale-space differential invariant features.

We applied SegNet to the two datasets and compared the results with polytree segmentation using scale-space differential invariants. As the size of the datasets was not sufficiently large for training the network, random elastic deformations of the training images and their annotations were added to the training sets during each cross validation experiment. This way, the size of the training sets for each experiment on the two datasets was increased to 400 images to improve shift and rotation invariance, and robustness to deformations and gray value variations [48], [49]. The trained network was then evaluated on its segmentation of the test set. Figure 10 shows the confusion matrices for polytree and SegNet segmentations of BBBC020 and BBB007 datasets. Overall segmentation
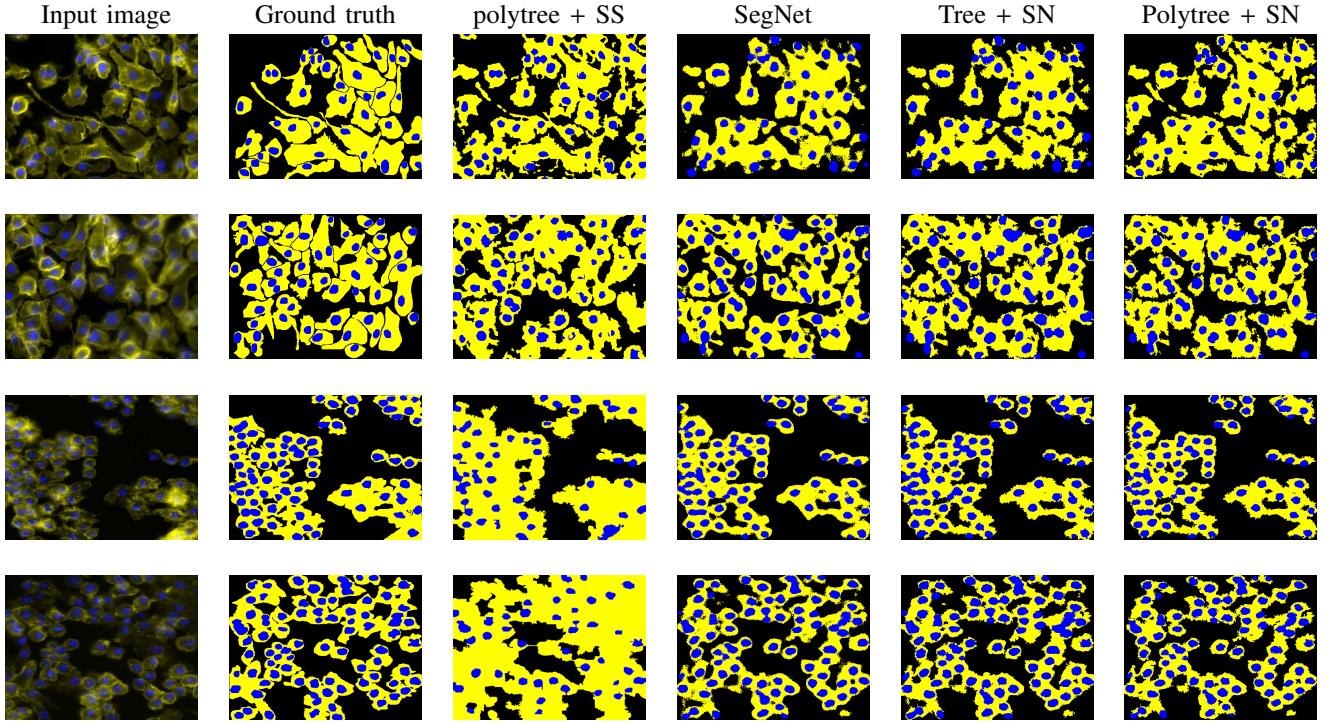
Fig. 8. Sample images from BBBC020 (first and second rows) and BBBC007 (third and fourth rows), their corresponding ground truth and automatic segmentations. Third column shows segmentation results using polytrees with scale-space (SS) features (section IV-D). Fourth column shows results of applying SegNet to the images. Last two columns depict segmentation results using directed trees and polytrees with features generated by SegNet, labeled Tree + SN and Polytree + SN, respectively.
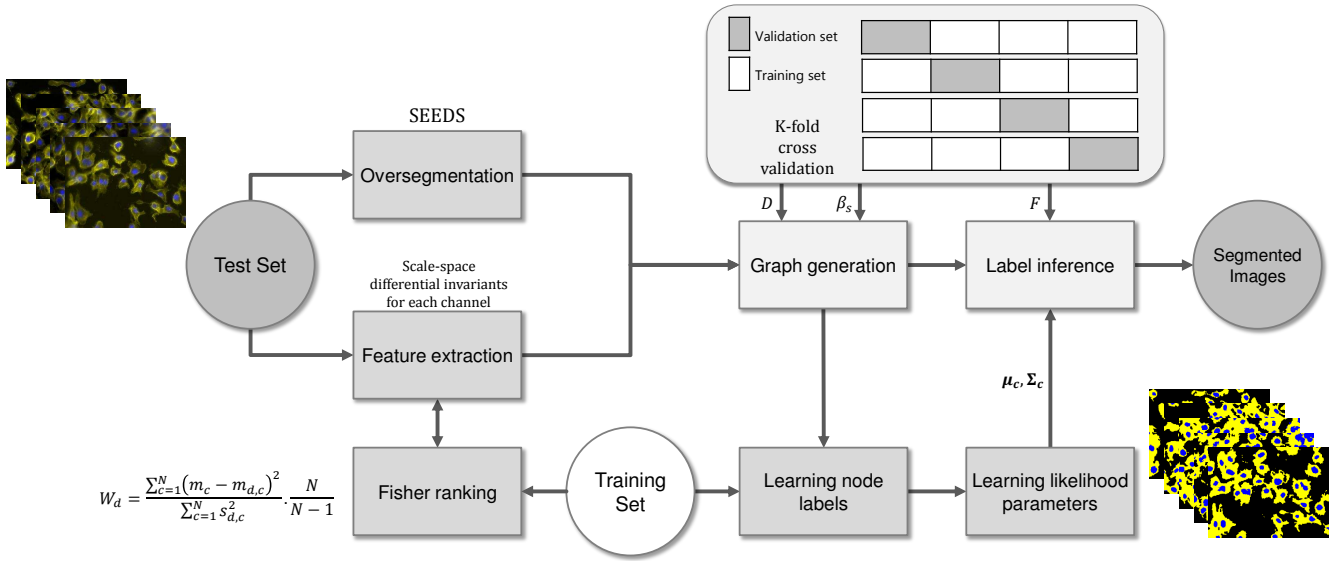

Fig. 9. Block diagram for polytree segmentation with scale-space differential invariant features.

accuracies are similar for the two methods on BBBC020 dataset, while SegNet outperforms polytree on BBBC007. Dice similarity coefficients (DSC) in Fig. 11 indicate SegNet is more accurate than polytree in both classes on the two datasets, except for segmentation of cells in BBBC020, where polytree provides a higher DSC.

Comparing SegNet with polytree using scale-space differential invariant features indicates outperformance of SegNet in segmentation. However, the two methods were compared using different experimental setups. First, SegNet was trained with a larger set of training images (through augmentation). The numbers of features (D) selected after ranking them based on the Fisher scores were 20 and 6, for BBBC020 and BBBC007 datasets, respectively, which are very small compared to the number of features extracted by SegNet. To investigate the two methods regardless of the type of features used, we proposed the use of polytrees with features employed by SegNet in the next section.

Fig. 10. Confusion matrices for SegNet with augmented images and polytree segmentations with scale-space differential invariants on the two real datasets. The overall accuracy of using polytree (a) was slightly higher than SegNet (b) on BBBC020 dataset, while SegNet (d) outperforms polytree (c) on BBBC007 dataset. Number of pixels corresponding to each percentage is shown in bold. Black and white percentages in each box show the proportion of correctly and incorrectly classified pixels, respectively.
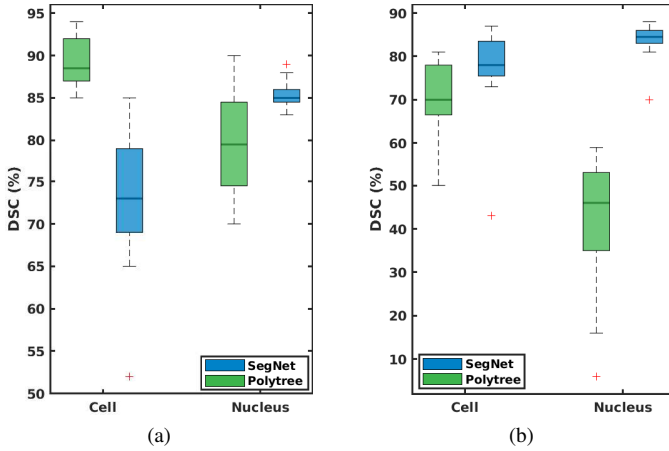

Fig. 11. Dice similarity coefficients (DSC) of polytree based segmentation using scale-space differential invariant features compared to SegNet on (a) BBBC020 and (b) BBBC007 datasets.

E. Polytree with SegNet-based deep features

To compare the developed polytree inference and SegNet using similar preprocessing, feature extraction and selection, and training size, we developed a framework to employ features calculated by SegNet, shown in Fig. 12. In this section, we have also applied directed trees with SegNet features to the segmentation of images in the two datasets. The directed tree was generated by reversing the directions of edges on the irregular polytree and the inference proposed by Laferte *et al.* was adapted to it. Softmax [29] functions were chosen as posteriors.

$$p(x_s = c | \mathbf{y}_s) \propto \frac{\exp(\mathbf{w}_c^T \mathbf{y}_s)}{\sum_k^N \exp(\mathbf{w}_k^T \mathbf{y}_s)} \qquad (12)$$

In Eq. 12, $\mathbf{w}_k$'s are the vectors of weights for each class k, calculated by the CNN to describe the distribution of each class, and N denotes the total number of classes ($N = 3$ in our case).

Note that Eq. 12 implies that a set of improper (unnormalized) class conditional likelihoods, i.e. exponentials, have been used. However, looking at Algorithm 1, the proposed inference algorithm normalizes every term that contains likelihood probability of nodes, facilitating the utilization of unnormalized likelihood functions. For this reason, we chose exponentials as the likelihood functions, i.e. $p(\mathbf{y}_s | x_s = c) \propto exp(\mathbf{w}_c \mathbf{y}_s)$. Both of the class parameters ($\mathbf{w}_c$) and feature vectors ($\mathbf{y}_s$) are provided by the SegNet. Therefore, having trained the SegNet, we do not require any additional training steps for polytree segmentation.

In applying CNN on BBBC020 and BBBC007 datasets, the same image augmentation procedure as explained in section IV-D was employed. Segmentation performance of the methods were compared at three different sizes of datasets; original dataset size (20 images for BBBC020 and 16 images for BBBC007), 200, and 400 augmented images. In each of the experiments, a four-fold cross validation was done. To perform a cross validation, the augmented images were generated based only on the images in the training folds, so that the network was trained independently of the testing set.

For these experiments, the images were first oversegmented using the SEEDS algorithm [43]. The features provided by SegNet were then used for graph generation and, in the next step, for label inference ($F = D$).

Figure 13 shows the DSC of the three methods when SegNet features are used with variable numbers of the training samples. Table II shows the average Dice similarity coefficients for each of the three methods and for each size of the training set for BBBC020 and BBBC007 datasets. The superior results of the directed tree and polytree indicate the effectiveness of the prior knowledge imposed by these directed graphical models, which cannot be captured by SegNet. It can also be seen that the performance of directed trees tend to have larger variances compared to polytrees. This higher uncertainty is likely to stem from the inability of directed trees to eliminate unfeasible label configurations, eliminated by polytrees, that allows semantically wrong segmentations (see section IV-F).

F. Prediction of segmentation error

Unlike discriminative models, generative models incorporate priors in calculating the posterior distributions. Accord-
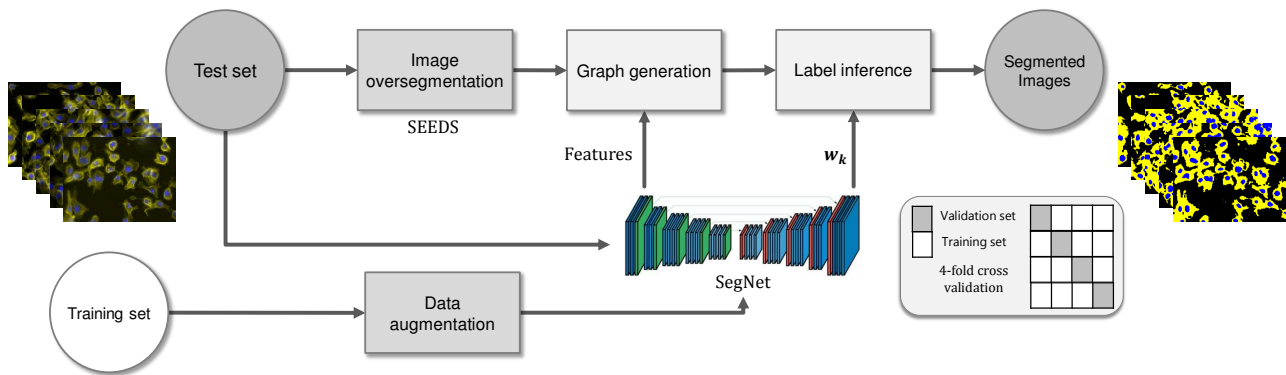
Fig. 12. The proposed architecture for using SegNet-based deep features and learning class conditional likelihood functions.
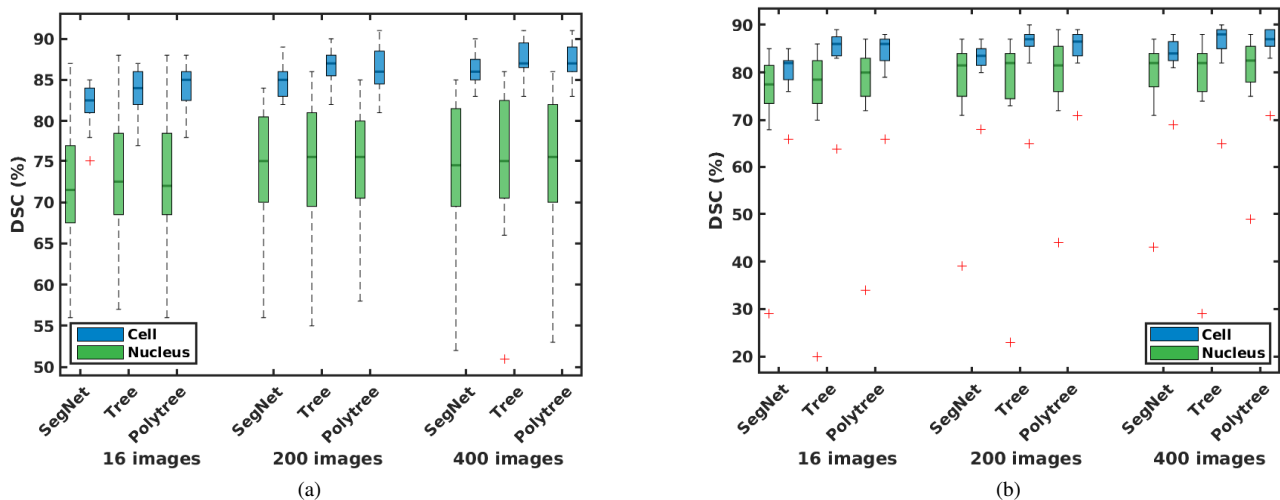

(a)

(b)

Fig. 13. Dice similarity coefficients of the three methods for segmenting cells and nuclei in (a) BBBC020 and (b) BBBC007 datasets, respectively.

TABLE II
MEAN DICE SIMILARITY COEFFICIENTS OF THE POLYTREE, DIRECTED TREE AND SEGNET ON BBBC020 AND BBBC007 DATASETS. P-VALUES RESULTING FROM A PAIRWISE t-TEST FOR COMPARISON OF POLYTREE AND SEGNET WERE LESS THAN 0.05 IN ALL EXPERIMENTS.

Dataset	BBBC020			BBBC007		
# Images	20	200	400	16	200	400
Polytree	**78.60 ± 5.42**	80.43 ± 4.76	81.35 ± 5.18	**80.28 ± 8.44**	**82.09 ± 7.46**	**83.06 ± 6.85**
Directed tree	78.45 ± 5.39	**80.52 ± 4.82**	**81.45 ± 5.21**	79.65 ± 10.62	81.00 ± 10.40	81.75 ± 9.64
SegNet	77.00 ± 5.28	79.42 ± 4.71	80.40 ± 5.06	77.40 ± 8.83	80.06 ± 7.79	81.03 ± 7.43

ingly, the proposed polytree graphical model can evaluate to what extent its estimated clique labels comply with the imposed priors. A strong disagreement can indicate an erroneous segmentation that can be flagged up for manual inspection. To implement this, the labels of cliques are read from the graph representing the segmented image, and their probabilities are calculated using the constraints in Fig. 4b. Areas in the image that correspond to cliques with unfeasible labels (zero probabilities) are then marked as potential segmentation errors. Figure 14 shows samples from BBBC020 and BBBC007 and the error predicted for them. To represent the confidence of the model in labeling the wrongly segmented areas, they are marked by red colors with different values, corresponding to the entropy of the joint posterior of the clique. Areas with

lower and higher error likelihoods (entropies), are shown in lighter and darker colors, respectively.

The error prediction ability of the directed trees was also evaluated. Figure 15 shows Dice similarity coefficients between the potentially incorrectly segmented areas and the actual segmentation error for both methods. Figure 16 shows the average Dice similarity coefficients for different thresholds of entropies for the models on the two datasets. These two figures indicate that polytrees are superior in predicting the segmentation error. This superiority is due to the more effective imposition of prior knowledge in polytrees compared to trees (three-wise constraints versus two-wise constraints, respectively).

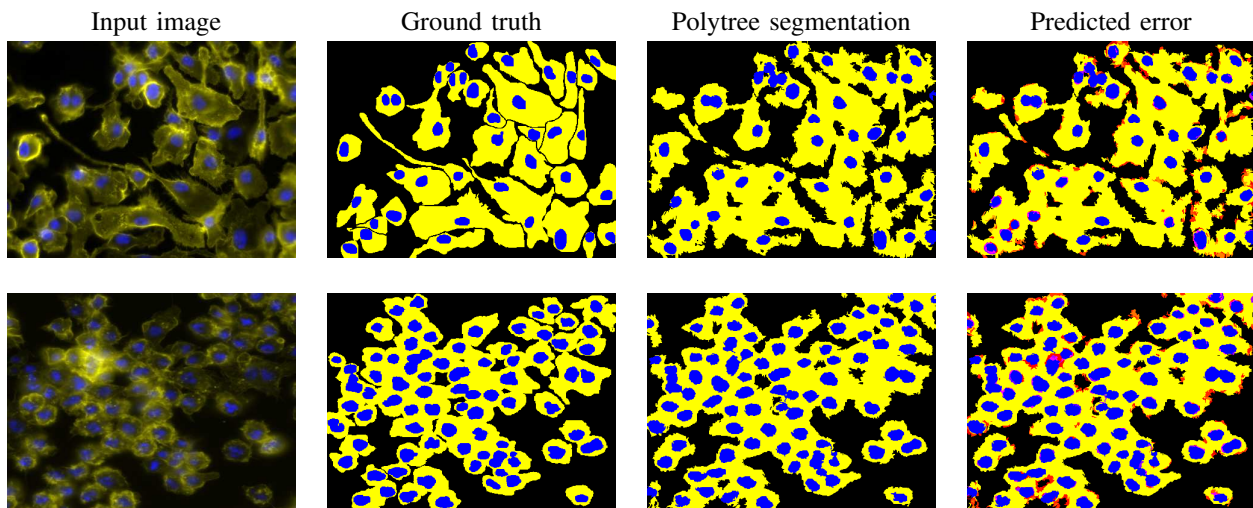Input image   Ground truth   Polytree segmentation   Predicted error

Fig. 14. The ability of the proposed method in nominating the possibly wrongly segmented areas shown for samples from BBBC020 (first row) and BBBC007 (second row) datasets. Value of red color is proportional to the probability of being an error in the segmentation.
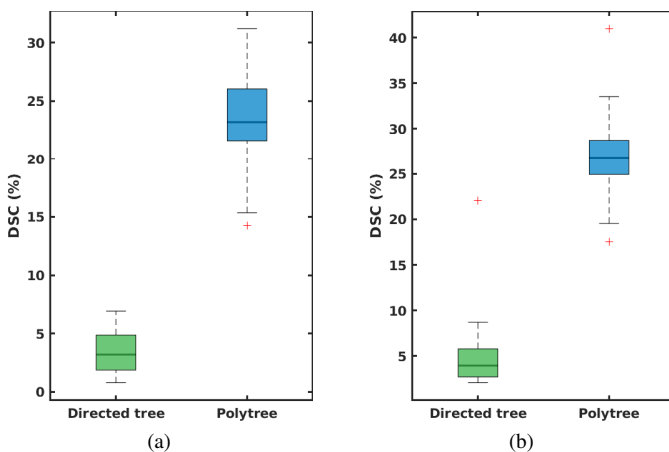

Fig. 15. Dice similarity coefficients between the predicted and the actual segmentation error for directed trees and polytrees on (a) BBBC020 and (b) BBBC007 datasets.
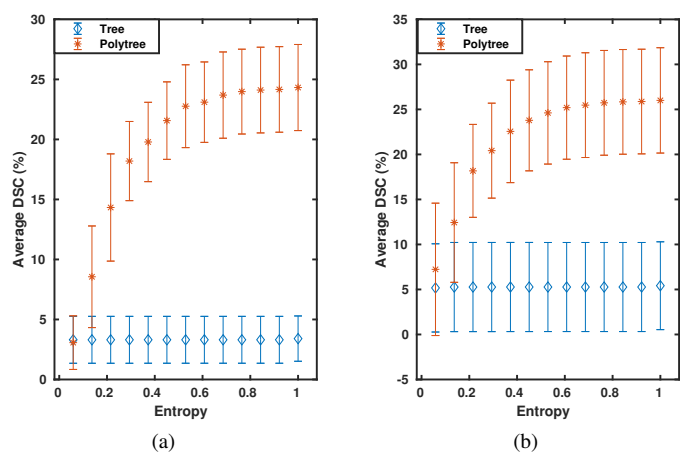
Fig. 16. Average Dice similarity coefficients between the predicted and actual segmentation error for directed trees and polytrees at different thresholds of entropies of cliques on (a) BBBC020 and (b) BBBC007 datasets.

V. DISCUSSION AND CONCLUSIONS

This work proposes a new inference algorithm for multi-class segmentation using irregular directed graphical models. The image is first oversegmented and a graph is generated by recursively merging the two most similar nodes in the graph until a hierarchical graphical model is generated that has no loops. Two types of features were used in this study: 1) scale-space differential invariants of intensity and 2) SegNet-based deep image representations. This was done to investigate the dependency of the method performance on the features used. Two publicly available real microscopy image datasets were used for evaluation. We showed that our polytree based method outperforms the state-of-the-art convolutional neural network, SegNet [42], and equals or surpasses the customized directed trees, in multi-class segmentation. Also, polytrees were shown to predict errors in segmentation more accurately compared to directed trees.

In the literature, directed graphical models have been em-

ployed to incorporate prior knowledge to improve segmentation [23], [24]. However, a large majority of the works rely on directed trees, due to more simple inference and the existence of efficient closed-form solutions for posteriors. This work introduces polytrees for multi-class segmentation and models more complex label dependencies between the child and parent nodes. We also derive, for the first time, closed-form solutions for posteriors on the polytrees. It should be noted that factor graphs [50] can also provide closed-form solutions as long as the original graph structure can be converted to a factor graph without loops. However, the proposed inference method does not require the extra step for generating a second factor graph, simplifying the implementation.

Using polytrees with scale-space differential invariant features (Fig. 11) suggests that depending on the choice of model features and parameters, it can outperform SegNet, even though the latter is trained on a much larger dataset (16 vs. 400 augmented images). Additionally, the distinct performance of the polytree segmentation on BBBC007 dataset, when

different types of features were used, reveals the key role of features in the segmentation performance. By using the same features as the SegNet, polytree provides a better segmentation compared to this convolutional neural network (see Table II). This superiority owes to the model's ability to explicitly enforce prior knowledge and to eliminate unfeasible label configurations, which in turn results in outperforming directed trees in predicting segmentation error (see Fig. 15 and 16). An example of these unfeasible configurations for the problem of segmenting cells and nuclei is the existence of a cell area inside a nucleus. SegNet can also learn such dependencies through its cascade of convolutional layers. However, its efficiency relies on the quality of the training data and the existence of sufficient instances of the dependencies, which might not be possible for every dataset.

In the current implementation of the proposed algorithm, the overall segmentation performance of the method can be confined by the graph generation quality. To address this, one line of future work can be the development of a Maximum Posterior (MAP) estimation [29] for graph generation that optimizes the graph structure jointly with label inference. On the other hand, it is worth mentioning that the small margin of improvement by the proposed graph based segmentation over SegNet is because features learned by the CNN are minimizing the cost function of SegNet rather than the cost function of the polytree. Another line of future work can be extracting features by neural networks that are specifically minimizing the cost of polytree.

The proposed application of the directed graphical models facilitates extracting statistics of relationships between class labels from the graph, in addition to the current use of the graph for imposing prior knowledge. For example, using the proposed method for the segmentation of host and pathogen cells, the proportions of intracellular and extracellular pathogen cells, infected and healthy host cells can be calculated from the labeled graph, both at a specific time point and over time for disease progression monitoring. Such applications introduce new capabilities of graph based segmentation for the behavioral analysis of diseases and biological systems.

Appendix A
Proofs of equations

- Expansion of Eq. 6 (top-down)

$$p(x_s|\mathcal{Y}_{a(s)}) \propto p(x_s, \mathcal{Y}_{a(s)})$$
$$= \sum_{x_{s^-}, x_{s'}} p(\mathcal{Y}_{a(s)}, x_{s^-}, x_s, x_{s'})$$
$$= \sum_{x_{s^-}, x_{s'}} p(\mathcal{Y}_{a(s)}|x_{s^-}, x_s, x_{s'})p(x_{s^-}, x_s, x_{s'})$$
$$= \sum_{x_{s^-}, x_{s'}} p(\mathbf{y}_s|x_s)p(\mathbf{y}_{s'}|x_{s'})p(\mathcal{Y}_{d(s')}|x_{s'}) \quad (13)$$
$$p(\mathcal{Y}_{a(s^-)}|x_{s^-})p(x_{s^-}, x_s, x_{s'})$$
$$\propto \sum_{x_{s^-}, x_{s'}} p(\mathbf{y}_s|x_s)p(\mathbf{y}_{s'}|x_{s'})p(x_{s'}|\mathcal{Y}_{d(s')})$$
$$\frac{p(x_s, x_{s'}, x_{s^-})}{p(x_{s^-})p(x_{s'})}p(x_{s^-}|\mathcal{Y}_{a(s^-)})$$

- Expansion of Eq. 8 (top-down)

$$p(x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{a(s_1^+, s_2^+)}) = p(x_{s_1^+}, x_{s_2^+}|\mathbf{y}_{s_1^+}, \mathbf{y}_{s_2^+}, \mathcal{Y}_{a(s)})$$
$$= \sum_{x_s} p(x_s, x_{s_1^+}, x_{s_2^+}|\mathbf{y}_{s_1^+}, \mathbf{y}_{s_2^+}, \mathcal{Y}_{a(s)})$$
$$\propto \sum_{x_s} p(\mathbf{y}_{s_1^+}, \mathbf{y}_{s_2^+}, \mathcal{Y}_{a(s)}|x_s, x_{s_1^+}, x_{s_2^+})p(x_s, x_{s_1^+}, x_{s_2^+}) \quad (14)$$
$$= \sum_{x_s} p(\mathbf{y}_{s_1^+}|x_{s_1^+})p(\mathbf{y}_{s_2^+}|x_{s_2^+})p(\mathcal{Y}_{a(s)}|x_s)p(x_s, x_{s_1^+}, x_{s_2^+})$$
$$\propto \sum_{x_s} p(\mathbf{y}_{s_1^+}|x_{s_1^+})p(\mathbf{y}_{s_2^+}|x_{s_2^+})p(x_{s_1^+}, x_{s_2^+}|x_s)p(x_s|\mathcal{Y}_{a(s)})$$

- Expansion of Eq. 9 (bottom-up)

$$p(x_s|\mathcal{Y}_{d(s)}) = \sum_{x_{s_1^+}, x_{s_2^+}} p(x_s, x_{s_1^+}, x_{s_2^+}|\mathcal{Y}_{d(s)})$$
$$\propto \sum_{x_{s_1^+}, x_{s_2^+}} p(\mathbf{y}_{s_1^+}, \mathcal{Y}_{d(s_1^+)}, \mathbf{y}_{s_2^+}, \mathcal{Y}_{d(s_2^+)}|x_s, x_{s_1^+}, x_{s_2^+})$$
$$p(x_s, x_{s_1^+}, x_{s_2^+})$$
$$= \sum_{x_{s_1^+}, x_{s_2^+}} p(\mathbf{y}_{s_1^+}|x_{s_1^+})p(\mathcal{Y}_{d(s_1^+)}|x_{s_1^+})p(\mathbf{y}_{s_2^+}|x_{s_2^+}) \quad (15)$$
$$p(\mathcal{Y}_{d(s_2^+)}|x_{s_2^+})p(x_s, x_{s_1^+}, x_{s_2^+})$$
$$\propto \sum_{x_{s_1^+}, x_{s_2^+}} p(\mathbf{y}_{s_1^+}|x_{s_1^+})p(\mathbf{y}_{s_2^+}|x_{s_2^+})$$
$$p(x_{s_1^+}|\mathcal{Y}_{d(s_1^+)})p(x_{s_2^+}|\mathcal{Y}_{d(s_2^+)})p(x_s|x_{s_1^+}, x_{s_2^+})$$

References

[1] A. Mantovani, F. Marchesi, A. Malesci, L. Laghi, and P. Allavena, "Tumour-associated macrophages as treatment targets in oncology," *Nature reviews Clinical oncology*, 2017.

[2] J. C. Kagan and G. M. Barton, "Emerging principles governing signal transduction by pattern-recognition receptors," *Cold Spring Harbor perspectives in biology*, vol. 7, no. 3, p. a016253, 2015.

[3] X. Xiao, X. Cheng, S. Su, Q. Mao, and K.-C. Chou, "pLoc-mGpos: incorporate key gene ontology information into general PseAAC for predicting subcellular localization of Gram-positive bacterial proteins," *Natural Science*, vol. 9, no. 09, p. 330, 2017.

[4] A. Fatica and I. Bozzoni, "Long non-coding RNAs: new players in cell differentiation and development," *Nature Reviews Genetics*, vol. 15, no. 1, pp. 7–21, 2014.

[5] A. Bojarczuk, K. A. Miller, R. Hotham, A. Lewis, N. V. Ogryzko, A. A. Kamuyango, H. Frost, R. H. Gibson, E. Stillman, R. C. May, S. A. Renshaw, and S. A. Johnston, "Cryptococcus neoformans Intracellular Proliferation and Capsule Size Determines Early Macrophage Control of Infection," *Scientific Reports*, vol. 6, p. srep21489, Feb. 2016.

[6] J. F. Gibson and S. A. Johnston, "Immunity to Cryptococcus neoformans and C. gattii during cryptococcosis," *Fungal Genetics and Biology*, vol. 78, pp. 76–86, May 2015.

[7] H. Irshad, A. Veillard, L. Roux, and D. Racoceanu, "Methods for Nuclei Detection, Segmentation, and Classification in Digital Histopathology: A Review-Current Status and Future Potential," *Biomedical Engineering, IEEE Reviews in*, vol. 7, pp. 97–114, 2014.

[8] M. N. Gurcan, L. E. Boucheron, A. Can, A. Madabhushi, N. M. Rajpoot, and B. Yener, "Histopathological Image Analysis: A Review," *IEEE Reviews in Biomedical Engineering*, vol. 2, pp. 147–171, 2009.

[9] W. H. De Vos, L. Van Neste, B. Dieriks, G. H. Joss, and P. Van Oostveldt, "High content image cytometry in the context of subnuclear organization," *Cytometry Part A*, vol. 77A, pp. 64–75, Jan. 2010.

[10] M. B. Resnick, T. Konkin, J. Routhier, E. Sabo, and V. E. Pricolo, "Claudin-1 is a strong prognostic indicator in stage II colonic cancer: a tissue microarray study," *Modern Pathology*, vol. 18, p. 511, Apr. 2005.

[11] T. Misteli and K. Meaburn, "Prostate Cancer Diagnostics and Prognostics Based on Interphase Spatial Genome Positioning," tech. rep., The Geneva Foundation Tacoma United States, The Geneva Foundation Tacoma United States, Mar. 2016.

[12] Y. Yin, X. Zhang, R. Williams, X. Wu, D. D. Anderson, and M. Sonka, "LOGISMOS-Layered Optimal Graph Image Segmentation of Multiple Objects and Surfaces: Cartilage Segmentation in the Knee Joint," *IEEE Transactions on Medical Imaging*, vol. 29, pp. 2023–2037, Dec. 2010.

[13] A. Lucchi, C. Becker, P. M. Neila, and P. Fua, "Exploiting enclosing membranes and contextual cues for mitochondria segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 65–72, Springer, 2014.

[14] S. Farhand, R. B. Montero, X. Vial, D. T. Nguyen, M. Reardon, S. M. Pham, F. M. Andreopoulos, and G. Tsechpenakis, "Probabilistic multi-compartment geometric model: Application to cell segmentation," in *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 174–177, May 2012.

[15] K. Mosaliganti, A. Gelas, A. Gouaillard, R. Noche, N. Obholzer, and S. Megason, "Detection of spatially correlated objects in 3d images using appearance models and coupled active contours," in *Medical Image Computing and Computer-Assisted InterventionMICCAI 2009*, pp. 641–648, Springer, 2009.

[16] S. Li, J. Wakefield, and J. A. Noble, "Automated segmentation and alignment of mitotic nuclei for kymograph visualisation," in *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pp. 622–625, IEEE, 2011.

[17] A. Gooya, K. M. Pohl, M. Bilello, L. Cirillo, G. Biros, E. R. Melhem, and C. Davatzikos, "GLISTR: Glioma Image Segmentation and Registration," *IEEE transactions on medical imaging*, vol. 31, pp. 1941–1954, Oct. 2012.

[18] F. O. Kaster, B. H. Menze, M.-A. Weber, and F. A. Hamprecht, "Comparative validation of graphical models for learning tumor segmentations from noisy manual annotations," in *International MICCAI Workshop on Medical Computer Vision*, pp. 74–85, Springer, 2010.

[19] M. L. Seghier, A. Ramlackhansingh, J. Crinion, A. P. Leff, and C. J. Price, "Lesion identification using unified segmentation-normalisation models and fuzzy clustering," *NeuroImage*, vol. 41, pp. 1253–1266, July 2008.

[20] M. J. Wainwright, M. I. Jordan, and others, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 12, pp. 1–305, 2008.

[21] Z. Kato and J. Zerubia, "Markov Random Fields in Image Segmentation," *Foundations and Trends in Signal Processing*, vol. 5, pp. 1–155, Oct. 2012.

[22] L. Zhang and Q. Ji, "Image Segmentation with a Unified Graphical Model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1406–1425, Aug. 2010.

[23] S.-C. Chen, T. Zhao, G. J. Gordon, and R. F. Murphy, "A novel graphical model approach to segmenting cell images," in *2006 IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology*, pp. 1–8, IEEE, 2006.

[24] F. Rathke, S. Schmidt, and C. Schnrr, "Probabilistic Intra-Retinal Layer Segmentation in 3-D OCT Images Using Global Shape Regularization," *Medical Image Analysis*, vol. 18, no. 5, pp. 781–794, 2014.

[25] K. Kampa, D. Putthividhya, and J. C. Principe, "Irregular Tree-Structured Bayesian Network for image segmentation," in *2011 IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, Sept. 2011.

[26] S. Todorovic and M. C. Nechyba, "Dynamic trees for unsupervised segmentation and matching of image regions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 11, pp. 1762–1777, 2005.

[27] E. Mossel, S. Roch, and A. Sly, "Robust estimation of latent tree graphical models: Inferring hidden states with inexact parameters," *IEEE transactions on information theory*, vol. 59, no. 7, pp. 4357–4373, 2013.

[28] L. Song, H. Liu, A. Parikh, and E. Xing, "Nonparametric Latent Tree Graphical Models: Inference, Estimation, and Structure Learning," *arXiv:1401.3940 [stat]*, Jan. 2014. arXiv: 1401.3940.

[29] C. M. Bishop and others, *Pattern recognition and machine learning*, vol. 4. springer New York, 2006.

[30] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[31] J. Pearl, *Probabilistic reasoning in intelligent systems: Networks of plausible reasoning*. Morgan Kaufmann Publishers, Los Altos, 1988.

[32] F. V. Jensen, *An introduction to Bayesian networks*, vol. 210. UCL press London, 1996.

[33] J.-M. Lafert, P. Prez, and F. Heitz, "Discrete Markov image modeling and inference on the quadtree," *Image Processing, IEEE Transactions on*, vol. 9, no. 3, pp. 390–404, 2000.

[34] X. Feng, C. Williams, and S. Felderhof, "Combining belief networks and neural networks for scene segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 467–483, Apr. 2002.

[35] C. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Transactions on Image Processing*, vol. 3, pp. 162–177, Mar. 1994.

[36] W. W. Irving, P. W. Fieguth, and A. S. Willsky, "An overlapping tree approach to multiscale stochastic modeling and estimation," *IEEE Transactions on Image Processing*, vol. 6, pp. 1517–1529, Nov. 1997.

[37] J. Li, R. M. Gray, and R. A. Olshen, "Multiresolution image classification by hierarchical modeling with two-dimensional hidden Markov models," *IEEE Transactions on Information Theory*, vol. 46, pp. 1826–1841, Aug. 2000.

[38] A. J. Slorkey and C. K. L. Williams, "Image modeling with position-encoding dynamic trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 859–871, July 2003.

[39] P. Zhang, G. Wu, Y. Gao, P.-T. Yap, and D. Shen, "A dynamic tree-based registration could handle possible large deformations among MR brain images," *Computerized Medical Imaging and Graphics*, vol. 52, pp. 1–7, Sept. 2016.

[40] A. Delong and Y. Boykov, "Globally optimal segmentation of multi-region objects," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 285–292, Sept. 2009.

[41] H. Isack, O. Veksler, I. Oguz, M. Sonka, and Y. Boykov, "Efficient optimization for Hierarchically-structured Interacting Segments (HINTS)," *arXiv:1703.10530 [cs]*, Mar. 2017. arXiv: 1703.10530.

[42] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[43] M. V. d. Bergh, X. Boix, G. Roig, B. d. Capitani, and L. V. Gool, "SEEDS: Superpixels Extracted via Energy-Driven Sampling," in *Computer Vision ECCV 2012* (A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, eds.), no. 7578 in Lecture Notes in Computer Science, pp. 13–26, Springer Berlin Heidelberg, 2012.

[44] J. Funke, C. Zhang, T. Pietzsch, and S. Saalfeld, "The Candidate Multi-Cut for Cell Segmentation," *ArXiv e-prints*, vol. 1707, p. arXiv:1707.00907, July 2017.

[45] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter, "Annotated high-throughput microscopy image sets for validation," *Nat Methods*, vol. 9, no. 7, p. 637, 2012.

[46] T. Lindeberg, "Scale-space theory: A basic tool for analyzing structures at different scales," *Journal of applied statistics*, vol. 21, no. 1-2, pp. 225–270, 1994.

[47] L. R. Dice, "Measures of the Amount of Ecologic Association Between Species," *Ecology*, vol. 26, pp. 297–302, July 1945.

[48] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis.," in *ICDAR*, vol. 3, pp. 958–962, 2003.

[49] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2015.

[50] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, pp. 498–519, Feb. 2001.