Learning to Segment: Training Hierarchical Segmentation under a Topological Loss

Jan Funke[1], Fred A. Hamprecht[2], and Chong Zhang[2,3]

[1] Institute of Neuroinformatics UZH/ETHZ, Switzerland
[2] CellNetworks & IWR/HCI, Heidelberg University, Germany
[3] Universitat Pompeu Fabra, Spain

Abstract. We propose a generic and efficient learning framework that is applicable to segment images in which individual objects are mainly discernible by boundary cues. Our approach starts by first hierarchically clustering the image and then explaining the image in terms of a cost-minimal subset of non-overlapping segments. The cost of a segmentation is defined as a weighted sum of features of the selected candidates. This formulation allows us to take into account an extensible set of arbitrary features. The maximally discriminative linear combination of features is learned from training data using a margin-rescaled structured SVM. At the core of our formulation is a novel and simple topology-based structured loss which is a combination of counts and geodesic distance of topological errors (splits, merges, false positives and false negatives) relative to the training set. We demonstrate the generality and accuracy of our approach on three challenging 2D cell segmentation problems, where we improve accuracy compared to the current state of the art.

1 Introduction

Accurately segmenting a large number of objects of similar type in crowded images is challenging, e.g. when cells of interest touch or overlap. Therefore, the development of robust and efficient algorithms, generic enough to be applicable for different scenarios, is of great importance. Since local measurements are usually ambiguous due to noise and imaging artefacts, priors about the objects to be segmented are needed. Two of the main challenges for automatic segmentation are how to reflect those priors in the cost function of the segmentation problem, and how to learn them from training data. Usually, the cost functions are designed "by hand" for a particular segmentation problem, or have user-adjustable tuning parameters like a foreground bias to adapt to different setups [15,14].

In this paper, we propose a generic framework for structured learning of the cost function for cell segmentation. The cost function is defined on *candidate segments*, of which we find a cost-minimal, non-overlapping subset to obtain a final segmentation. The main contributions of our approach are: 1) The novel counting-and-propagating topological loss is simple and generic. 2) Our formulation supports a large set of expressive features on the segment candidates. 3) Optimal feature weights are learned from annotated samples to minimize a topological loss on the final segmentation. The capacity to combine and weigh automatically all features from a large set can reduce the effort previously required to manually select suitable features and tune parameters. 4) By considering candidate segments obtained by iteratively merging superpixels [3,4,7], our method is very efficient while still improving on the current state of the art.

Considering candidates from hierarchy of regions has demonstrated to be an effective method for cell detection and segmentation [3,5,8,7,9]. It has been shown that a globally cost-minimal and non-overlapping set of candidates can efficiently be found, either by ILP or dynamic programming [3,7]. However, the proposed methods differ greatly in the way the candidate costs are obtained. For the cell *detection* proposed by Arteta *et al.* [3], the costs are learned using a structured SVM, taking into account the non-overlap constraint of the inference problem. In the context of cell *segmentation* from candidates obtained from iterative merging [5,8,7,9], the costs are learned by providing samples of positive and negative candidates to a binary classifier.

In the following section, we show how the structured learning framework can be used to learn the candidate costs for segmentation under consideration of the problem structure. For that, we introduce a novel loss function which minimizes the propagated topological errors on the resulting segmentation. In Section 3, we demonstrate the effectiveness of our approach on three datasets.

2 Method

Our method is based on selecting candidates extracted from a hierarchy of segmentations, an approach that has recently attracted attention in a range of computer vision problems [2,3,5,8,7]. The underlying idea is to extract a *merge tree* of candidates (i.e., parent nodes are merged segments of their children, see Fig. 2 for an example) that span the whole range from over- to undersegmentation. A segmentation of the image can now be expressed by a selection of non-overlapping candidates. It remains to rate different segmentations by assigning costs to each candidate, which represent the likelihood of being a correct segment. We propose to train a structured SVM to learn those costs as a linear combination of candidate features. For that, we propose a novel loss function minimizing topological errors during learning.

Candidates Extraction. We perform a watershed transformation on a boundary prediction image, for which we trained a random forest classifier using the open source software ilastik [12], to obtain initial segment candidates. Let $G = (C, E, s)$ be the candidate adjacency graph of the initial segmentation, where $C = \{c_1, \ldots, c_N\}$ are the initial candidates and $E \subset C \times C$ is a set of edges representing adjacency, and let $s : E \mapsto \mathbb{R}$ be an edge scoring function. Iteratively, the candidates connected by the edge $\operatorname{argmin}_E s(E)$ with the lowest scores are merged, and the adjacency graph updated accordingly. For the experiments presented in this paper, we use $s(c_i, c_j) = m(c_i, c_j) \min(|c_i|, |c_j|)$, with $m(c_i, c_j)$ being the median boundary prediction value of all boundary pixels between c_i and c_j, and $|c_i|$ the size of a candidate. Note that this way we obtain a binary merge tree structure, but the candidates extraction is a replaceable part of our pipeline. More sophisticated methods for a general tree like ultrametric contour maps [2], graph-based active learning of agglomeration [9], or hierarchical planar correlation clustering [14] can be used as well.

Inference. Let $T = (C, S, f)$ be a *merge tree* of candidates, where C is the set of all candidates, $S \subset C \times C$ are directed edges indicating superset relations (for an example, see Fig. 2), and let $f : C \mapsto \mathbb{R}$ be a candidate cost function. We introduce binary indicator variables $\mathbf{y} = (y_i, \dots, y_N)$, where $y_i = 1$ represents the selection of candidate c_i for the final segmentation. To ensure that only non-overlapping candidates are selected, we restrict the possible configurations of $\mathbf{y}$ to select at most one candidate per path in the merge tree and denote the restricted set of configurations by $\mathcal{Y}$. Let $\mathcal{P}$ be the set of all paths in T. We find the cost-optimal selection of candidates with the following LP:

$$\min_{\mathbf{y} \in \mathbb{R}^N} \sum_i f(c_i) y_i, \quad \text{s.t.} \sum_{i \in P} y_i \leq 1 \quad \forall P \in \mathcal{P}; \quad y_i \geq 0 \quad \forall i. \tag{1}$$

Note that there is no need to explicitly express the integrality constraints, here, since the constraint matrix is totally unimodular and the optimization problem is known to have a polynomial-time solution [3].

Structured Learning of Candidate Costs. We propose to learn the candidate cost function f using a structured SVM [13] from labeled training data. For that, we model the costs $f(c_i)$ as a linear combination of features $\boldsymbol{\phi}_i$ extracted for each candidate individually, i.e., $f(c_i) = \langle \boldsymbol{w}, \boldsymbol{\phi}_i \rangle$. Note that here the linearity can be relaxed, as features can also be a function of arbitrary combinations of groups of features. The weights $\boldsymbol{w}$ can be learned to minimize a topological loss on the training data. Let

$$E(\mathbf{y}) = \sum_{i=1}^{N} f(c_i) y_i = \sum_{i=1}^{N} \langle \boldsymbol{w}, \boldsymbol{\phi}_i \rangle \, y_i = \langle \boldsymbol{w}, \boldsymbol{\Phi} \mathbf{y} \rangle \tag{2}$$

be the cost of a segmentation represented by binary indicator variables $\mathbf{y}$, where $\boldsymbol{\Phi}$ is a combined feature matrix for all candidates. In order to learn the weights $\boldsymbol{w}$, the structured SVM framework needs a training sample $(\mathbf{y}', \boldsymbol{\phi})$ with $\mathbf{y}' \in \mathcal{Y}$. Note that multiple training samples can easily be concatenated into a single one. Since the extracted segment candidates may not perfectly correspond to the (human annotated) ground truth (GT), the training sample $\mathbf{y}'$ is obtained by maximizing the spatial overlap to the GT. Therefore, $\mathbf{y}'$ represents the *best-effort* segmentation compared to the GT. Note that since the proposed learning method uses only the best-effort solution for training, small imprecisions in the human generated ground truth are tolerable.

Given a training sample $(\mathbf{y}', \boldsymbol{\Phi})$, we find an optimal set of weights $\boldsymbol{w}^*$ by minimization of the structured SVM objective

$$\boldsymbol{w}^* = \operatorname*{argmin}_{\boldsymbol{w}} \frac{\lambda}{2} |\boldsymbol{w}|^2 + \max_{\mathbf{y} \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\Phi} \mathbf{y}' - \boldsymbol{\Phi} \mathbf{y} \rangle + \Delta(\mathbf{y}', \mathbf{y}), \tag{3}$$

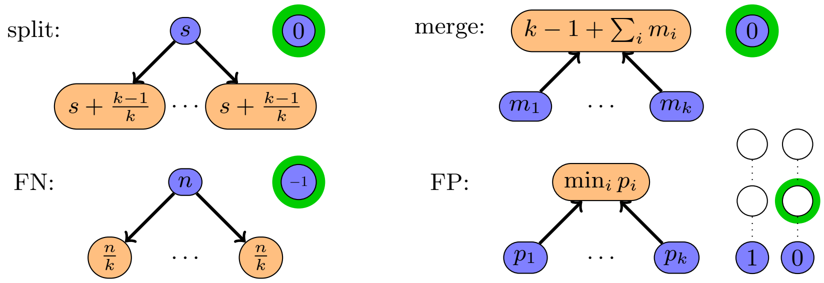
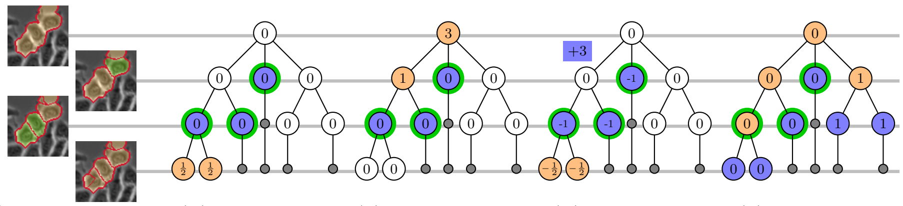


Fig. 1: Illustration of the topological loss used for structured learning: splits, merges, false negatives (FN), and false positives (FP). Starting from nodes with known values (blue): Given the values of parent candidates, split and FN values are propagated downwards; given the values of child candidates, merge and FP values are propagated upwards.

where λ is the weight of the quadratic regularizer and $\Delta(\mathbf{y}', \mathbf{y})$ is an application specific loss function to be minimized. This loss function can be seen as a distance between the desired segmentation $\mathbf{y}'$ and a potential segmentation $\mathbf{y}$. A proper choice of this function is crucial for the success of learning. However, the expressiveness of the loss function is limited by the tractability of the maximization in Eq. 3, which has to be carried out repeatedly during learning.

Here, we propose a topological loss function that directly counts and propagates the number of split, merge, false positive (FP), and false negative (FN) errors of a potential segmentation compared to the best-effort. Due to the tree structure of the candidate subset relations, this loss decomposes into a sum of individual contributions: For each candidate, we determine four values (s, m, p, n), which represent the candidate's contribution to the overall number of split, merge, FN, and FP errors. The distribution of the values is based on the following intuition: Whenever a best-effort candidate is selected, no topological error was made by this selection. If, however, all k children of a best-effort candidate are selected, $k-1$ split errors have been made. In a similar fashion, the selection of a parent of k best-effort candidates causes $k-1$ merge errors. This observation suggests a simple propagation scheme of topological errors: Initial split values of all best-effort candidates and their ancestors are $s = 0$; merge values of all best-effort candidates and their descendants are $m = 0$. In addition to that, the FN value of each best-effort candidate is set to $n = -1$, i.e., as a reward, such that an equivalent amount is payed by not selecting (and therefore missing) the candidate. Initial FP values are set for all leaf nodes in the tree as $p = 0$ if the leaf is a descendant of a best-effort candidate (in this case, selecting the candidate does not cause an FP error), or as $p = 1$ if it is not a descendant of a best-effort candidate. These initial values are then propagated up- and downwards the candidate tree using the rules presented in detail in Fig. 1. For a simple example, see Fig. 2. The sum of the values of a candidate gives the topological error that is caused by selecting this candidate.



(a) segmentations (b) split values (c) merge values (d) FN values (e) FP values

Fig. 2: Example split, merge, false negative (FN), and false positive (FP) values of candidates in a merge tree. The best-effort solution is highlighted in green.

To combine the topological error values (s, m, p, n) of each candidate into the final topological loss, we suggest a linear combination, i.e.,

$$\Delta(\mathbf{y}', \mathbf{y}) = \sum_{i=1}^{N} (\alpha s_i + \beta m_i + \gamma p_i + \delta n_i)\, y_i + \delta c, \tag{4}$$

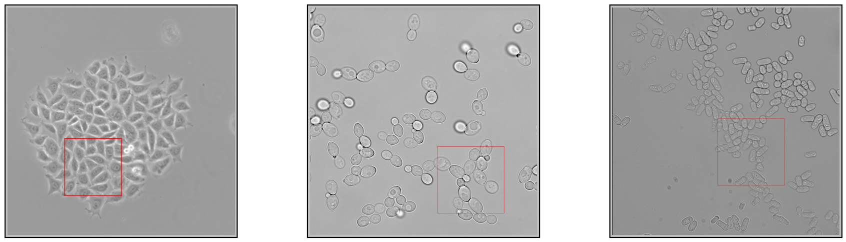
where c is the number of best-effort candidates ensuring that the loss is positive. The parameters $(\alpha, \beta, \gamma, \delta)$ can be used to assign different weights to different error types, which can be useful in situations where split errors are preferred over merge errors, since they are usually faster to repair.

The resulting topological loss function is linear in $\mathbf{y}$, which allows us to solve the maximization in Eq. 3 efficiently using the LP formulation in Eq. 1 for any given $\boldsymbol{w}$. It remains to find the minimum of Eq. 3, which is a piece-wise quadratic and convex function of $\boldsymbol{w}$. For that, we use a bundle method [1].

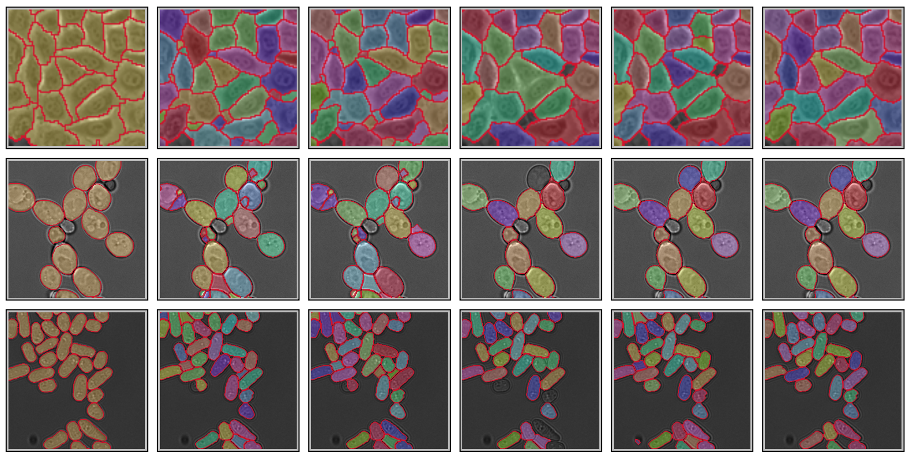
3 Experiments and Results

Datasets. We validate our method on three datasets (see examples in Fig. 3 (a-c)) which present high variabilities in image resolution, microscopy modality, and cell type (shape, appearance and size): 1) phase contrast images of cervical cancer Hela cells from [3] (cells for training/testing: 656/384); 2) bright field images of in-focus Diploid yeast cells from [15] (cells for training/testing: 109/595); 3) bright field images of Fission yeast cells from [10] (cells for training/testing: 1422/240). Challenging problems in general include densely packed and touching cells, intra shape and size variations within the same dataset, weak boundary cues, out-of-focus artifacts, and similar boundaries from other structures.

Experimental setup. The used intensity- and morphology-based candidate features are: size, intensity (sum, mean, variance, skewness, kurtosis, histogram (20 bins), 7 histogram quantiles), circularity, eccentricity, contour angle histogram (16 bins). The intensity features are extracted from the raw and boundary prediction images, on the whole candidate and on the candidate contours. For each pair of features ϕ_i and ϕ_j, the product $\phi_i * \phi_j$ is also added. All features are normalized to be in the range [0,1]. The weights $(\alpha, \beta, \gamma, \delta)$ for combining the four



(a) Hela cancer cells (b) Diploid yeast cells (c) Fission yeast cells



ground truth (d) pCC (e) npCC (f) Hamming (g) cover (h) topological

Fig. 3: Example segmentation results on three datasets (a–c). Results (*inset views*) using correlation clustering (d–e) and using our structured learning formulation with different loss functions $\Delta(\mathbf{y}', \mathbf{y})$ (f–h).

topological errors are chosen through a Leave-one-out cross-validation, specifically, being (1,2,1,2), (1,2,1,5), and (1,2,1,10), for the three datasets respectively.
Comparison. We compare to the two correlation clustering variants (pCC and npCC) on the investigated datasets [15]. For our structured learning formulation, we also include results from using, as loss $\Delta(\mathbf{y}', \mathbf{y})$, a simple Hamming distance and a cover loss. The latter counts the errors from GT region centroids that are either not covered or wrongly covered by the picked candidates, which is similar to the loss in [3]. As opposed to pCC and npCC, the flexible features enrich learning with boundary and region cues, including e.g. texture and shape priors. This can be seen in Fig. 3, where background regions are correctly identified, owing to concavity or texture. Instead of assigning the same loss to each wrong candidate as in Hamming and cover losses, our topological loss explicitly counts errors proportional to the distance in the merge tree. This is more indicative and robust, which is reflected on less FN and splits in the figure.
Evaluation. We report five quantitative measures: variation of information (VOI) and tolerant edit distance (TED) [6] indicate both detection and segmen-

	VOI			TED [6]					DS			dice	overlap
	split	merge	total	FP	FN	split	merge	total	prec.	rec.	F-sc.		
planarCC	0.33	0.52	0.84	20	43	77	30	170	0.86	0.91	0.89	0.79	69.35
nonplanarCC	0.39	0.43	0.82	31	25	53	41	150	0.89	0.93	0.91	0.80	69.21
Hamming	0.33	0.44	**0.77**	2	41	62	6	111	0.95	0.99	**0.97**	0.83	72.40
cover	0.37	0.43	0.80	9	36	109	3	157	0.82	0.99	0.90	0.81	70.12
topological	0.35	0.42	**0.77**	5	33	49	11	**98**	0.95	0.98	**0.97**	**0.84**	**73.20**

Dataset 1

	VOI			TED [6]					DS			dice	overlap
	split	merge	total	FP	FN	split	merge	total	prec.	rec.	F-sc.		
planarCC	0.72	0.24	0.97	237	36	11	74	358	0.70	0.86	0.77	0.87	79.99
nonplanarCC	1.61	0.21	1.81	402	22	5	59	488	0.59	0.92	0.72	0.86	78.07
Hamming	0.44	0.53	0.97	28	159	3	16	206	0.96	0.72	0.82	0.89	80.91
cover	1.04	0.30	1.34	121	50	2	13	186	0.83	0.91	0.87	0.87	79.51
topological	0.50	0.33	**0.83**	40	69	3	4	**116**	0.93	0.89	**0.91**	**0.90**	**82.22**

Dataset 2

	VOI			TED [6]					DS			dice	overlap
	split	merge	total	FP	FN	split	merge	total	prec.	rec.	F-sc.		
planarCC	0.19	0.07	**0.26**	19	2	8	7	**36**	0.92	0.97	**0.94**	**0.91**	**84.67**
nonplanarCC	0.30	0.07	0.37	56	2	9	6	73	0.82	0.97	0.89	0.91	84.34
Hamming	0.84	0.16	1.00	67	37	1	12	117	0.81	0.86	0.83	0.89	82.05
cover	0.42	0.10	0.52	97	6	12	1	116	0.74	0.99	0.85	0.90	81.84
topological	0.19	0.14	0.33	6	25	1	8	40	0.99	0.90	**0.94**	0.90	82.85

Dataset 3

Table 1: Detection and segmentation results on three datasets. The best value in each column is highlighted.

tation errors; Detection score (DS) measures detection accuracy only; Dice coefficient and area overlap provide segmentation accuracy. For DS, we establish possible matches between found regions and GT regions based on overlap, and find a Hungarian matching using the centroid distance as minimizer. Unmatched GT regions are FN, unmatched segmentation regions are FP. We report precision, recall, and F-score. Area overlap is computed between the true positive (TP) detection regions R_{tpd} and the GT region R_{gt}: $(R_{tpd} \cap R_{gt})/(R_{tpd} \cup R_{gt}) \times 100$. TED measures the minimal number of topological errors under tolerable modifications of the segmentation (here, we tolerated boundary shifts up to 10 pixels for Hela cells, and 25 pixels for two datasets of yeast cells). Results are summarized in Table 1. Our proposed method robustly outperforms for the two more challenging datasets (1 and 2), and achieves amongst the best for dataset 3.

4 Discussion and Conclusions

We have proposed a generic framework for structured learning of the cost function for segmenting cells primarily only with boundary cues (Code available at: `http://github.com/funkey/tobasl`). This reduces the effort previously required for manual feature selection or parameter tuning. The tree-topology based loss from the hierarchical segment regions has demonstrated to be efficient and can improve segmentation. The potential restriction on candidate regions imposed by using only trees can partly be lifted by, e.g., constructing multiple trees, at least in test stage, in the spirit of perturb-and-MAP or as in [11] for user correction. We have validated our approach in the context of biological applications, however we expect that it can be applied in other fields. Also, data which has primarily region rather than boundary cues can be turned into the latter, and hence fit into this mould, by using, e.g., pixel-level classifier to delineate region pixels that are on the border. These form part of our future work.

References

1. sbmrm - Bundle Method for Structured Risk Minimization. `https://github.com/funkey/sbmrm`.
2. P. Arbeláez, M.l Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *IEEE PAMI*, 33(5):898–916, 2011.
3. C. Arteta, V. Lempitsky, J.A. Noble, and A. Zisserman. Learning to Detect Cells Using Non-overlapping Extremal Regions. In N. Ayache et al., editor, *MICCAI*, volume 7510, pages 348–356. Springer, Heidelberg, 2012.
4. C. Arteta, V. Lempitsky, J.A. Noble, and A. Zisserman. Learning to Detect Partially Overlapping Instances. In *CVPR*, 2013.
5. J. Funke, B. Andres, F.A. Hamprecht, A. Cardona, and M. Cook. Efficient Automatic 3D-Reconstruction of Branching Neurons from EM Data. In *CVPR*, 2012.
6. Jan Funke, Jonas Klein, Albert Cardona, and Matthew Cook. A Tolerant Edit Distance for Evaluation and Training of Electron Microscopy Reconstruction Algorithms. *CoRR*, 2015.
7. F. Liu, F. Xing, and L. Yang. Robust Muscle Cell Segmentation using Region Selection with Dynamic Programming. In *ISBI*, pages 1381–1384, 2014.
8. T. Liu, E. Jurrus, M. Seyedhossein, M. Ellisman, and T. Tasdizen. Watershed Merge Tree Classification for Electron Microscopy Image Segmentation. In *ICPR*, 2012.
9. J. Nunez-Iglesias, R. Kennedy, S.M. Plaza, A. Chakraborty, and W.T. Katz. Graph-based Active Learning of Agglomeration (GALA): a Python Library to Segment 2D and 3D Neuroimages. *Front. Neuroinform.*, 8:34, 2014.
10. J.Y. Peng, Y.J. Chen, M.D. Green, S.A. Sabatinos, S.L. Forsburg, and C.N. Hsu. PombeX: Robust Cell Segmentation for Fission Yeast Transillumination Images. *PLoS One*, 8(12):e81434, 2013.
11. M. Schiegg, B. Heuer, C. Haubold, S. Wolf, U. Koethe, and F.A. Hamprecht. Proofreading Guidance in Cell Tracking by Sampling from Tracking-by-Assignment Models. In *ISBI*, 2015.
12. C. Sommer, C. Straehle, U. Koethe, and F. A. Hamprecht. ilastik: Interactive Learning and Segmentation Toolkit. In *ISBI*, 2011.
13. I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer. Large Margin Methods for Structured and Interdependent Output Variables. *J Mach. Learn. Res.*, 6:1453–1484, 2005.
14. J. Yarkony, C. Zhang, and C.C. Fowlkes. Hierarhcical Planar Correlation Clustering for Cell Segmentation. In *EMMCVPR*, 2015.
15. C. Zhang, J. Yarkony, and F.A. Hamprecht. Cell Detection and Segmentation using Correlation Clustering. In P. Golland et al., editor, *MICCAI*, volume 8673, pages 9–16. Springer, 2014.