

# Image segmentation & BPDFL in microscopy

Jiri Borovec  
@ CMP, FEE, CTU in Prague

15.03.2018

<http://cmp.felk.cvut.cz/~borovji3/>

<https://www.researchgate.net/publication/323120618>

# Resources

- Publications:

- Borovec J. (2017). **Automatic analysis of gene expressions in Drosophila microscopy images.** <ftp://cmp.felk.cvut.cz/pub/cmp/articles/borovec/Thesis-TR-2017-07.pdf>
- Borovec J., Svihlik J., Kybic J., Habart D. (2017). **Supervised and unsupervised segmentation using superpixels, model estimation, and Graph Cut.** SPIE Journal of Electronic Imaging 26(6), 061610, <http://doi.org/10.1117/1.JEI.26.6.061610>
- Borovec J., Kybic J., Nava R. (2017) **Detection and Localization of Drosophila Egg Chambers in Microscopy Images.** In: Wang Q., Shi Y., Suk H.I., Suzuki K. (eds) Machine Learning in Medical Imaging. MLMI 2017. LNCS, vol 10541. Springer, Cham. [http://doi.org/10.1007/978-3-319-67389-9\\_3](http://doi.org/10.1007/978-3-319-67389-9_3)
- Borovec J., Kybic J., Sugimoto, A. (2017). **Region growing using superpixels with learned shape prior.** SPIE Journal of Electronic Imaging 26(6), 061611, <http://doi.org/10.1117/1.JEI.26.6.061611>
- Borovec J., Kybic J. (2016) **Binary Pattern Dictionary Learning for Gene Expression Representation in Drosophila Imaginal Discs.** In: Computer Vision – ACCV 2016 Workshops. Lecture Notes in Computer Science, vol 10117, Springer, [http://doi.org/10.1007/978-3-319-54427-4\\_40](http://doi.org/10.1007/978-3-319-54427-4_40)

# Overview

1. Problem statement
2. Used datasets
3. Instance segmentation
  - a. Structure (tissue) segmentation on superpixels
  - b. Center detection and ellipse fitting
  - c. Region growing on superpixels
4. Binary Pattern Dictionary Learning
5. Conclusion & Future work

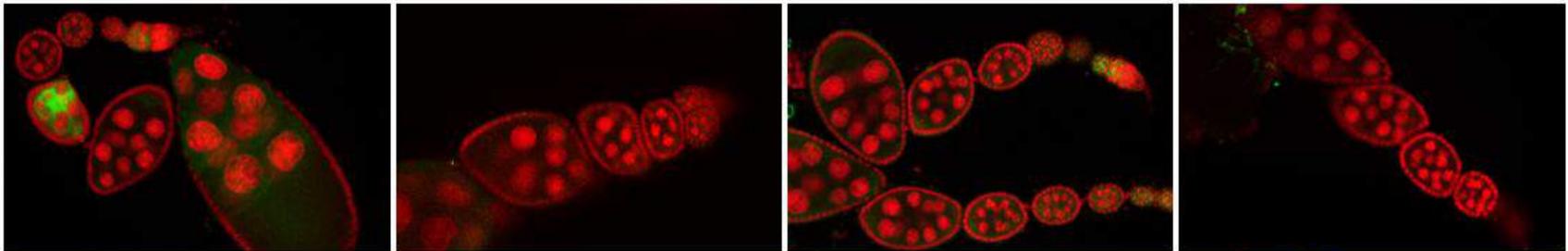
# Motivation

## Why Drosophila?

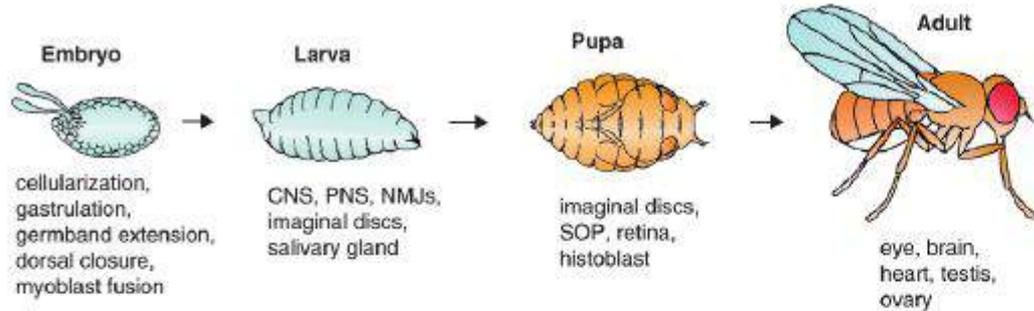
- High gene similarity with mammals (~ 93%)
- Short life cycles -> gene evaluation in generations

## Automatic image processing:

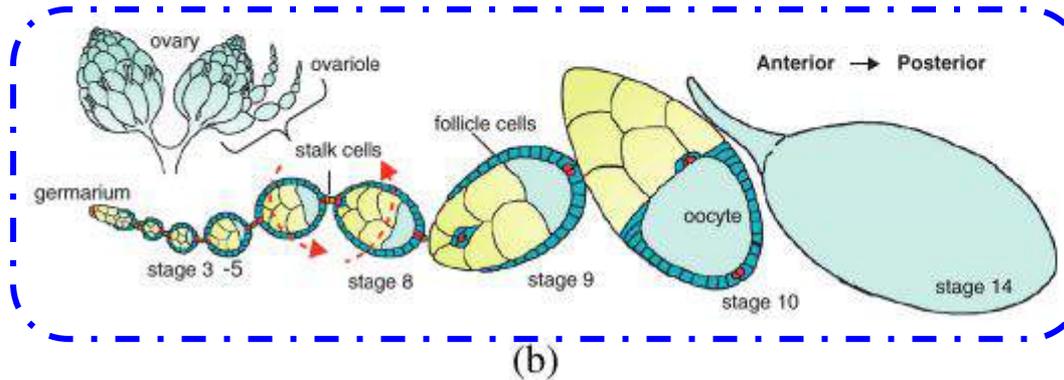
- Analyses require thousands of images to be processed
- Very time consuming for experts



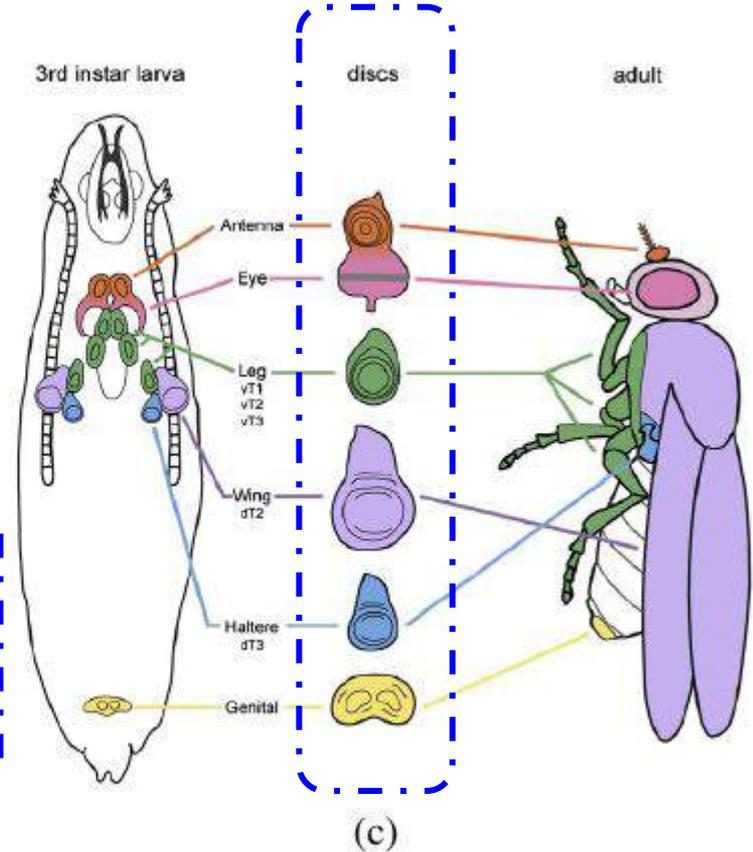
# Schema of Drosophila development



(a)



(b)



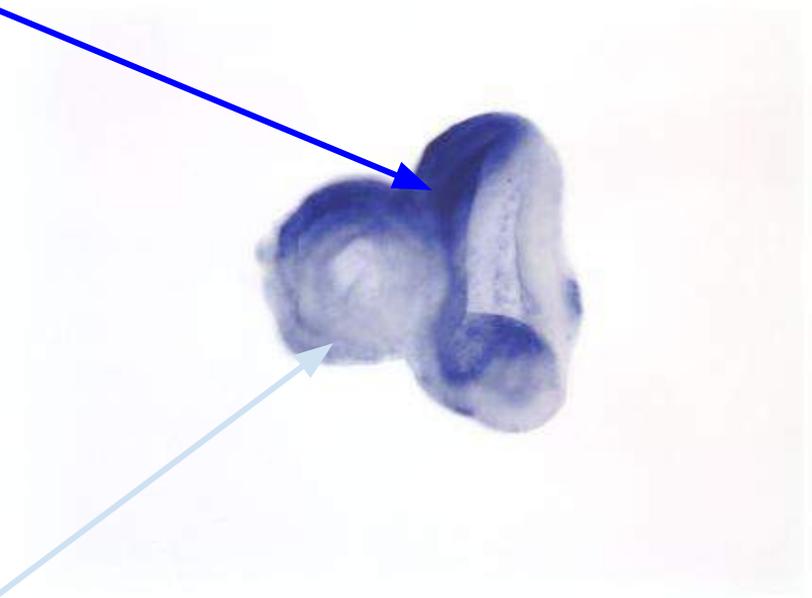
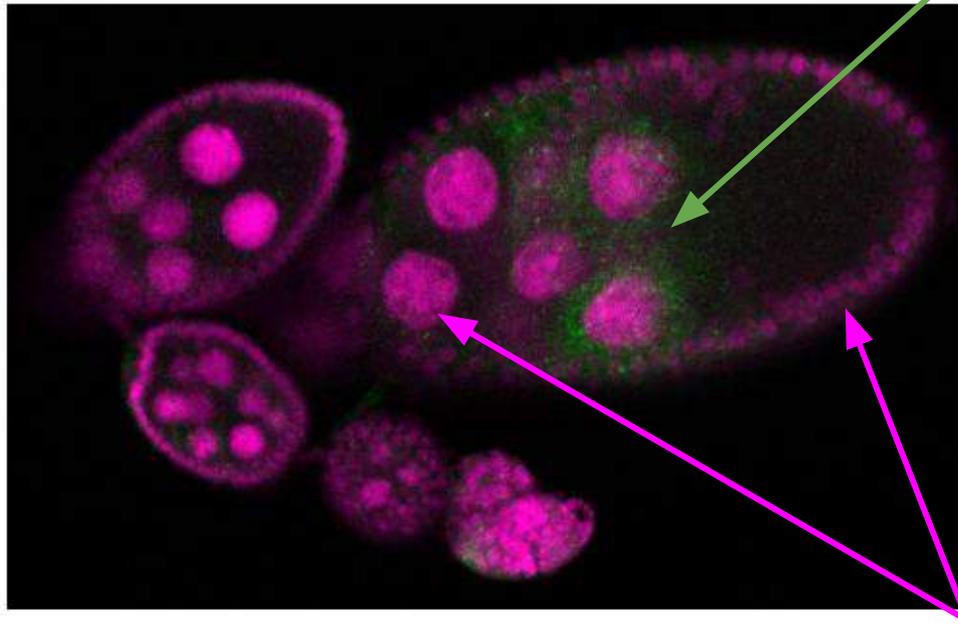
(c)

# Sample images

Drosophila ovaries

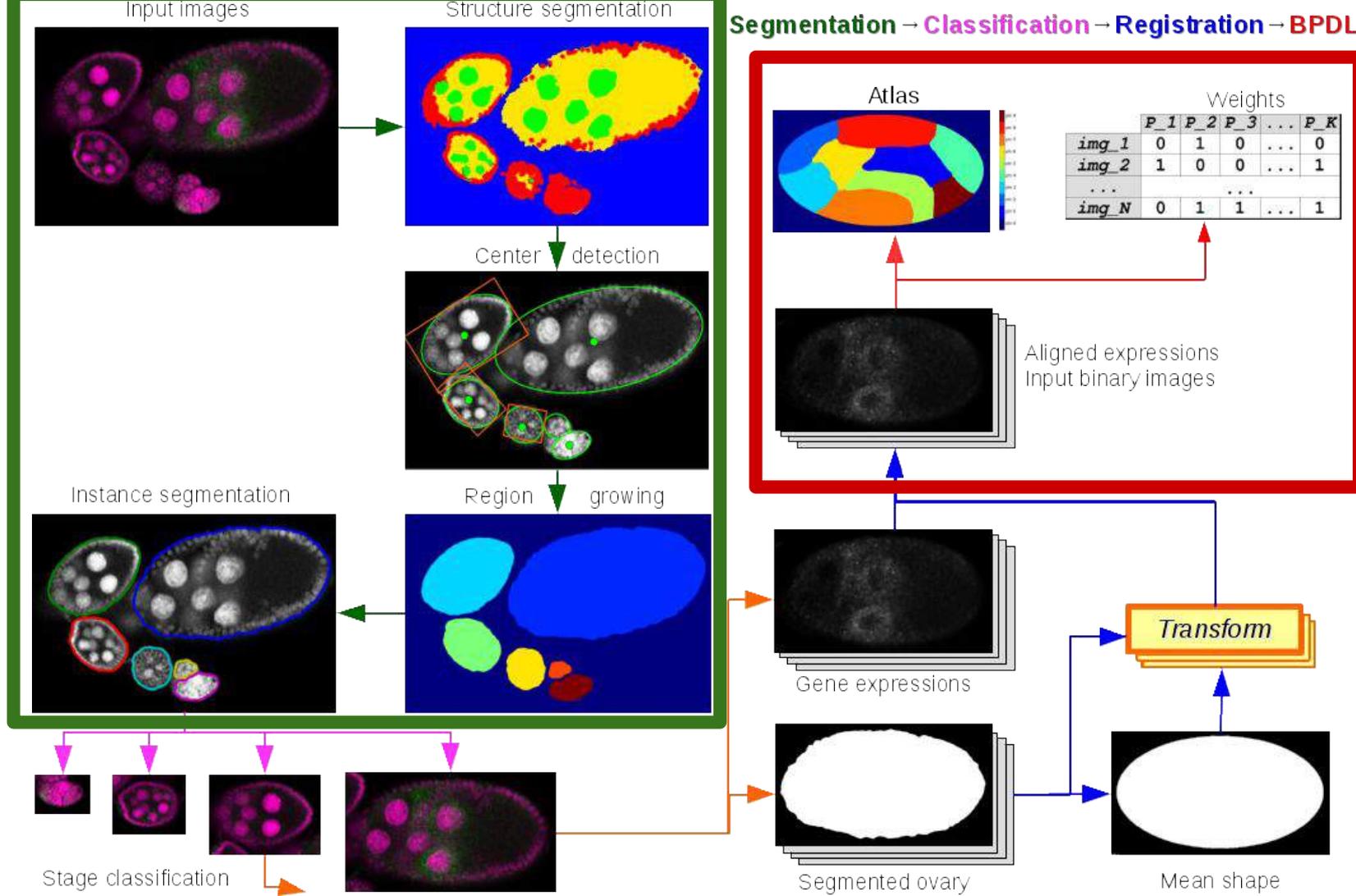
Imaginal disk

**Gene activations**



**Tissue**

# Image analysis pipeline

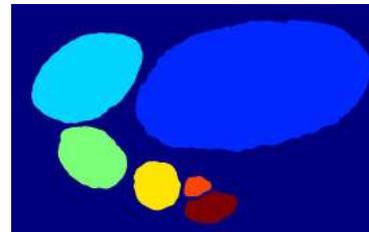
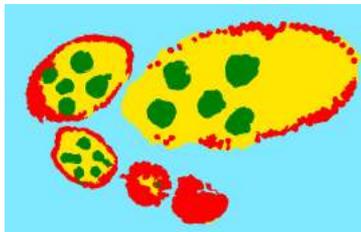
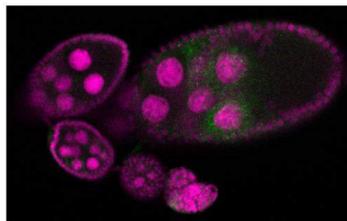


# Drosophila Datasets

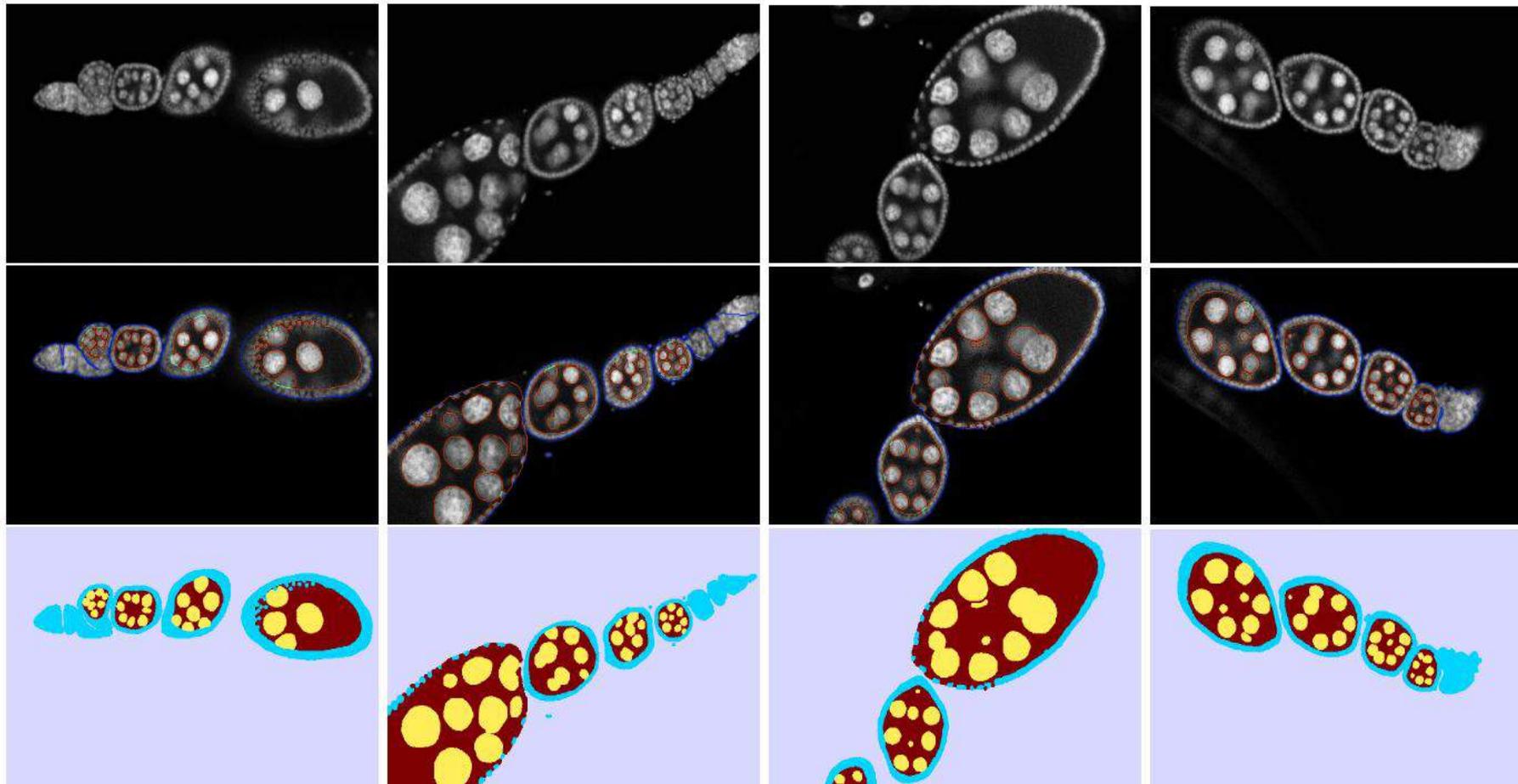
Just a few annotations in biomedical imaging is common issue...

- Noisy images with local deformations
- Low image variance (patterns)

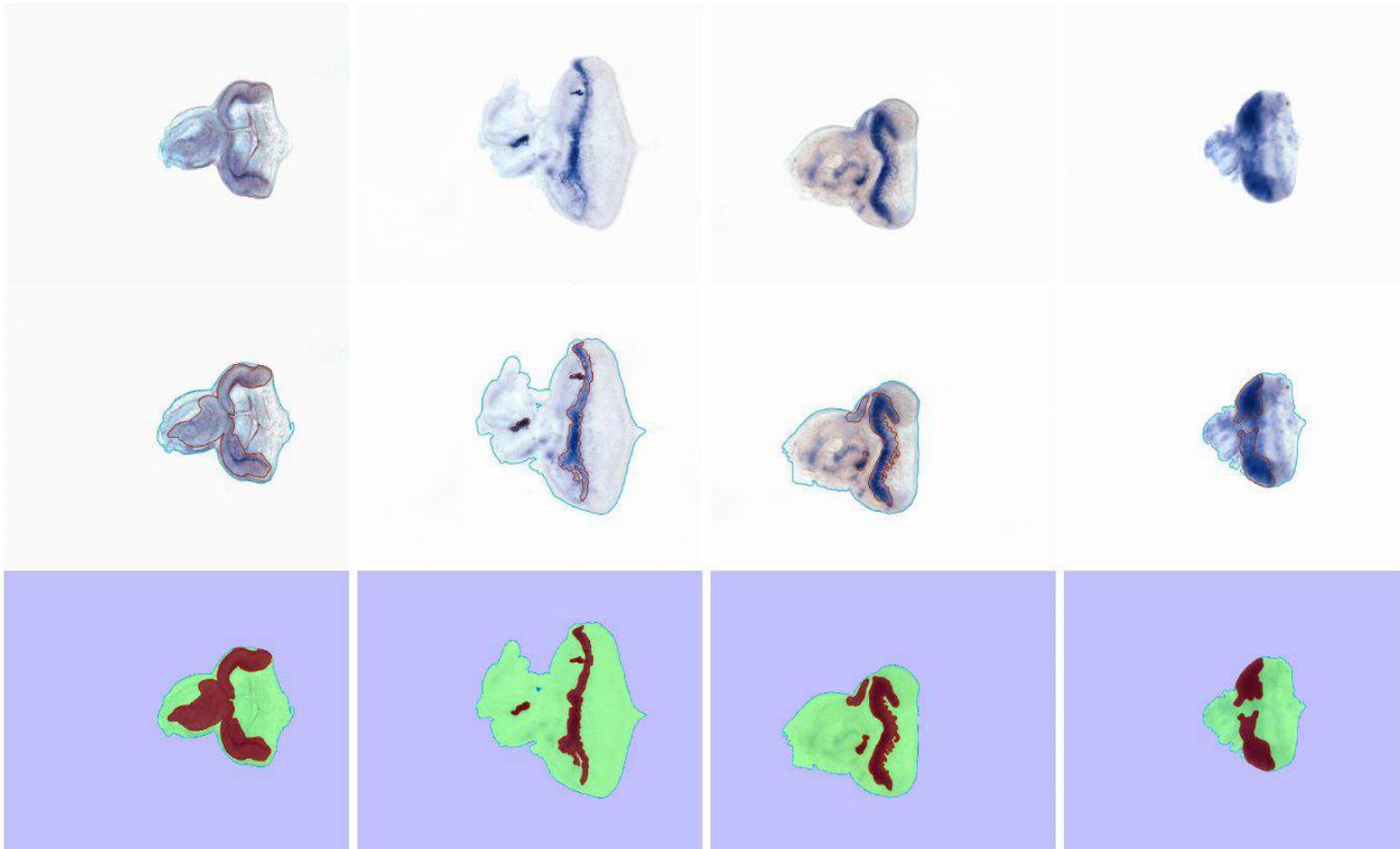
	Domain	Semantic annot.	Instance annot.
Ovaries	2.5D	72	250
Imaginal discs	2D	15	-



# Drosophila ovary - semantic segmentation



# Drosophila imaginal discs - semantic segmentation



# Notations

## Image related

$\Omega$	set of pixels (image plane)
$I$	input image function $I : \Omega \rightarrow \mathbb{R}^m$
$L$	set of labels

## Superpixels

$\eta$	original regularization
$\xi$	proposed regularization
$v$	initial superpixel size
$s$	superpixel
$S$	set of superpixels $s \in S$
$D$	weight distance
$d_{\{c,s\}}$	color and spatial distance respectively
$\Omega_s$	pixels belonging to a superpixel $s$ , $\Omega_s \subset \Omega$

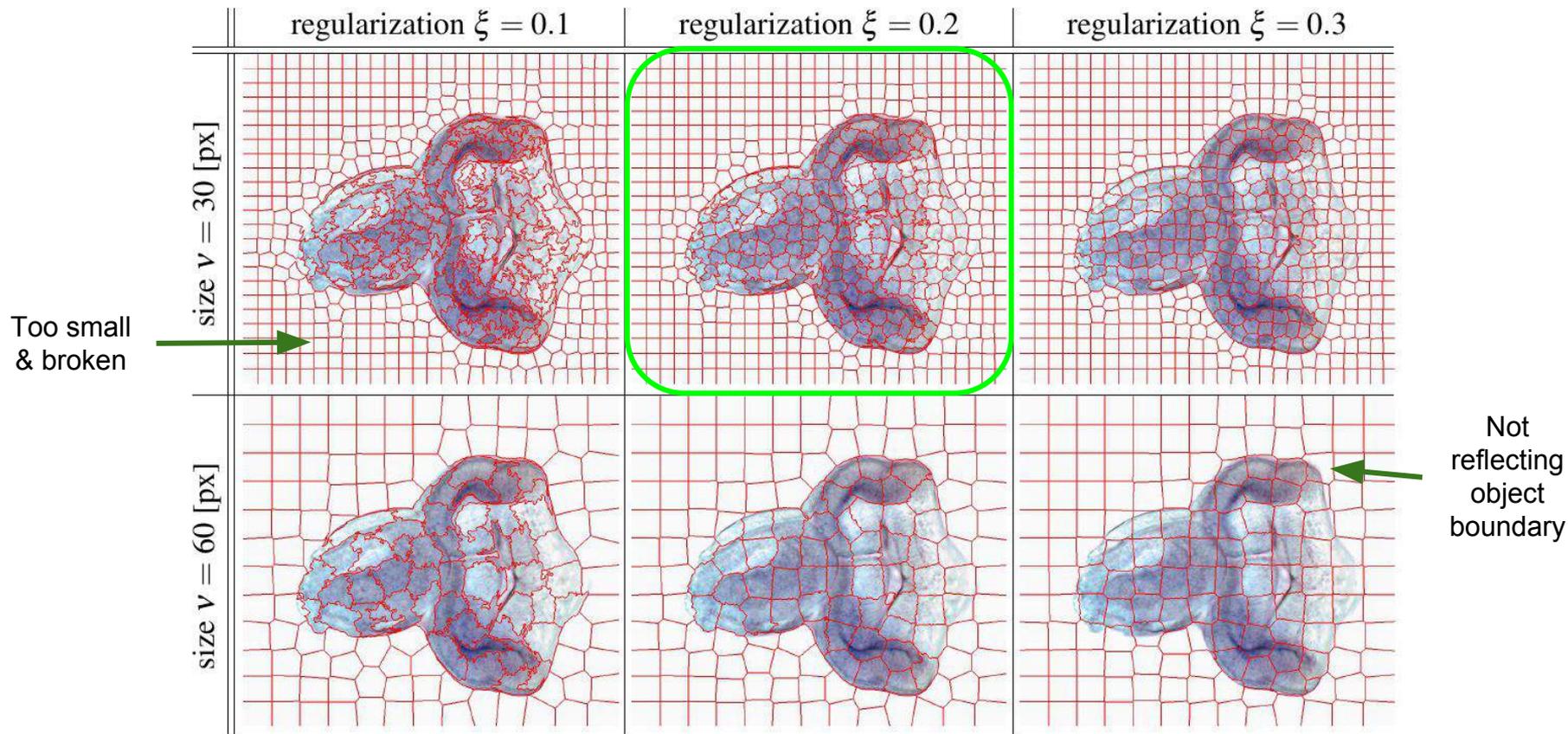
## Segmentation & Region growing

$y_\Omega$	pixel-wise segmentation function $y_\Omega : \Omega \rightarrow L$
$Y_\Omega$	ordered set of pixel-wise segmentation $Y_\Omega = y_\Omega(\Omega)$
$y$	superpixel segmentation function $y : S \rightarrow L$ with abbrev. for $y_s = y(s)$
$Y$	ordered set of superpixel segmentation $Y = y(S)$
$x$	feature vector
$X$	set of features $x_s \in X$ for all superpixels
$d_{\{M,E,T\}}$	Manhattan, Euclidean and Tchebychev distances
$U, B$	unary (data) and binary (pairwise) term for GC respectively
$g$	image descriptor
$r$	vector of ray distances
$c$	vector of object centers
$m$	statistical shape model $m = [c, r, \Theta, w]$
$M$	mixture of shape model
$w$	vector with model weights
$q$	cumulative probability of spatial prior

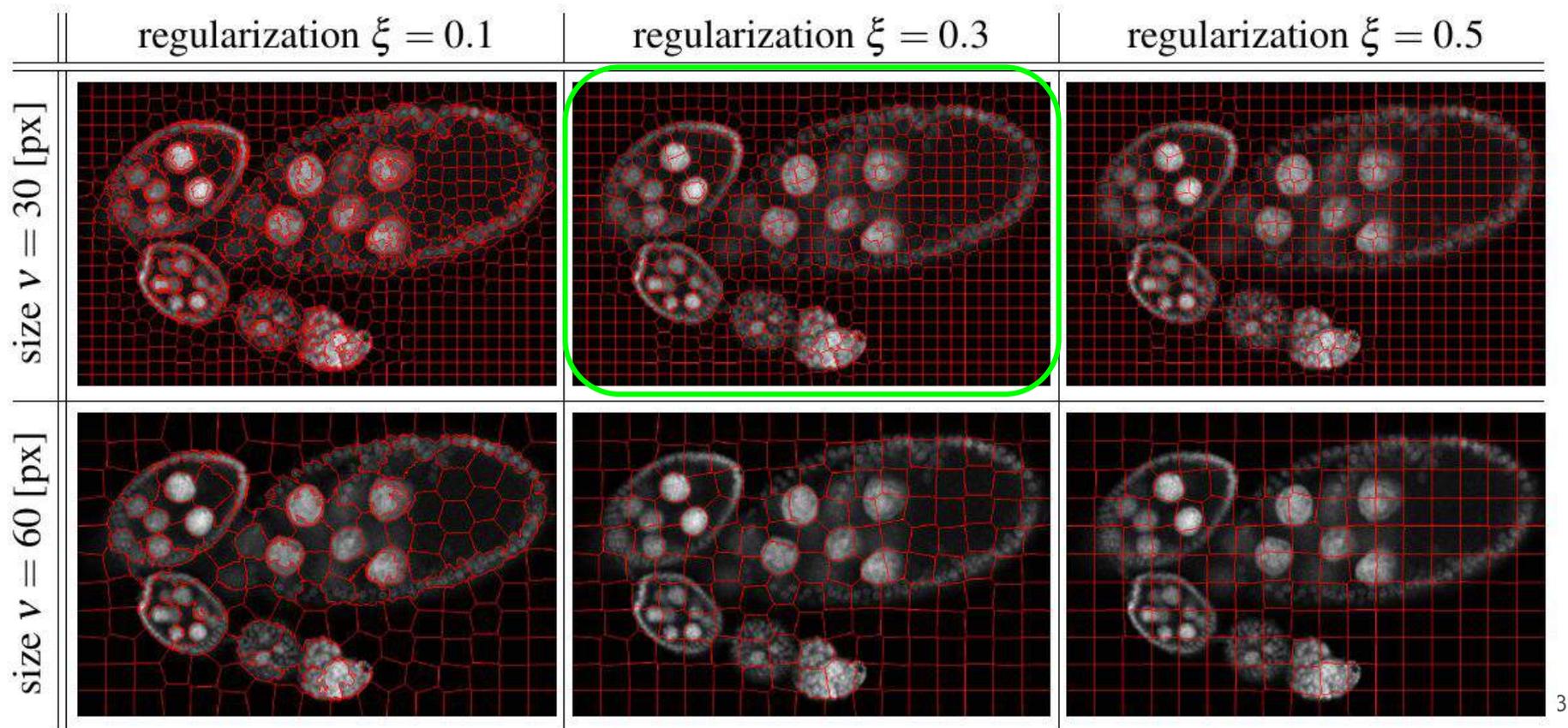
## Dictionary Learning

$g$	image appearance (binary association to a class)
$G$	set of images $g \in G$
$y_\Omega, Y_\Omega$	atlas (binary patterns, segmentation)
$w$	vector with binary weights
$W$	matrix with binary weights $w \in W$

# Superpixels - Imaginal discs



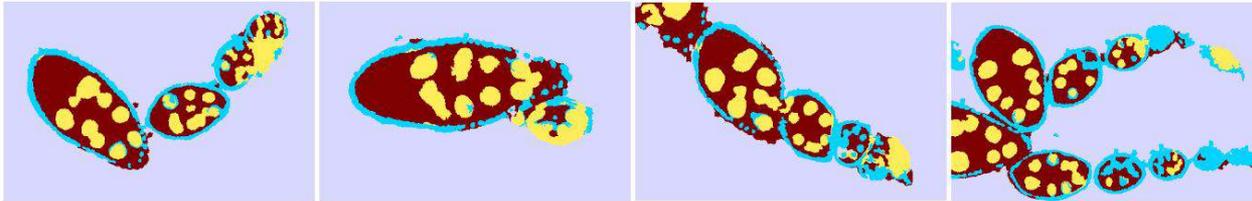
# Superpixels - Drosophila ovary



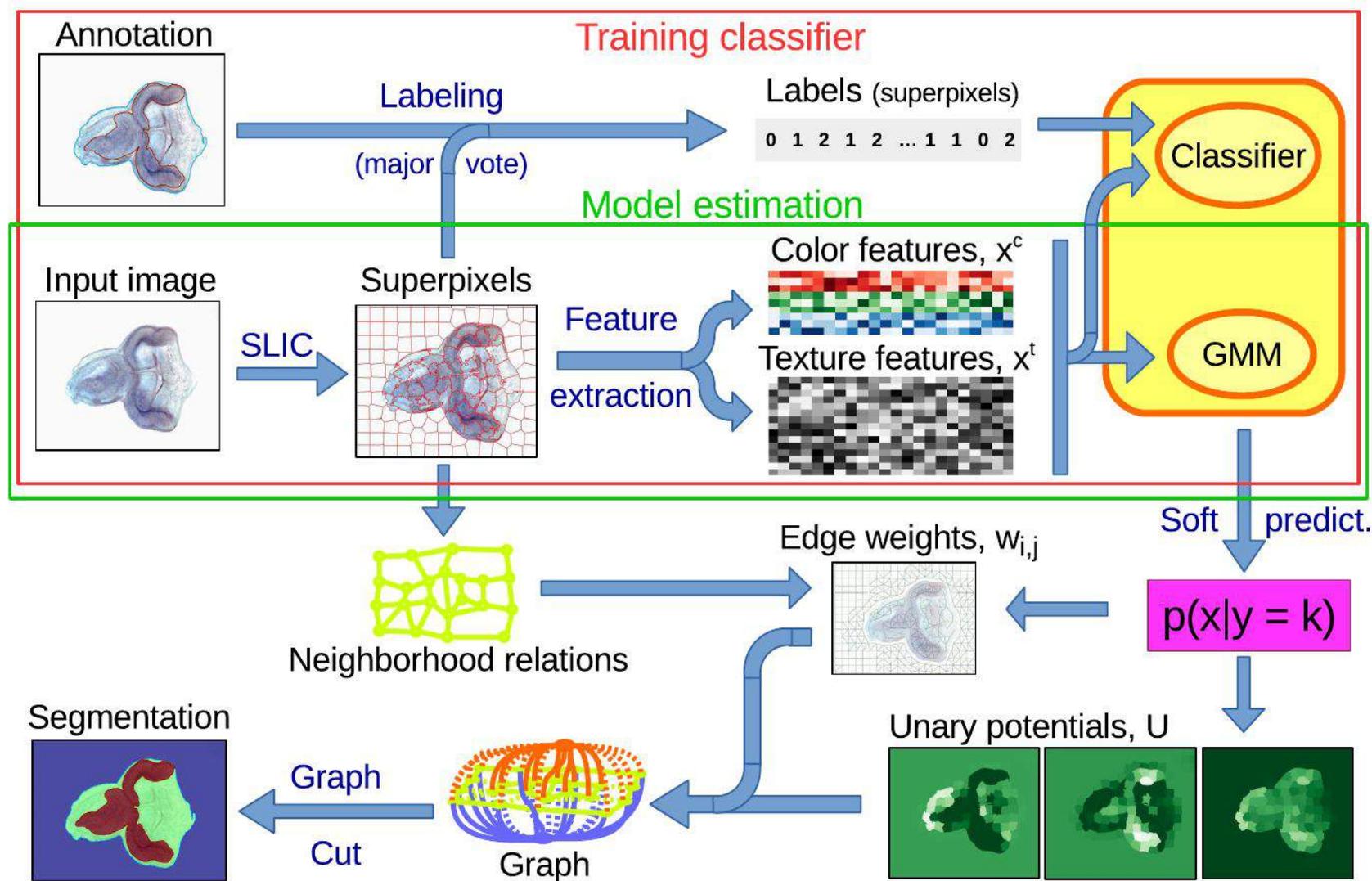
# Overview

1. Problem statement
2. Used datasets
3. Instance segmentation
  - a. Structure (tissue) segmentation on superpixels
  - b. Center detection and ellipse fitting
  - c. Region growing on superpixels
4. Binary Pattern Dictionary Learning
5. Conclusion & Future work

# *Supervised and unsupervised segmentation using superpixels, model estimation, and Graph Cut*



Jiří Borovec, Jan Švihlík, Jan Kybic, David Habart, “**Supervised and unsupervised segmentation using superpixels, model estimation, and graph cut,**” *Journal Electron. Imaging* 26(6), 061610 (2017), [DOI: 10.1117/1.JEI.26.6.061610](https://doi.org/10.1117/1.JEI.26.6.061610).



# Segmentation method overview

Image segmentation method consisting of the following steps:

1. Computation of superpixels - SLIC
2. Extraction of superpixel-based descriptors:
  - a. Color - mean, median, energy, STD
  - b. Texture - Leung-Malik filter bank
3. Calculating image-based class probabilities:
  - a. Supervised - Random Forest, k-NN, Adaboost, ...
  - b. Unsupervised - Gaussian Mixture Model
4. Spatial regularized superpixel classification using Graph Cut
  - a. Edge weights - color, features, model

# Problem formulation

Formulation (standard) as

$$Y^* = \arg \max_Y P(Y|X) = \arg \max_Y \frac{p(X|Y) \cdot P(Y)}{p(X)}$$

$$P(Y) = \prod_{s \in S} h(y_s) \cdot \prod_{(i,j) \in \mathcal{N} \subseteq S^2} R(y_i, y_j)$$

$$Y^* = \arg \max_Y \prod_{i \in S} (p(\mathbf{x}_i | y_i) \cdot h(y_i)) \cdot \prod_{(i,j) \in \mathcal{N}} R(y_i, y_j)$$

Energy minimisation

$$Y^* = \arg \min_Y \sum_s \underbrace{-\log(p(\mathbf{x}_s | y_s) \cdot h(y_s))}_{U_s(y_s)} + \sum_{(i,j) \in \mathcal{N}} \underbrace{-\log R(y_i, y_j)}_{\beta w_{i,j} B(y_i, y_j)}$$

# Superspixel features

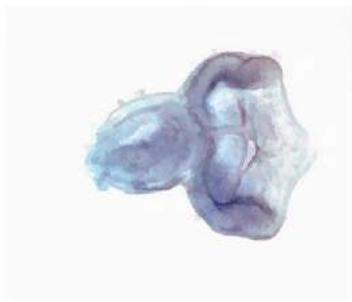
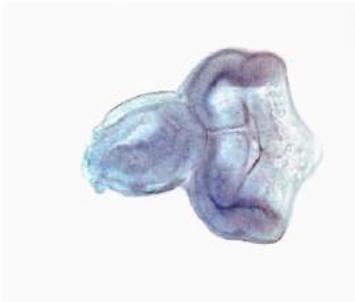
input image

$N = 15$  pixels

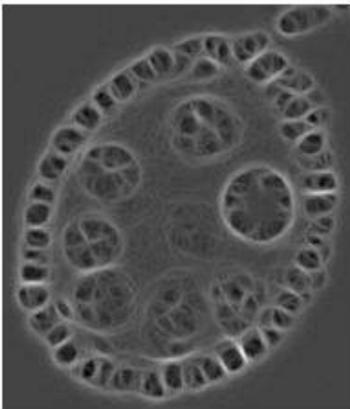
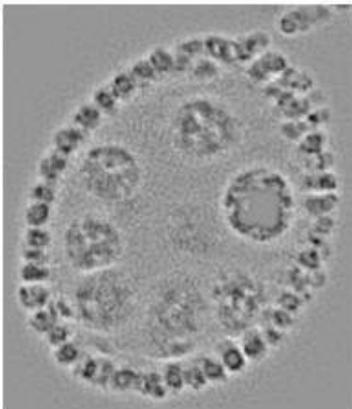
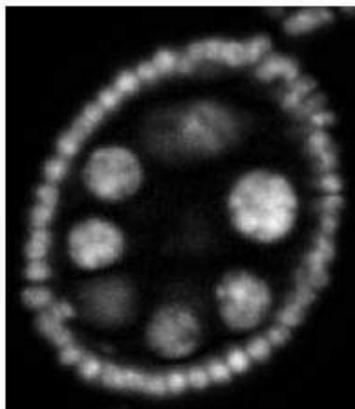
$N = 30$  pixels

$N = 45$  pixels

Superspixel  
color means



Leven-Malik  
filter bank

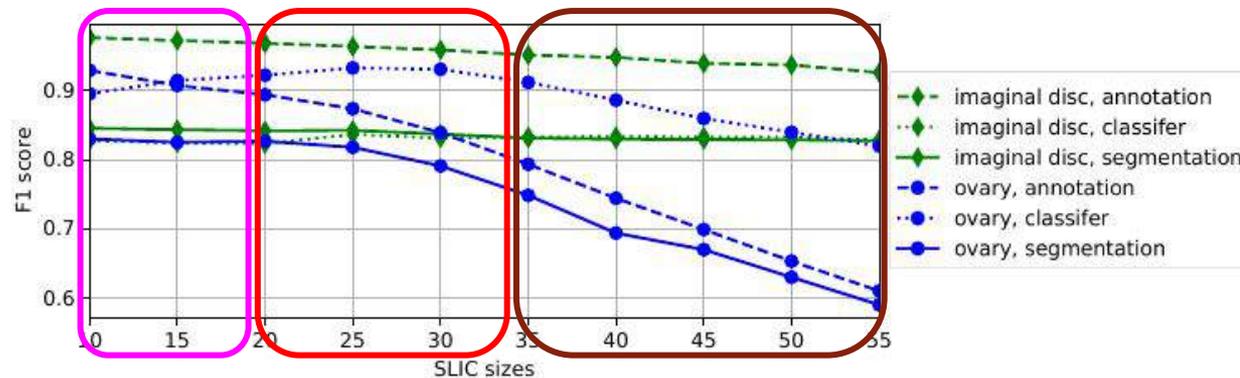


# Influence of superpixel parameters

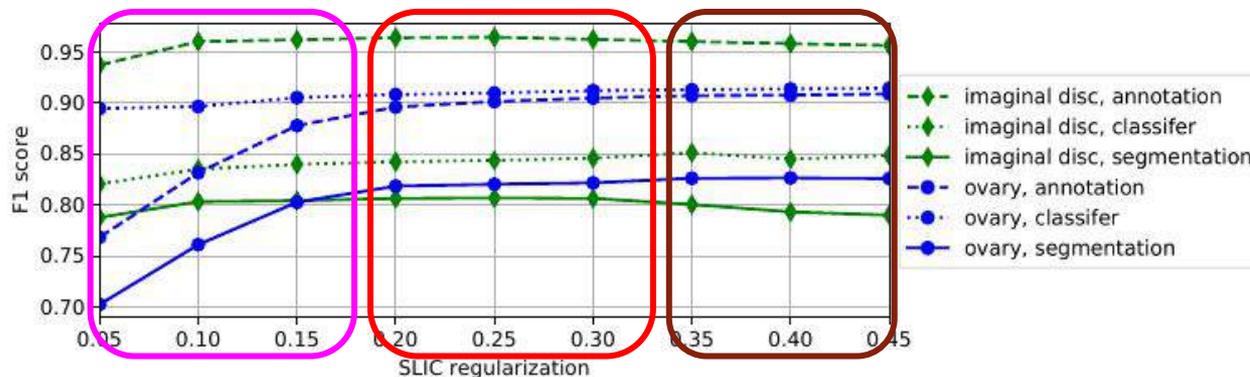
Low feature  
representativeness

Optimum  
(~compromise)

Low separability



(a)



# Graph Cut - Edge weights

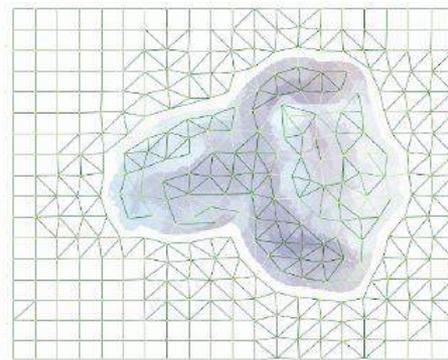
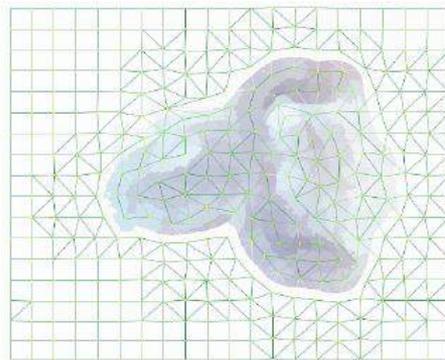
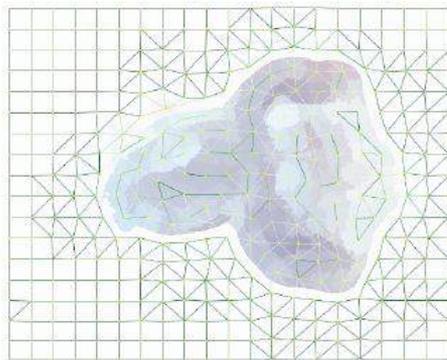
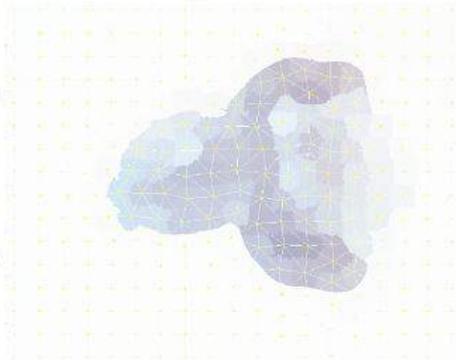
Spatial

Color

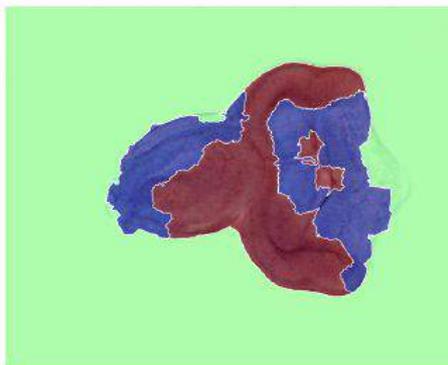
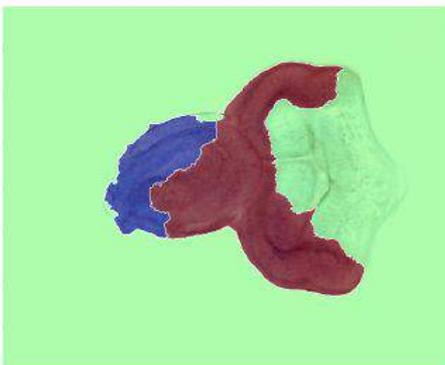
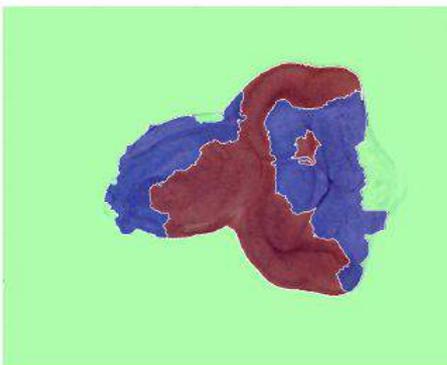
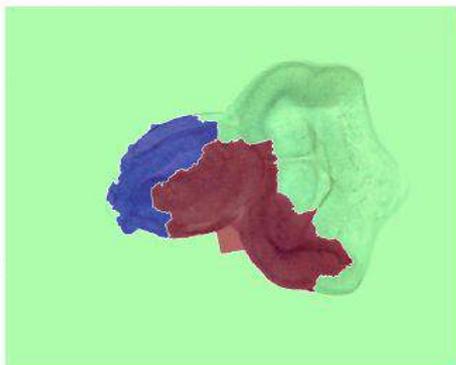
Features

Models

weighted edges



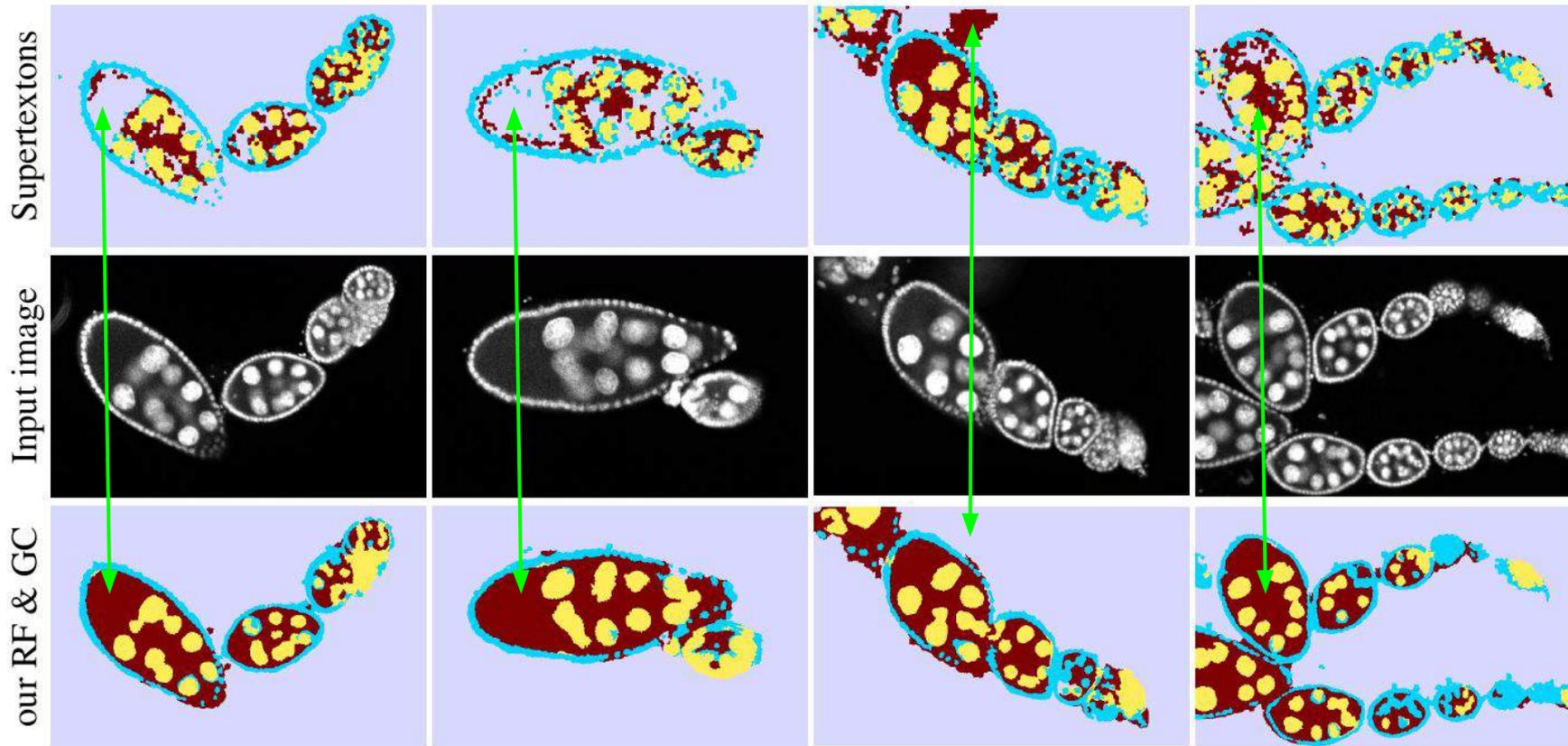
segmentation



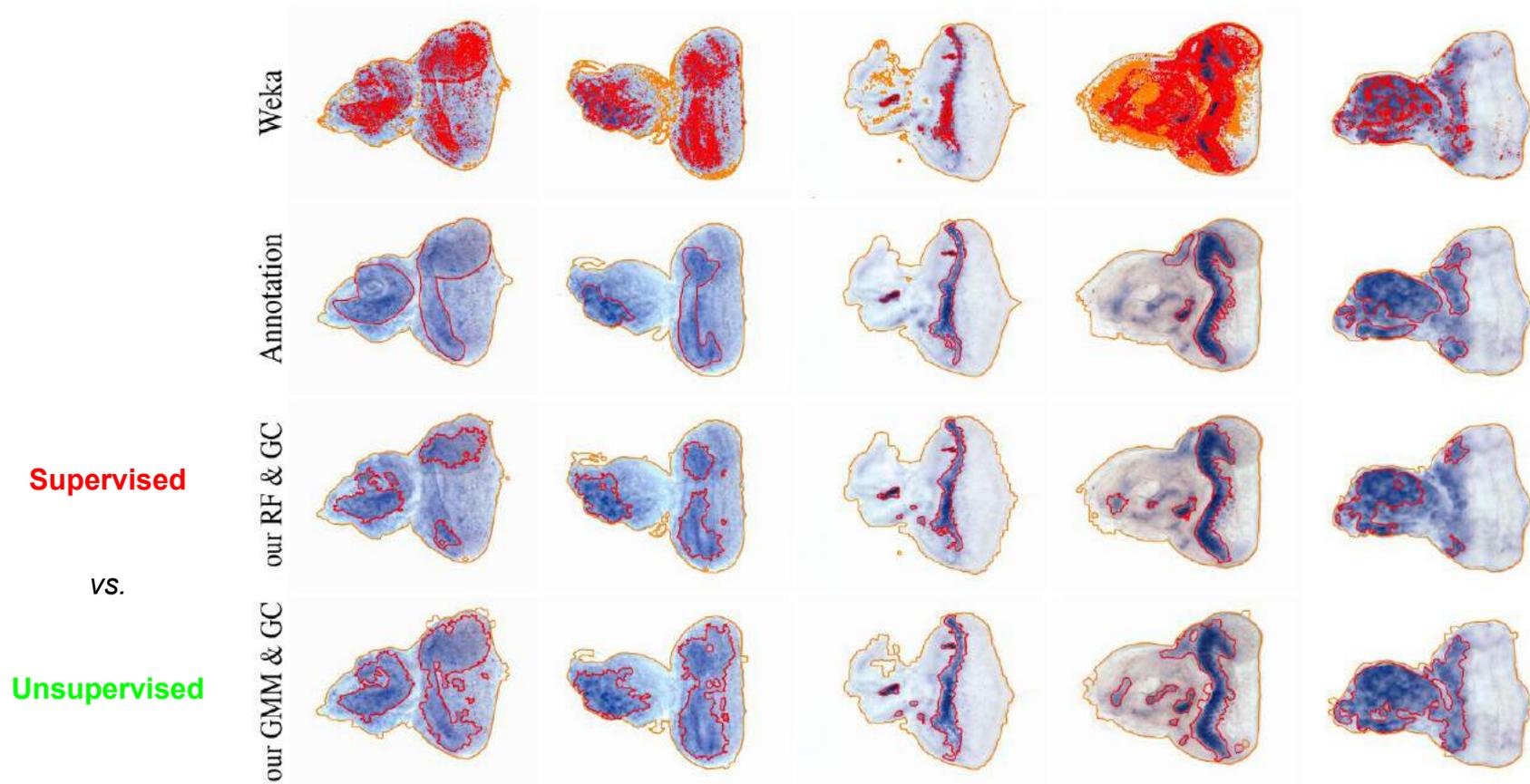
# Segmentation results with SOA (F1-score)

		Method	imaginal disc	ovary
Pixel-wise	Supervised	Weka	0.6923	0.5800
		Weka & GC(0, 100)	0.6887	0.5810
		Weka & GC(1, 50)	0.6887	0.5965
		Weka & GC(10, 50)	0.6887	0.1395
		Weka & GC(50, 100)	0.6850	0.6007
Superpixels	Supervised	ideal segm. $Y_A$	0.9696	0.9067
		Supertextons	-	0.7488
		our RF	0.8181	0.8201
		our RF & GC	<b>0.8229</b>	<b>0.8600</b>
	Unsuper.	our GMM	0.7542	0.5967
		our GMM & GC	0.7644	0.6039
		our GMM [gr]	0.7301	0.6009
		our GMM [gr] & GC	0.7564	0.6083

# Advantage of using Graph Cut



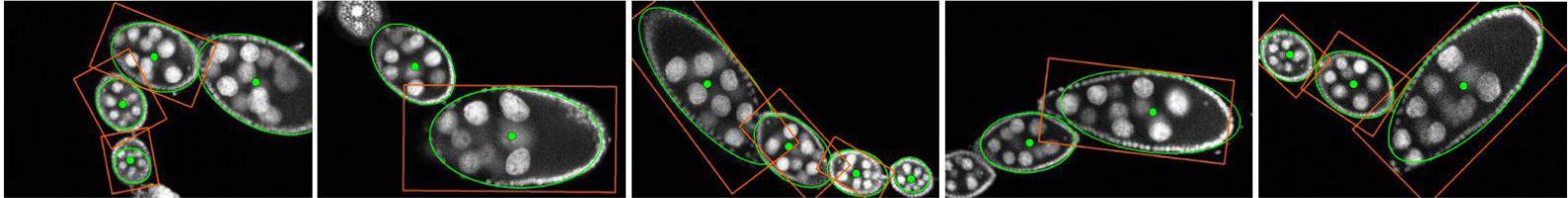
# Supervised vs Unsupervised



# Overview

1. Problem statement
2. Used datasets
- 3. Instance segmentation**
  - a. Structure (tissue) segmentation on superpixels
  - b. Center detection and ellipse fitting**
  - c. Region growing on superpixels
4. Binary Pattern Dictionary Learning
5. Conclusion & Future work

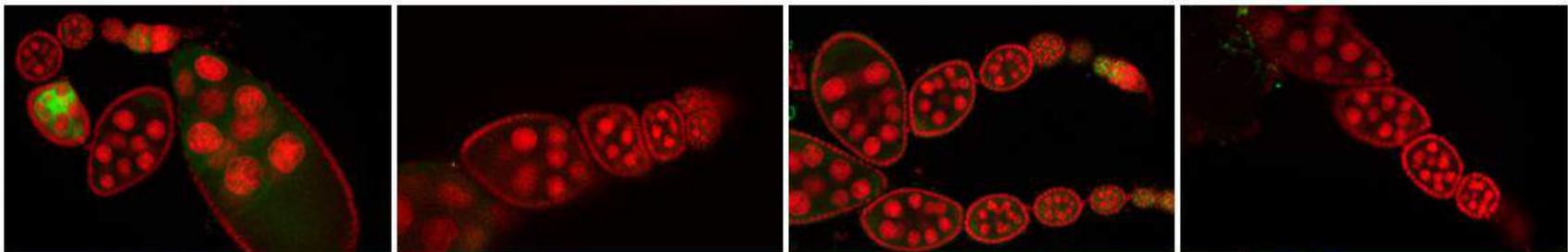
# *Detection and localization of Drosophila egg chambers in microscopy images*



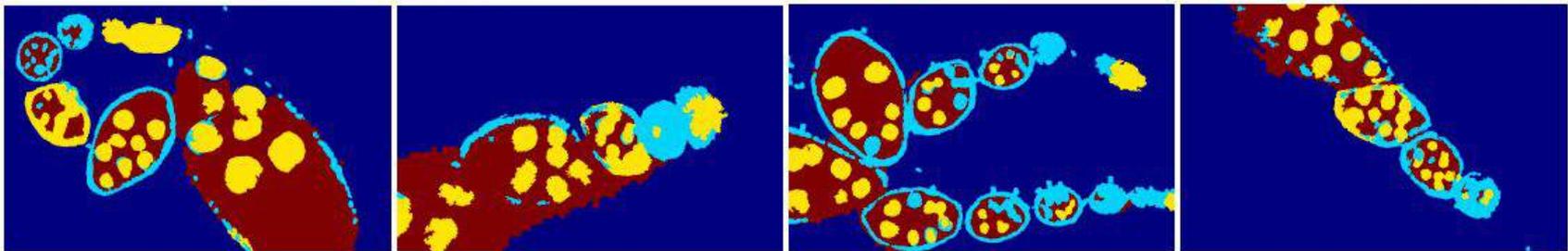
Borovec J., Kybic J., Nava R. (2017) **Detection and Localization of Drosophila Egg Chambers in Microscopy Images**. In: Machine Learning in Medical Imaging. LNCS, vol 10541. Springer, [DOI: 10.1007/978-3-319-67389-9\\_3](https://doi.org/10.1007/978-3-319-67389-9_3).

# Center detections - illustration

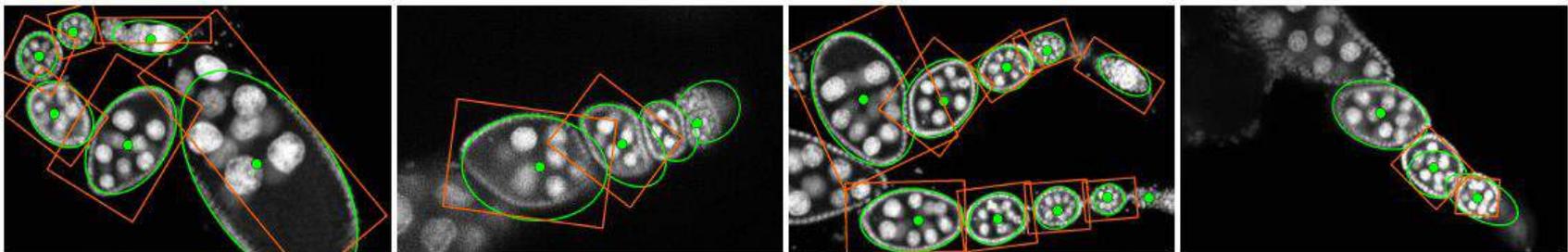
Image



Input segm.



Goal



# Schema

## 1. Extract pixel features

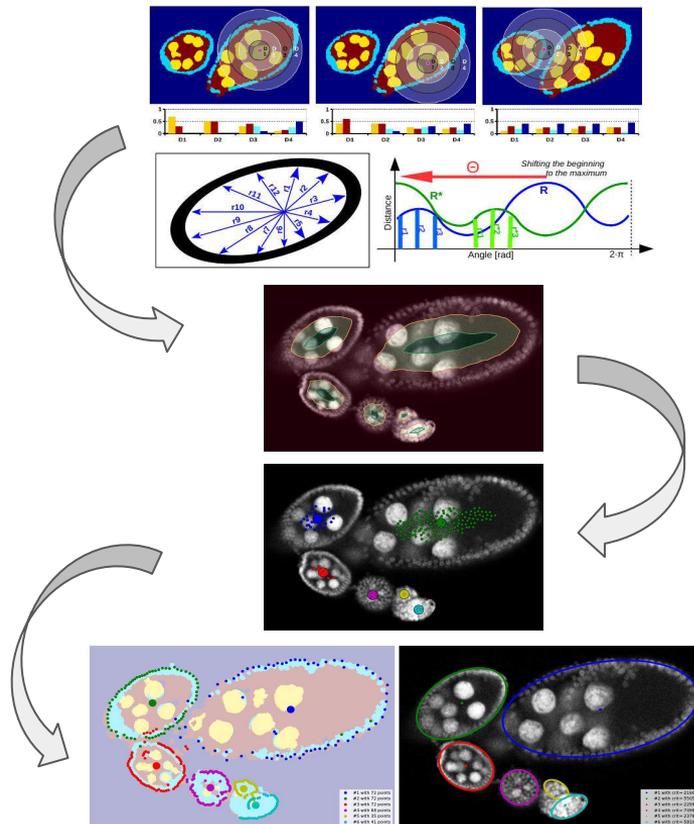
a. Label histogram

b. Ray features

## 2. Train classifier

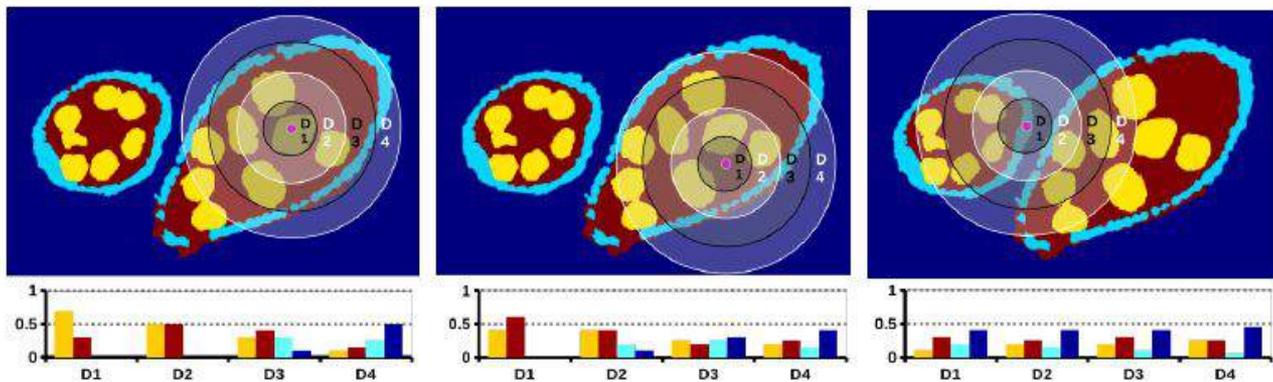
## 3. Group center candidates

## 4. Ellipse fitting

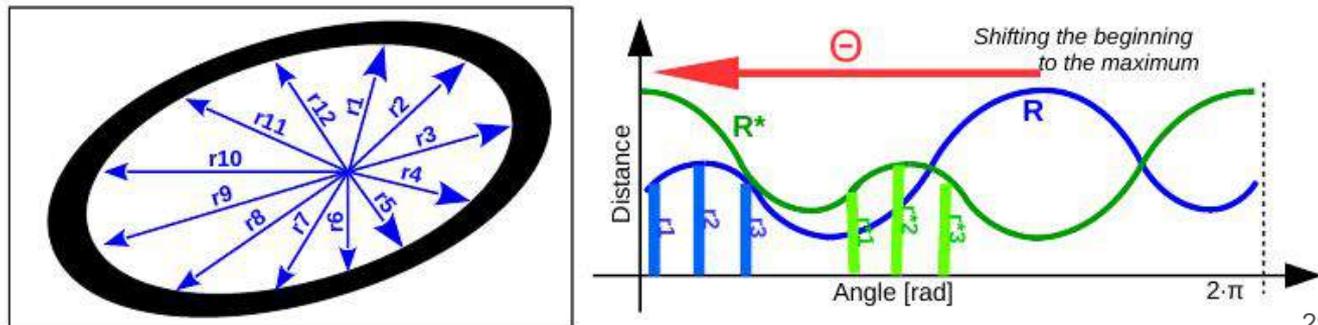


# Features for center detection

- Label histogram



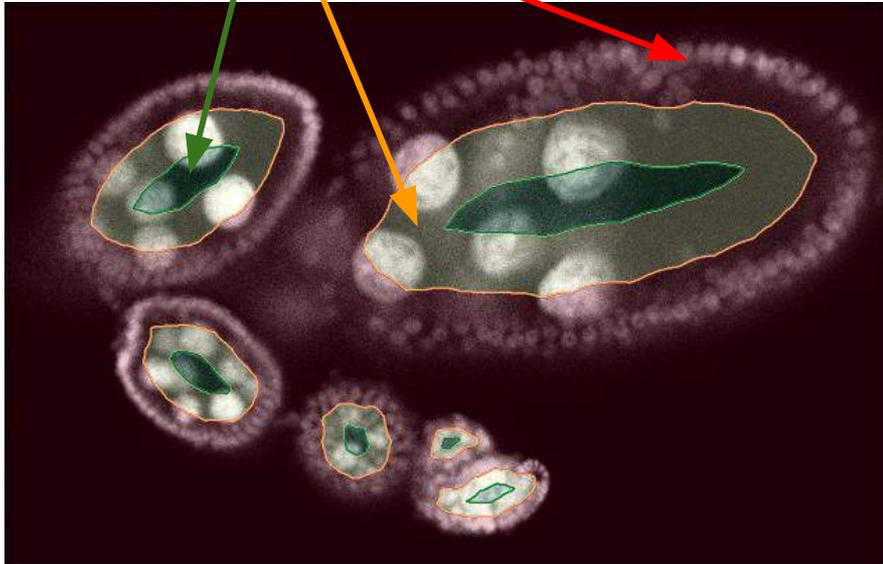
- Ray features



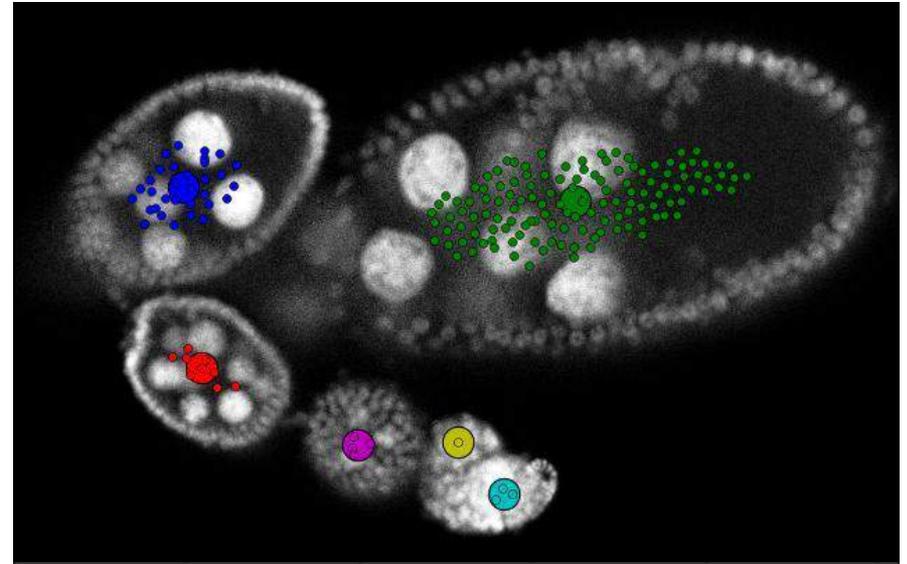
# Classification & Grouping

Train classifier on 3 zones

- Negative
- No not case
- Positive

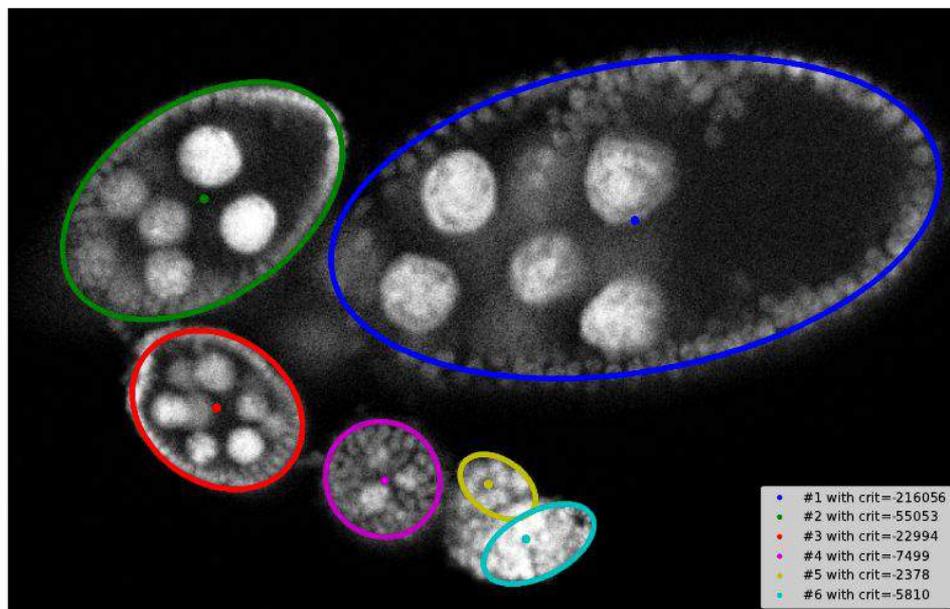
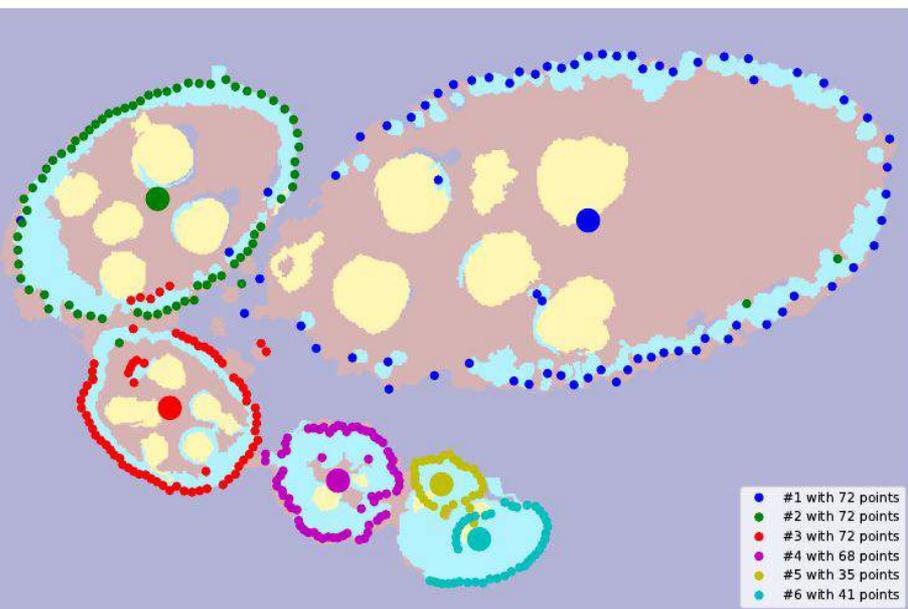


Clustering of center candidates with [DBSCAN](#)



# Ellipse fitting

Fitting ellipses to boundary points to maximise foreground labels inside ellipse



# Ellipse fitting

Formulate as Maximize likelihood

$$\arg \max_{F,B} \prod_{i \in \Omega_F} P_F(Y_i) \cdot \prod_{i \in \Omega \setminus \Omega_F} P_B(\mathbf{y}_i)$$

$$h_{\bullet} = -\log P_{\bullet}$$

$$\sum_{i \in \Omega} h_B(Y_i) = \sum_{i \in \Omega_F} h_B(Y_i) + \sum_{i \in \Omega \setminus \Omega_F} h_B(Y_i)$$

$$\arg \min_{F,B} \sum_{i \in \Omega_F} h_F(Y_i) + \sum_{i \in \Omega} h_B(Y_i) - \sum_{i \in \Omega_F} h_B(Y_i)$$

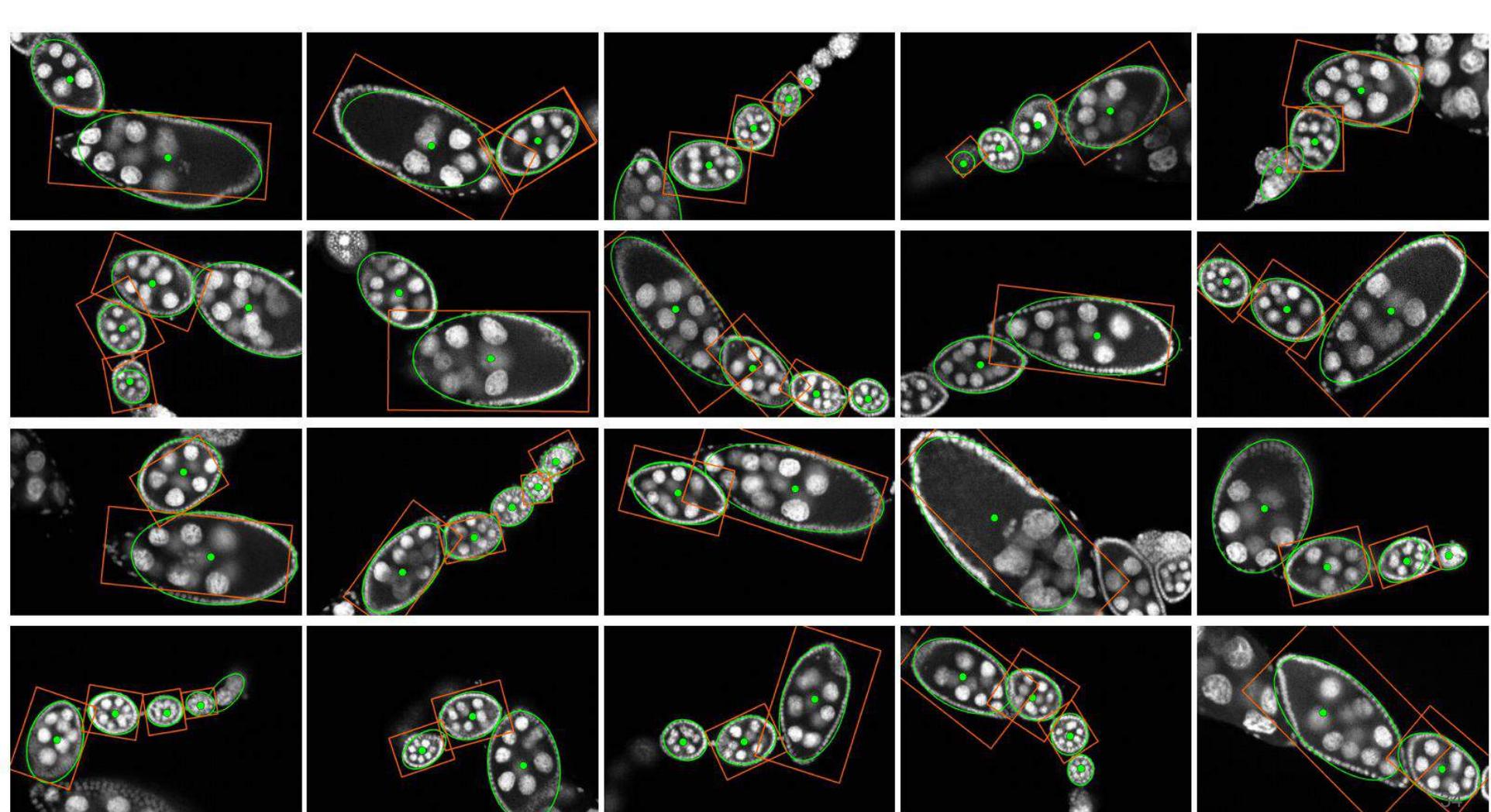
$\sum_{i \in \Omega} h_B(Y_i)$  is constant

$$\arg \min_{F,B} \sum_{i \in \Omega_F} h_F(Y_i) - h_B(Y_i)$$

# Ellipse pruning

Ellipse approximation eliminate multiple center detection in single egg

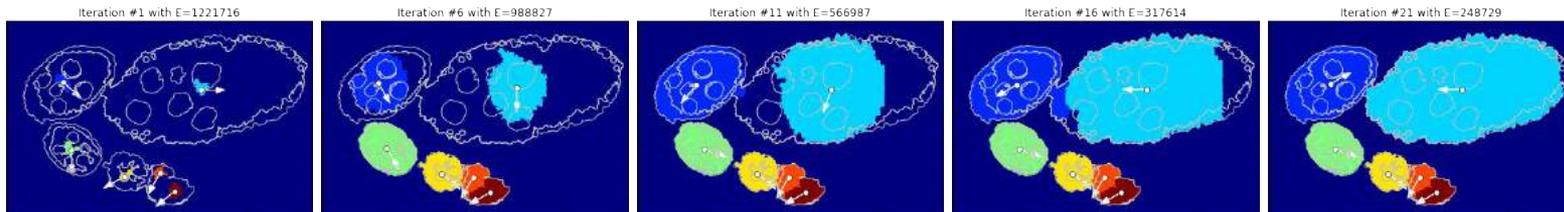
Egg chambers	Stage				
	1	2	3	4	5
number	921	1403	865	834	836
false negatives	306 (33%)	158 (11%)	6 (0.7%)	1 (0.1%)	0 (0.0%)
multiple detections (MD)	37 (4.0%)	31 (2.2%)	109 (12%)	80 (9.6%)	90 (11%)
MD after ellipse fitting	18 (2.0%)	13 (0.9%)	27 (3.1%)	20 (2.4%)	30 (3.6%)
false positives	43 (0.9%)				



# Overview

1. Problem statement
2. Used datasets
- 3. Instance segmentation**
  - a. Structure (tissue) segmentation on superpixels
  - b. Center detection and ellipse fitting
  - c. Region growing on superpixels
4. Binary Pattern Dictionary Learning
5. Conclusion & Future work

# *Region growing using superpixels with learned shape prior*



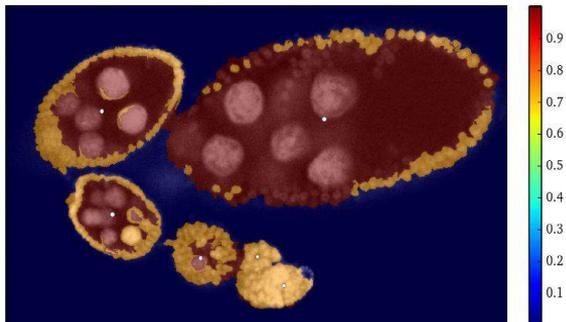
Jiří Borovec, Jan Kybic, Akihiro Sugimoto, “**Region growing using superpixels with learned shape prior,**” *Journal Electron. Imaging* 26(6), 061610 (2017), [DOI: 10.1117/1.JEI.26.6.061611](https://doi.org/10.1117/1.JEI.26.6.061611).

# Region growing - variational framework

Formulated as:

$$P(\mathbf{y}_s | y, M) = \frac{1}{Z(M, y)} P_g(g | y) P_m(g | M) P_R(g)$$

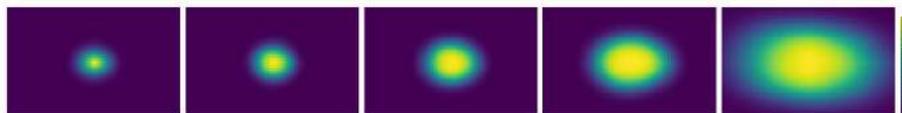
Appearance model



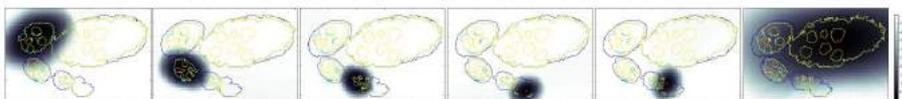
Pairwise term

$$P_R(g) = \prod_{(u,v) \in \mathcal{N}_S} H(\mathbf{y}(u), \mathbf{y}(v))$$

Mixture model



(b)



(c)

# Region growing - variational framework

Where:

$$P_g(g | y) = \prod_{i \in \Omega} P_g(g(s(i)) | y(s(i))) = \prod_{s \in S} P_g(\mathbf{y}_s | y(s))^{|\Omega_s|}$$

$$P_m(g | M) = \prod_{i \in \Omega} P_m(g(s(i)) | M) = \prod_{s \in S} P_m(\mathbf{y}_s | M)^{|\Omega_s|}$$

$$P_R(g) = \prod_{(u,v) \in \mathcal{N}_S} H(\mathbf{y}(u), \mathbf{y}(v))$$

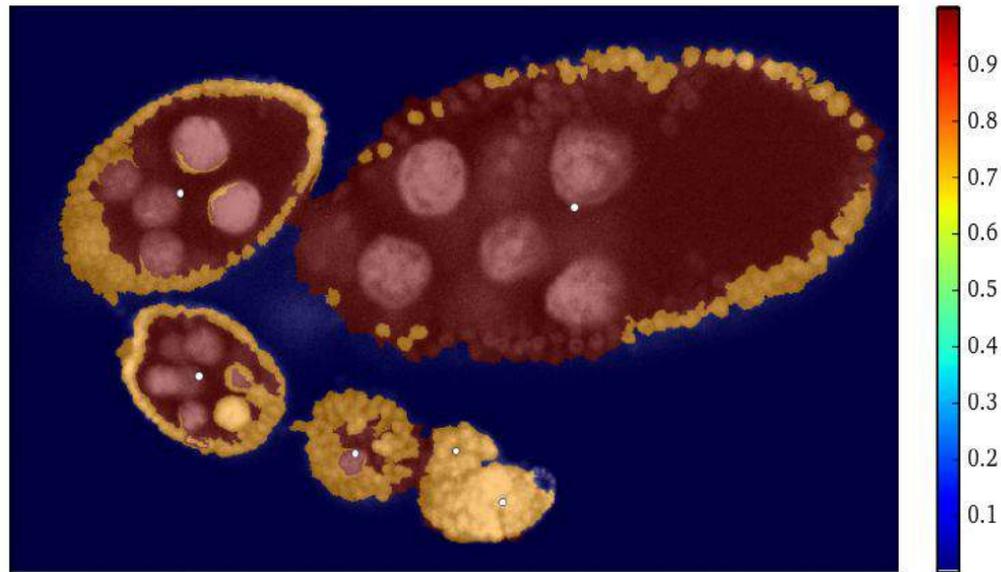
Resolves in energy minimisation:

$$E'(g) = \sum_{s \in S} |\Omega_s| \underbrace{\left[ D_s(\mathbf{y}_s) + \beta V_s(\mathbf{y}_s) \right]}_{U_s(\mathbf{y}_s)} + \gamma \sum_{(u,v) \in \mathcal{N}_S} B(\mathbf{y}(u), \mathbf{y}(v))$$

# Appearance model

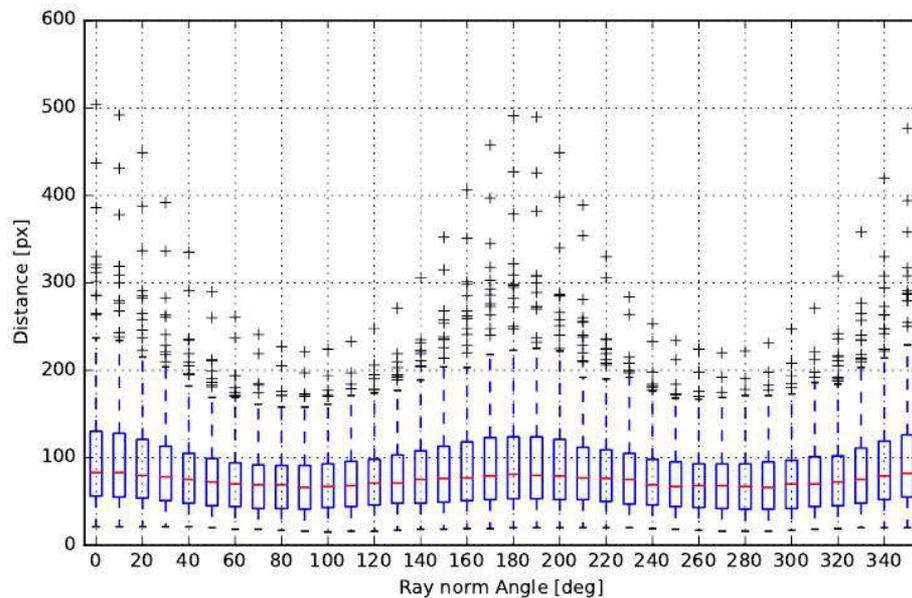
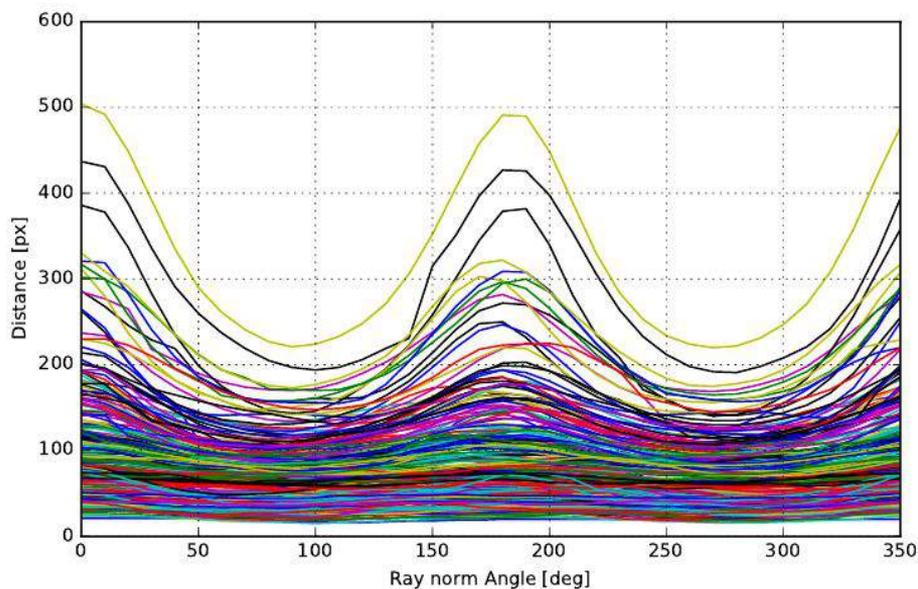
Associating a probability for each pixel (superpixel) whether it belongs to an object or not

$$P_g(\mathbf{y}(s)|g_s) = \begin{cases} P_g(g_s) & \text{for } \mathbf{y}(s) \neq 0 \\ 1 - P_g(g_s) & \text{for } \mathbf{y}(s) = 0 \end{cases}$$



# Measured shapes

Measured egg shapes as ray features from training examples (~250 eggs)

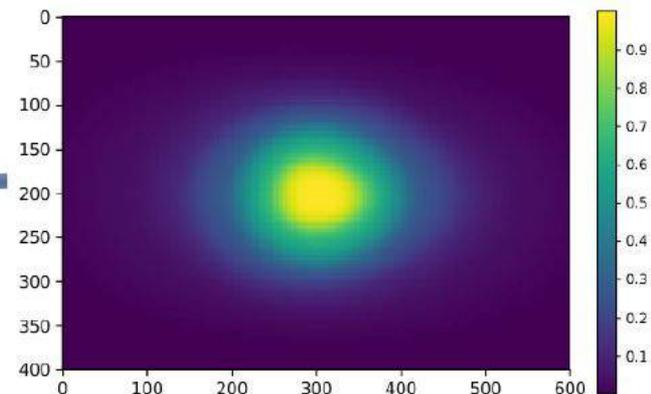
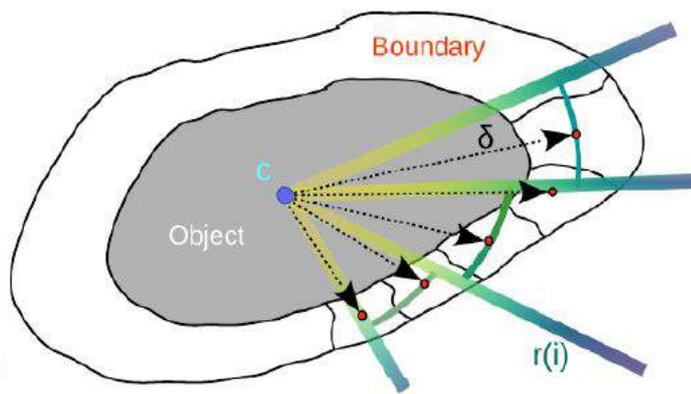
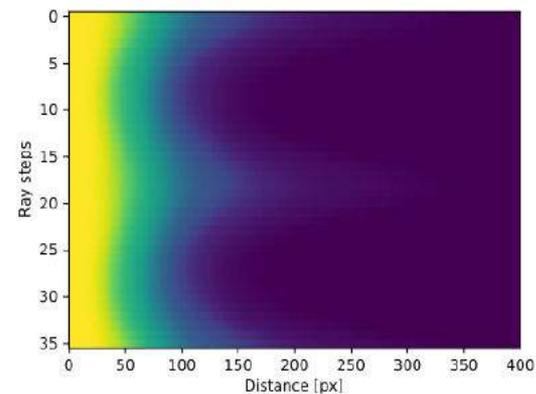


# Shape model & prior

Gaussian model from all measured Ray features

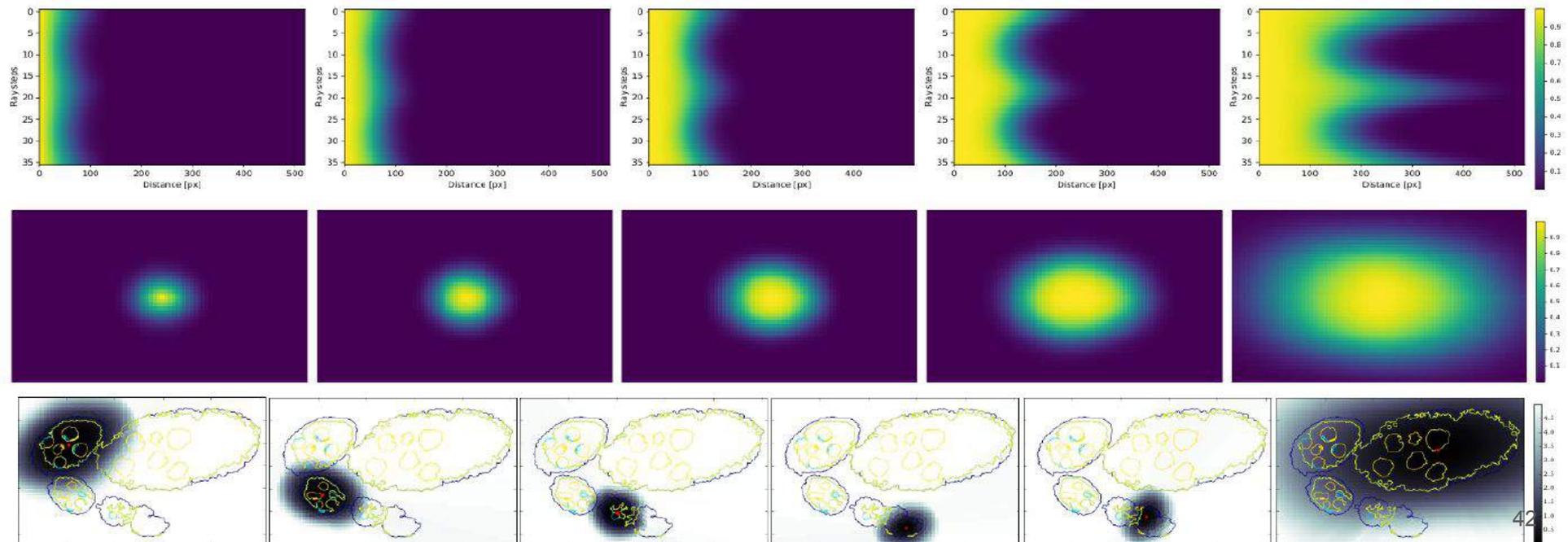
Prior is represented as integral of probabilities

$$q(s, m_k) = \int_{\delta}^{\infty} \rho(r) dr = 1 - \int_0^{\delta} \rho(r) dr$$



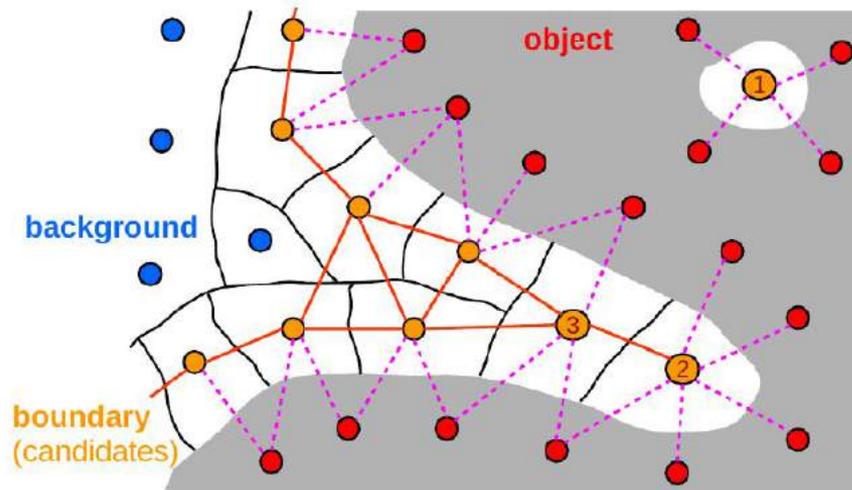
# Mixture of Gaussian models

$$P_m(\mathbf{y}_s = k | M) = \begin{cases} q(s, m_k) & \text{for } k > 0 \\ \prod_l (1 - q(s, m_l)) & \text{for } k = 0 \end{cases}$$



# RG - optimisation

- Iterative approach on object boundaries
- Alternating: region growing & update shape prior
- Strategies:
  - Greedy
  - Graph Cut
    - Binary
    - Multi-class
  - Object swapping



---

**Algorithm 1:** Region growing.

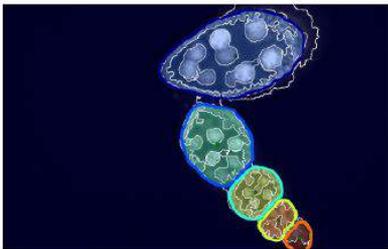
---

**Input:**  $S$ : superpixels,  $g$ : superpixel descriptors,  $c_k$ : initial object centers,  $M$ : mixture of statistical shape models

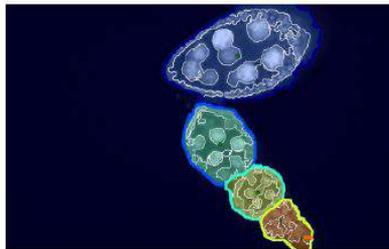
**Output:** object segmentation  $\mathbf{y}$

- 1 compute data cost  $D$ ;
  - 2 from object centers  $c_k$  set initial segmentation  $\mathbf{y}$  and model shape parameters  $m_k$ ;
  - 3 compute shape cost  $V$ ;
  - 4 **while** *not converged* **do**
    - 5 | update object pose parameters  $c_k$  and  $\Theta_k$  ;
    - 6 | **if** *significant change of center  $c_k$  position, orientation  $\Theta_k$  and object area* **then**
      - 7 | | update remaining object shape parameters  $m_k$ ;
      - 8 | | update shape costs  $V$  for all  $s$  close to  $k$ ;
    - 9 | **end**
    - 10 | find superpixels  $\partial S_k$  on the external object boundary of  $k$ ;
    - 11 | optimize (7.10) wrt  $\mathbf{y}$  by changing  $s \in \partial S_k$  using the greedy or Graph Cut algorithms;
  - 12 **end**
-

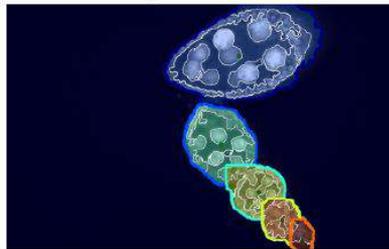
Annotation



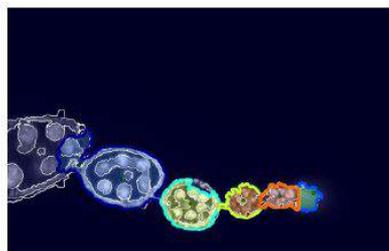
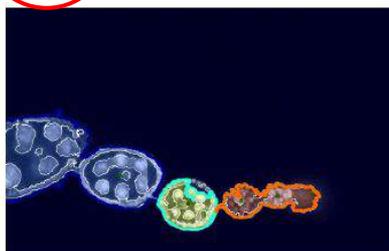
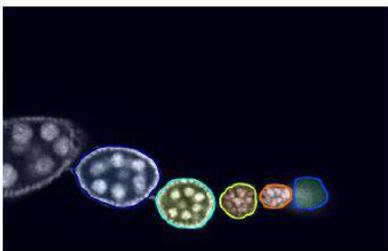
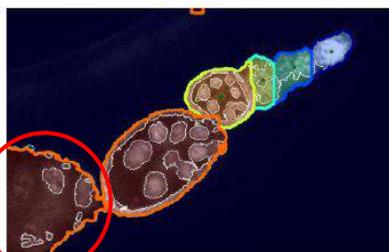
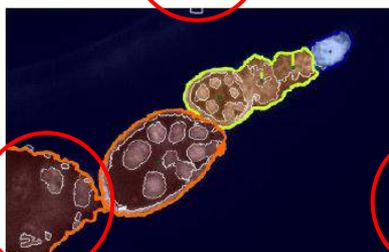
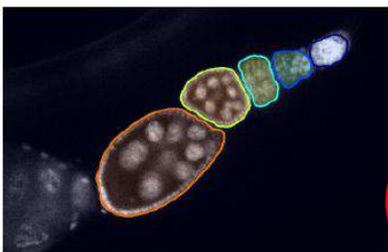
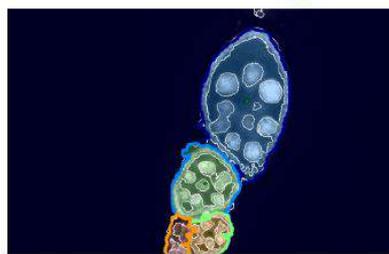
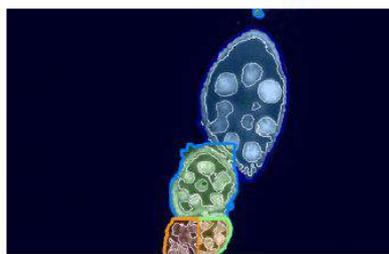
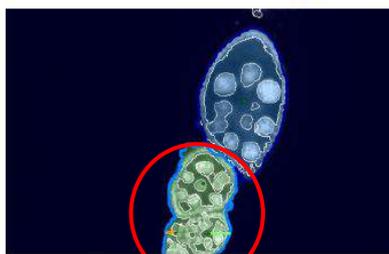
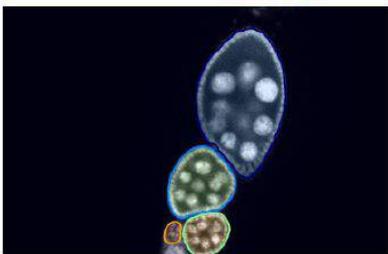
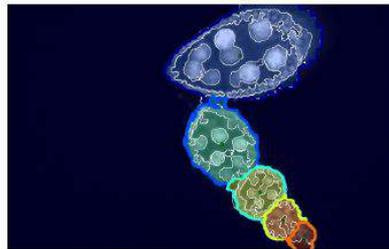
Watershed



GC (pixel-wise)



RG2Sp



# Result compared to SOA

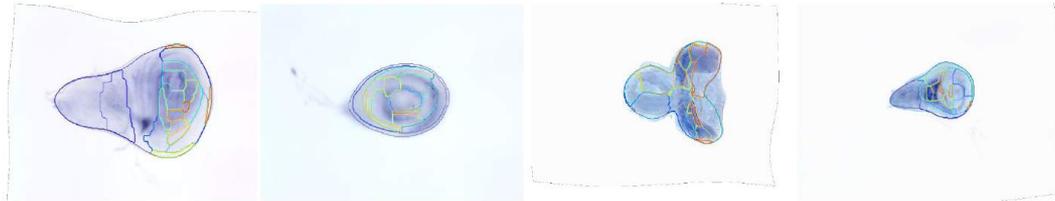
High Jaccard index with reasonable processing time

	Jaccard	accuracy	$F_1$ score	precision	recall	time [s]
Watershed	0.5705	0.9246	0.9246	0.9246	0.9246	5
Watershed (w. morph.)	0.5705	0.9270	0.9198	0.9136	0.9327	7
Morph. snakes (image)	0.4251	0.8769	0.8070	0.9053	0.7987	784
Morph. snakes ( $P_y$ )	0.6494	0.8812	0.8812	0.8812	0.8812	968
Graph Cut (pixel-level)	0.7143	0.9204	0.9204	0.9204	0.9204	15
Graph Cut (superpixels)	0.3164	0.8643	0.8643	0.8643	0.8643	3
RG2Sp (greedy)	0.7527	0.9577	0.9577	0.9577	0.9577	72
RG2Sp (Graph Cut)	0.7544	0.9568	0.9568	0.9568	0.9568	9

# Overview

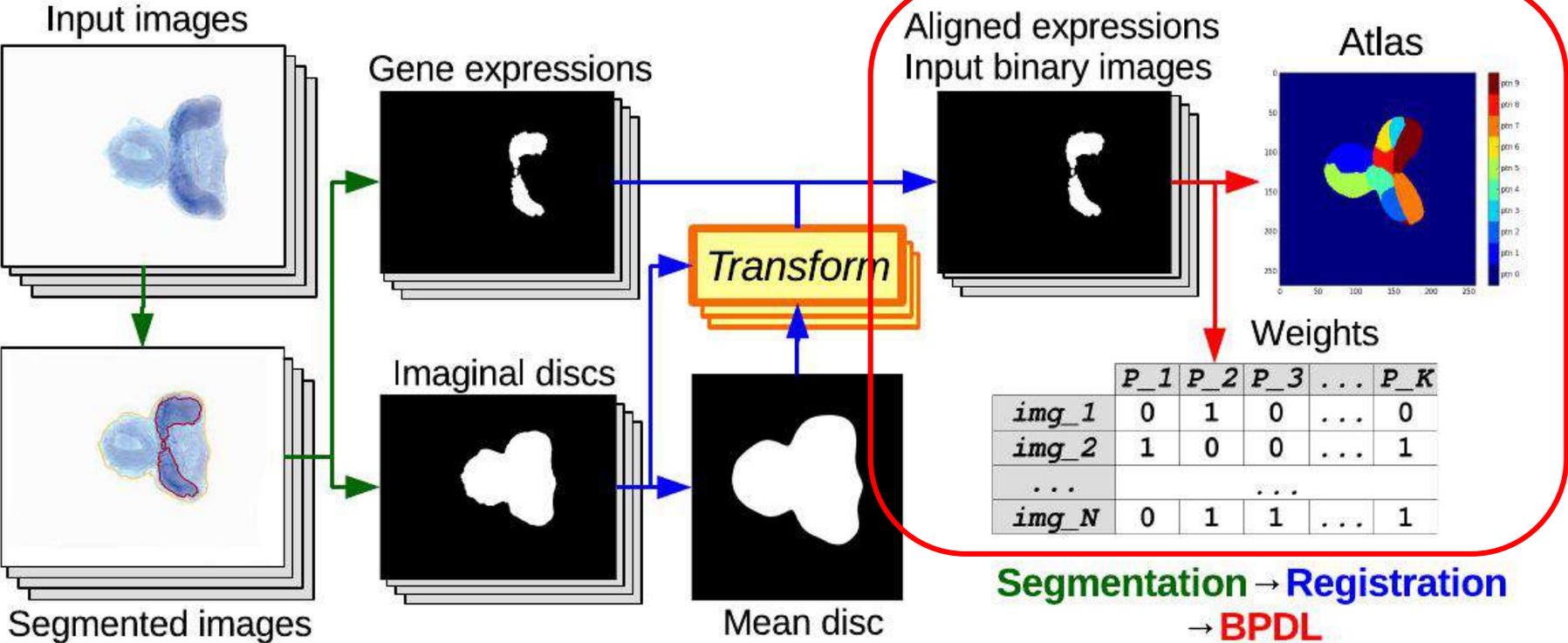
1. Problem statement
2. Used datasets
3. Instance segmentation
  - a. Structure (tissue) segmentation on superpixels
  - b. Center detection and ellipse fitting
  - c. Region growing on superpixels
- 4. Binary Pattern Dictionary Learning**
5. Conclusion & Future work

# *Binary Pattern Dictionary Learning of gene expressions*



Borovec J., Kybic J. (2016) **Binary Pattern Dictionary Learning for Gene Expression Representation in *Drosophila* Imaginal Discs**. In: Computer Vision – ACCV 2016 Workshops. Lecture Notes in Computer Science, vol 10117, Springer, [DOI: 10.1007/978-3-319-54427-4\\_40](https://doi.org/10.1007/978-3-319-54427-4_40).

# Pipeline: segmentation - registration - BPDFL



	$P_1$	$P_2$	$P_3$	...	$P_K$
<i>img_1</i>	0	1	0	...	0
<i>img_2</i>	1	0	0	...	1
...				...	
<i>img_N</i>	0	1	1	...	1

# Decomposition methods

## Formulation

$$\min_{Y,W} \|X - Y \cdot W\|^2$$

- Standard approaches
  - Non Negative Matrix Factorisation
  - Fast Independent Component Analysis
  - Sparse Principal Component Analyses
  - Dictionary Learning with Matching pursuit

# Formulation

- Image representation

$$\hat{\mathbf{g}} = \sum_{l \in \mathbb{L}} \mathbf{w}_l \cdot \llbracket \mathbf{y} = l \rrbracket$$

- Similarity measure (Hamming distance)

$$F(\mathbf{g}, \mathbf{y}, \mathbf{w}) = \sum_{i \in \Omega} \llbracket \mathbf{g}_i \neq \hat{\mathbf{g}}_i \rrbracket$$

- Regularize neighbouring pixels

$$H(\mathbf{y}) = \sum_{\substack{i, j \in \Omega, i \neq j, \\ d(i, j) = 1}} \llbracket \mathbf{y}_i \neq \mathbf{y}_j \rrbracket$$

- Optimization criterion

$$\mathbf{y}^*, \mathbf{w}^* = \arg \min_{\mathbf{y}, \mathbf{W}} \frac{1}{N} \sum_n F(\mathbf{g}^n, \mathbf{y}, \mathbf{w}^n) + \beta \cdot H(\mathbf{y})$$

# Alternating minimization

- Update weights - maximise overlap

$$w_l = \llbracket P(\mathbf{g}, \mathbf{y}, l) \geq \sigma \rrbracket \quad \text{where } \sigma = 1$$

$$\text{and } P(\mathbf{g}, \mathbf{y}, l) = \frac{\sum_{i \in \Omega, \mathbf{y}_i = l} \llbracket \mathbf{g}_i = 1 \rrbracket}{\sum_{i \in \Omega, \mathbf{y}_i = l} \llbracket \mathbf{g}_i \neq 1 \rrbracket} = \frac{\|\llbracket \mathbf{y} = l \rrbracket\|}{\sum_{i \in \Omega, \mathbf{y}_i = l} (1 - \mathbf{g}_i)} - 1$$

- Update atlas



$$\frac{1}{N} \sum_{i \in \Omega} \sum_n \underbrace{\left| \mathbf{g}_i^s - \sum_{l \in \mathbb{L}} \mathbf{w}_l^s \cdot \llbracket \mathbf{y} = l \rrbracket \right|}_{U_i(\mathbf{y}_i)} + \sum_{\substack{i, j \in \Omega, i \neq j, \\ d(i, j) = 1}} \llbracket \mathbf{y}_i \neq \mathbf{y}_j \rrbracket$$

# Algorithm

---

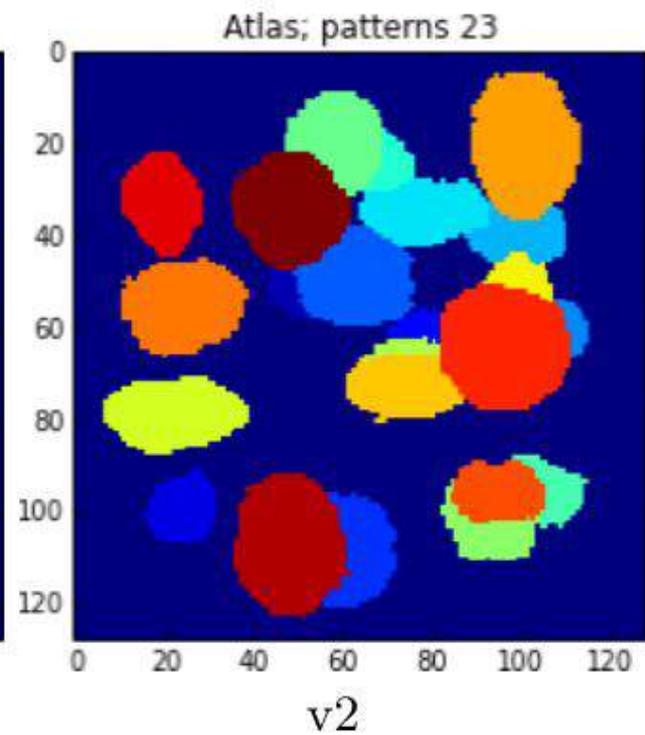
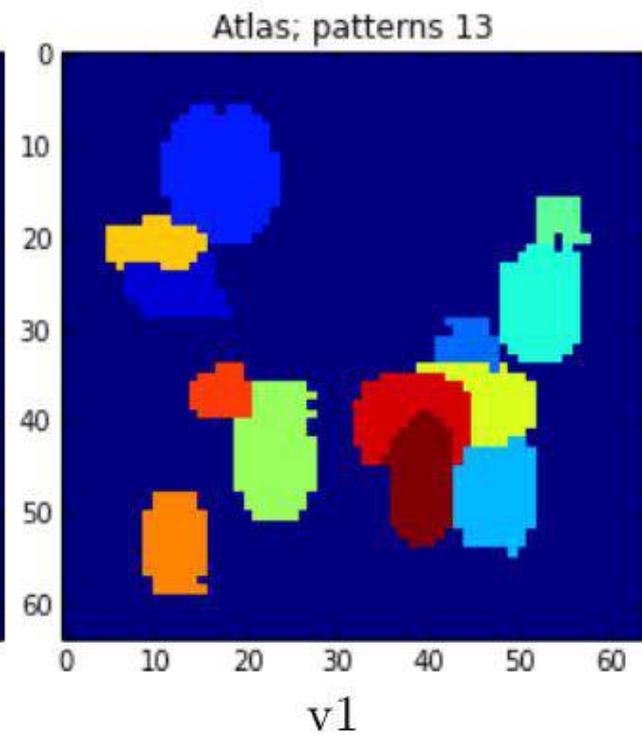
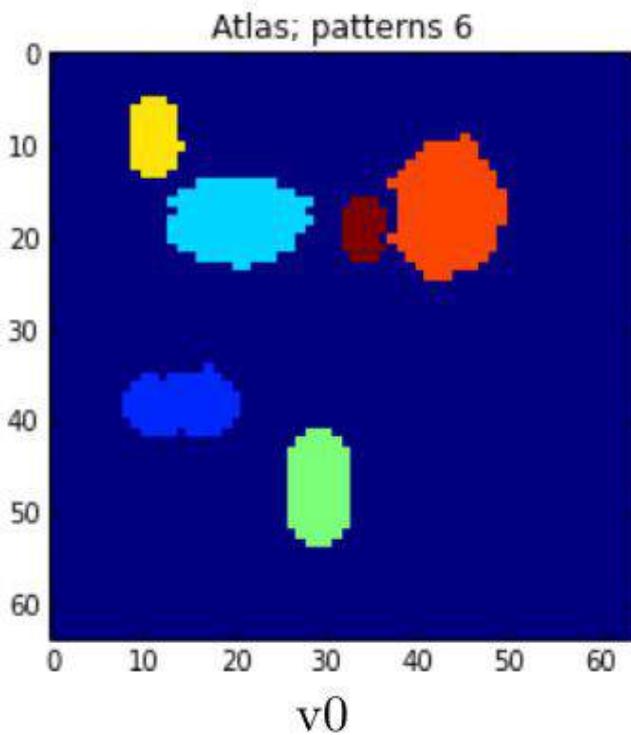
**Algorithm 1** General schema of BPDFL algorithm.

---

- 1: initialise atlas  $\mathbf{y}$
  - 2: **while** not converged **do**
  - 3:     update weights  $\mathbf{w} \in \mathbf{W}$
  - 4:     reinitialise empty patterns in  $\mathbf{y}^*$
  - 5:     update atlas  $\mathbf{y}^*$  via Graph Cut
  - 6: **end while**
- 

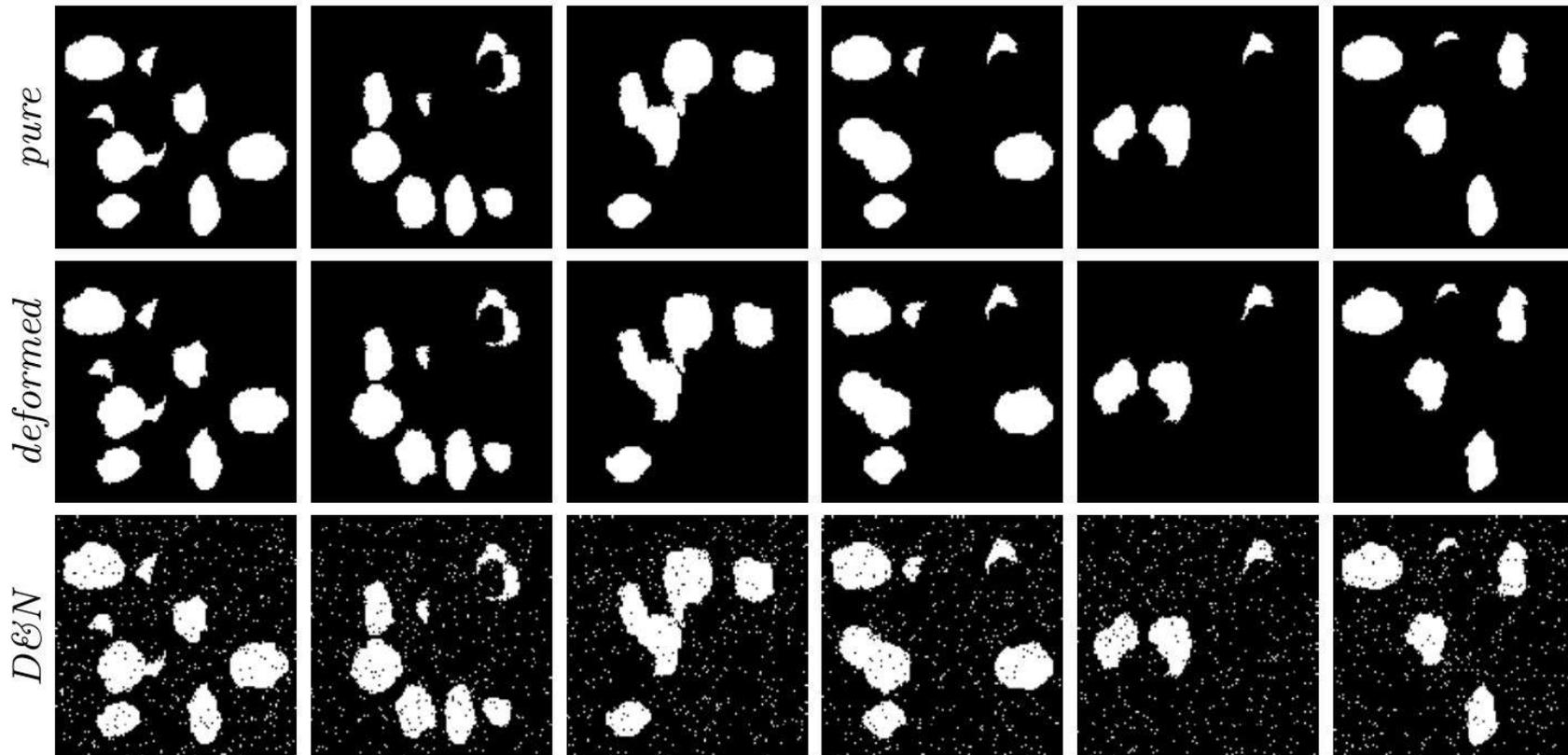
*It makes the algorithm more robust to initialisation.*

# Synthetic datasets



# Synthetic images

Simulates real images



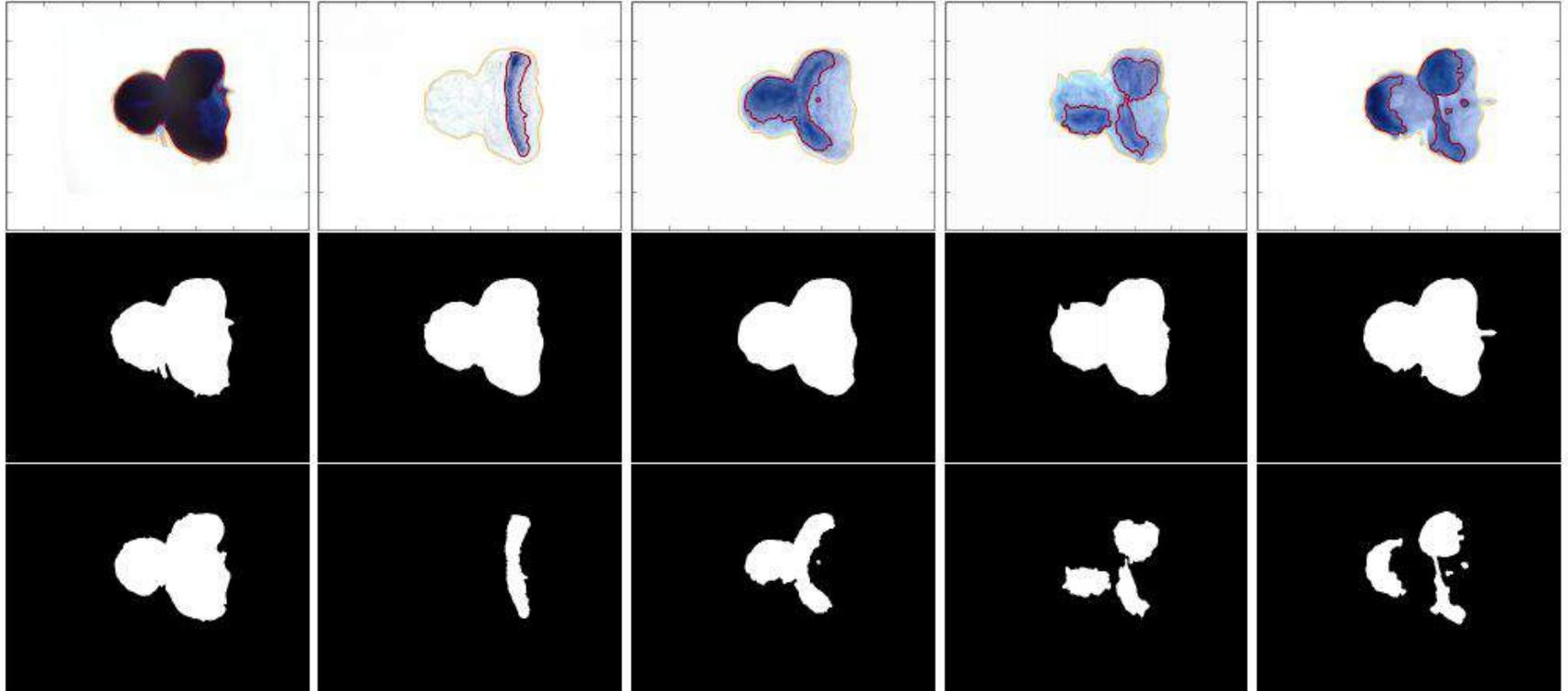
# Comparison on synth. images

datasets		NMF	FastICA	sPCA	DL	BPDL
v1		<i>(size <math>64 \times 64</math> px, 13 patterns)</i>				
<i>pure</i>	ARS	<b>1.0</b>	<b>1.0</b>	0.992	<b>0.995</b>	<b>0.999</b>
	diff.	<b>0.0</b>	<b>0.0</b>	0.0298	<b>0.019</b>	<b>0.0</b>
	time	<b>2.333</b>	340.32	18.291	737.47	<b>6.029</b>
<i>deform</i>	ARS	0.785	<b>0.948</b>	0.780	0.779	<b>0.992</b>
	diff.	0.017	<b>0.004</b>	0.029	0.033	<b>0.005</b>
	time [s]	<b>4.001</b>	312.18	15.000	700.03	<b>7.561</b>
<i>D&amp;N</i>	ARS	0.091	<b>0.878</b>	0.009	0.0727	<b>0.951</b>
	diff.	0.048	<b>0.010</b>	0.061	0.0499	<b>0.003</b>
	time [s]	<b>4.490</b>	439.04	11.420	697.599	<b>9.562</b>

# Comparison on synth. images

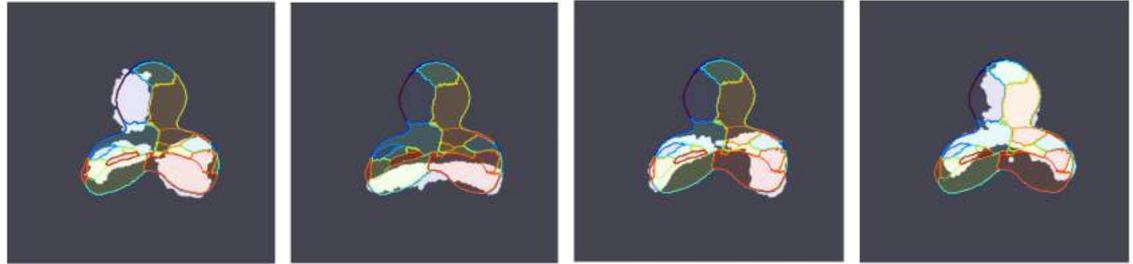
datasets		NMF	FastICA	sPCA	DL	BPDL
<b>v2</b>		<i>(size <math>128 \times 128</math> px, 23 patterns)</i>				
<i>pure</i>	ARS	<b>1.0</b>	<b>1.0</b>	<b>0.989</b>	<b>1.0</b>	<b>0.999</b>
	diff.	<b>0.0</b>	<b>0.0</b>	0.037	<b>0.0</b>	<b>0.005</b>
	time [s]	<b>82.329</b>	5533.4	460.82	14786.	<b>88.260</b>
<i>deform</i>	ARS	0.818	<b>0.846</b>	0.801	0.807	<b>0.970</b>
	diff.	0.019	<b>0.015</b>	0.056	0.046	<b>0.004</b>
	time [s]	<b>144.10</b>	5683.2	477.47	13619.	<b>165.22</b>
<i>D&amp;N</i>	ARS	0.120	<b>0.612</b>	0.024	0.144	<b>0.877</b>
	diff.	<b>0.036</b>	<b>0.036</b>	0.092	0.039	<b>0.013</b>
	time [s]	<b>77.399</b>	6912.9	485.44	13729.	<b>289.51</b>

# Input segmented imaginal discs

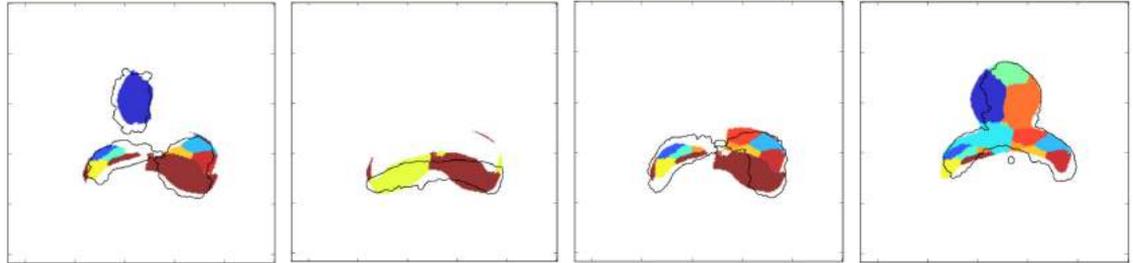


# Visualise results on imaginal discs

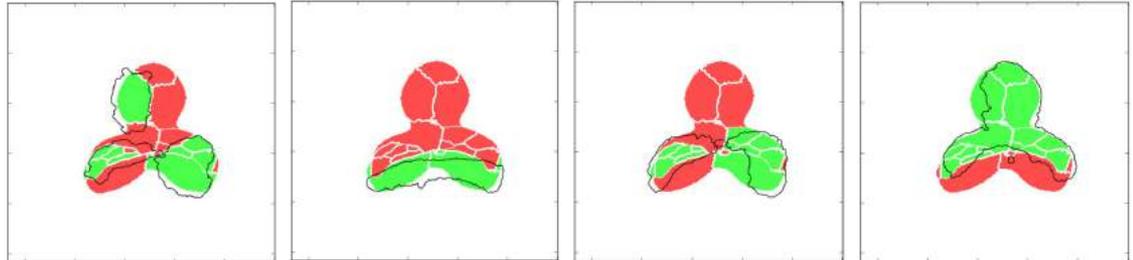
Gene & atlas



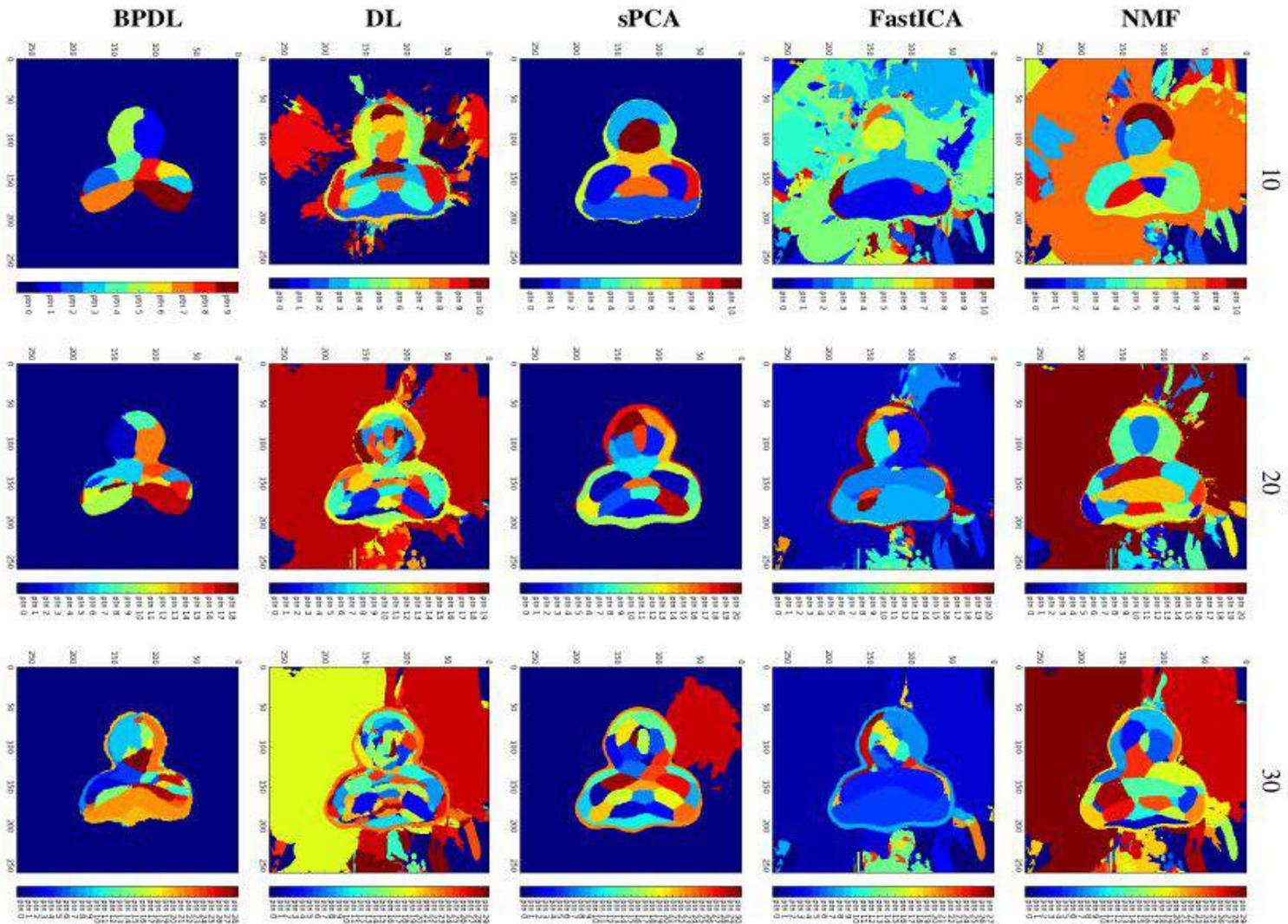
Used patterns



Pattern activation



# Extracted Atlases for Imaginal discs



10

Number of patterns  
20

30

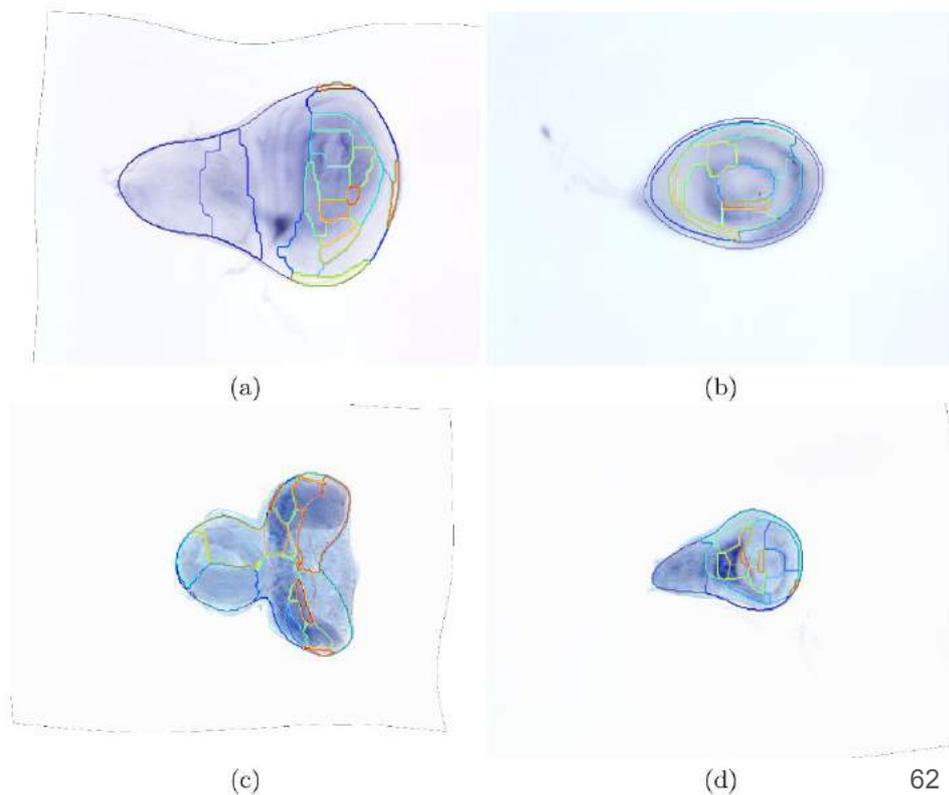
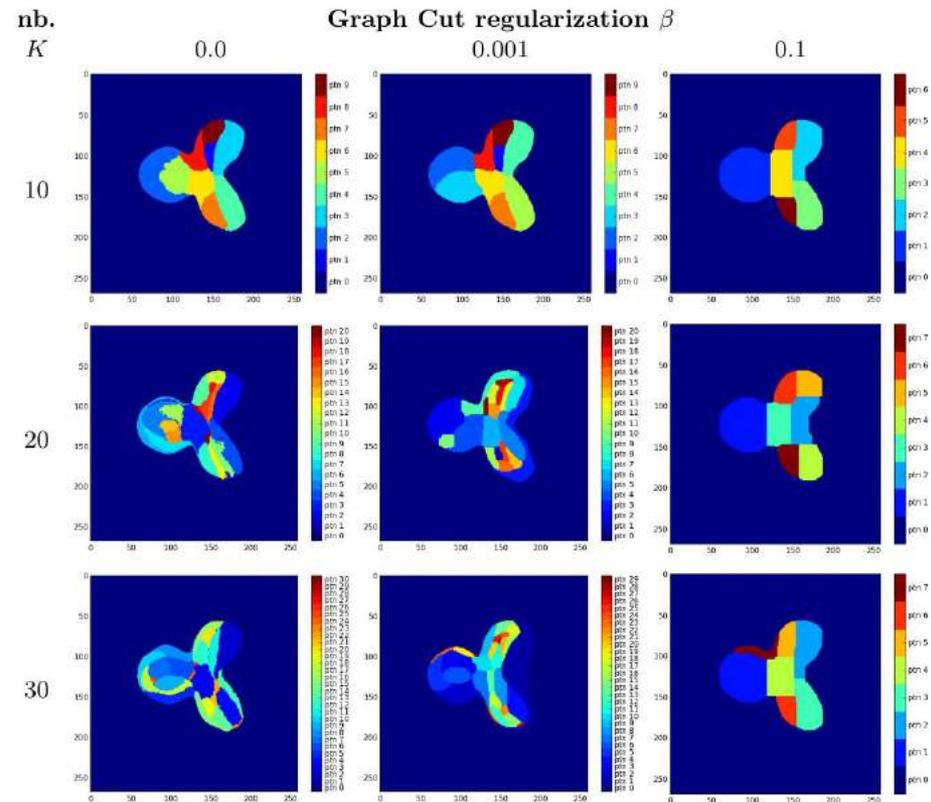
# Results on imaginal discs

No ground truth atlases...

Method	Number of patterns $K$			Time [min]
	10	20	30	
NMF	0.0939	0.0823	0.0723	<b>10</b>
FastICA	0.1197	0.0779	0.0485	24
sPCA	<b>0.0476</b>	<b>0.0413</b>	<b>0.0352</b>	477
DL	0.0939	0.0648	0.0596	338
BPDL	<b>0.0467</b>	<b>0.0395</b>	<b>0.0361</b>	<b>20</b>

# Parameter selection

# & More discs



# Overview

1. Problem statement
2. Used datasets
3. Instance segmentation
  - a. Structure (tissue) segmentation on superpixels
  - b. Center detection and ellipse fitting
  - c. Region growing on superpixels
4. Binary Pattern Dictionary Learning
5. Conclusion & Future work

# Conclusion

- Presented four image processing methods:
  - Image segmentation on superpixels
  - Center detection on segmented images
  - Region growing with shape prior
  - Binary Pattern Dictionary Learning
  - **Implementation:**
    - <http://borda.github.com/pyImSegm>
    - <http://borda.github.com/pyBPDL>
- Future work
  - Complete image analysis pipeline in 2.5D
  - Try instance segmentation with CNNs
  - ...

The screenshot shows the GitHub repository for 'pyImSegm'. At the top, it says 'pyImSegm' and 'Image segmentation - general superpixel segmentation & center detection & region growing'. Below that is a yellow bar with 'View On GitHub'. The main content area is titled 'Image segmentation toolbox' and includes a progress bar with 'BUILD PASSED', 'CODEcov 97%', 'CODACY A', 'TODOLIST 100', 'SHAPEFORM', 'COVINTEST 93%', and 'PASSED'. The main heading is 'Superpixel segmentation with GraphCut regularisation'. The text describes the toolbox as an initial phase of many image processing tasks, mentioning features like superpixels, descriptors, classification, and segmentation using Graph Cut.

The screenshot shows the GitHub repository for 'pyBPDL'. At the top, it says 'pyBPDL' and 'Binary Pattern Dictionary Learning for gene activation in microscopy images'. Below that is a dark blue bar with 'View pyBPDL on GitHub', 'Download', and 'Universal'. The main content area is titled 'Binary Pattern Dictionary Learning' and includes a progress bar with 'BUILD PASSED', 'CODEcov 100%', 'CODACY A', 'TODOLIST 100', 'SHAPEFORM', 'COVINTEST 93%', and 'PASSED'. The text describes the pipeline for gene activation analysis, mentioning input images, gene expressions, aligned expressions, and atlases. A diagram illustrates the process: 'Input images' and 'Segmented images' lead to 'Gene expressions' and 'Integral discs'. 'Gene expressions' and 'Integral discs' are processed by a 'Transform' step to produce 'Aligned expressions' and 'Mean disc'. 'Aligned expressions' and 'Mean disc' are used for 'Segmentation - Registration - BPDL' to produce an 'Atlas'. A table shows the atlas data: 'img\_1' [0, 1, 0, 0, 0], 'img\_2' [0, 1, 0, 0, 0], 'img\_3' [0, 1, 0, 0, 0]. Below the diagram, it says 'For the image segmentation and individual object detection we used image segmentation toolbox.' and 'Comparable (SoA) Methods'. At the bottom, it says 'We have our method BPDL and also we compare it to state-of-the-art, see [Figure 1 \(left\)](#)'.

