Using image data for statistical analysis and modeling

Wolfgang Huber &

Gregoire Pau

EMBL

EMBL European Molecular Biology Laboratory

EBImage

Fast and user-friendly image processing toolbox for R

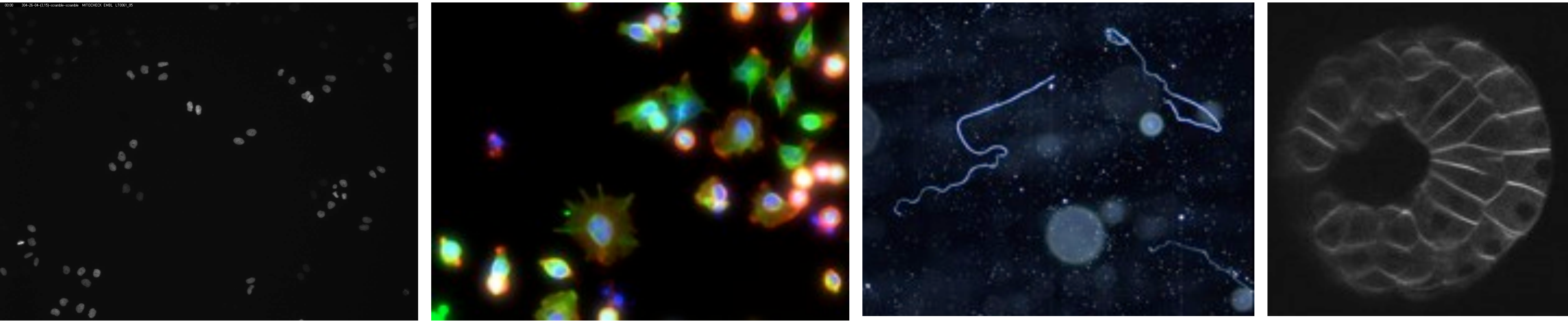
Provides functionality for

- Reading/writing/displaying images
- Image processing (pixel arithmetic, filtering, geometric transformations)
- Object segmentation

Goals

Process multidimensional images

Extract quantitative descriptors from (microscopy) images



EBImage

What it is and isn't:

the package offers basic infrastructure for working with images in RAM as matrices.

Incomplete w.r.t. sophisticated algorithms (we aim to make it easier to call out to ImageJ plugins)

Does not support huge images (but we're planning to better integrate working with images in netcdf)

Your contributions are welcome:

- contribute a function + man page
- write your own package on top of `EBImage`

Image representation

Multidimensional array of intensity values

Seamless integration with R's native arrays

21	20	21	28	43	53	67	54
12	31	30	41	52	71	98	78
11	14	33	49	72	110	133	144
12	19	29	39	57	74	121	100
16	21	28	31	59	74	98	74
18	23	27	38	50	61	62	49
17	19	24	39	42	48	47	52
16	15	23	37	41	38	36	41

Lena: 512x512 matrix

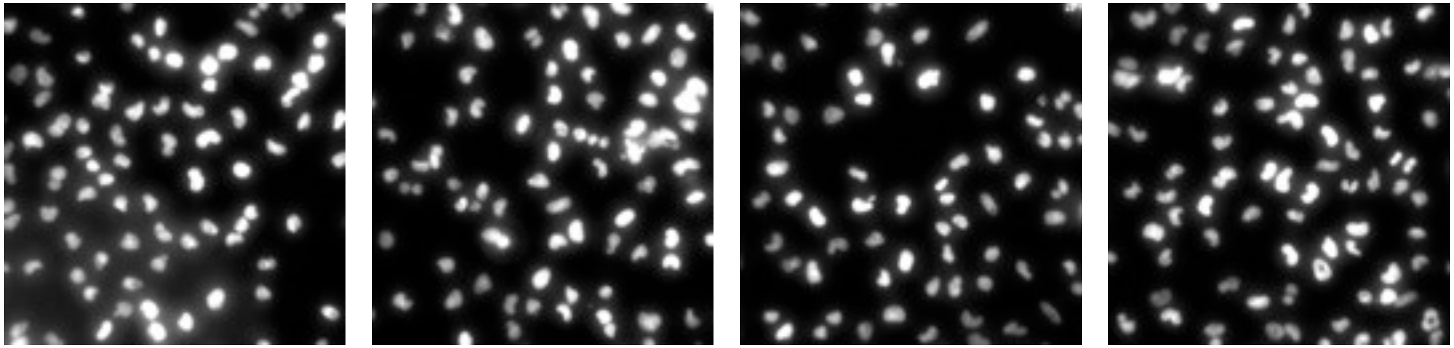
Image representation

Multidimensional array

2 first dimensions: spatial dimensions

Other dimensions: replicate, color, time point, condition, z-slice…

Nuclei
4 replicates



r0 r1 r2 r3

Lena
3 color
channels

R G B

Image rendering

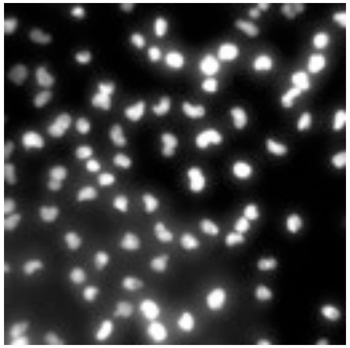
Rendering dissociated from representation

2 rendering modes

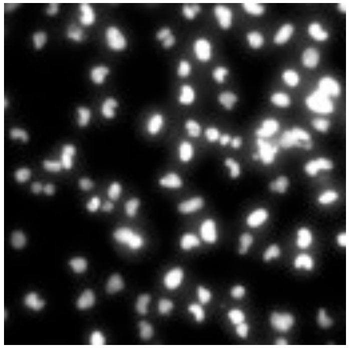
as sequence of
grayscale images

as color
image

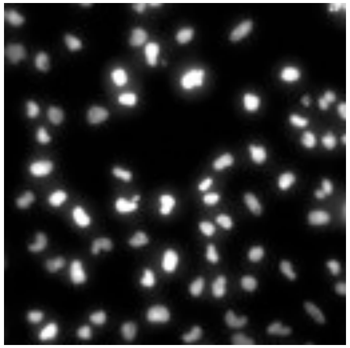
Nuclei
4 replicates



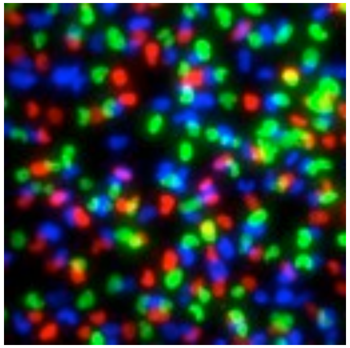
r0



r1



r2



Lena
3 color
channels

R

G

B

IO

Functions readImage(), writeImage()

Reads an image, returns an array

Supports more than 80 formats (JPEG, TIFF, PNG, GIF, ...)

Supports HTTP, sequences of images

Example: format conversion

```
library('EBImage')
x = readImage('sample-001-02a.tif')
writeImage(x, 'sample-001-02a.jpeg', quality=95)
```

Display

Function display()

GTK+ interactive: zoom, scroll, animate

Supports RGB color channels and sequence of images

```
x = readImage('lena.png')
display(x)
```

Frame: 3/6 Image: 512x512x6 Zoom: 200% Position: (237,206)

Pixel arithmetic

Seamless integration with R's native arrays

E.g. adjust brightness, contrast and gamma-factor

`x`

`x+0.5`

`3*x`

`(x+0.2)^3`

Spatial transformations

Cropping, thresholding, resizing, rotation

`x[45:90, 120:165]`

`resize(x, w=128)`

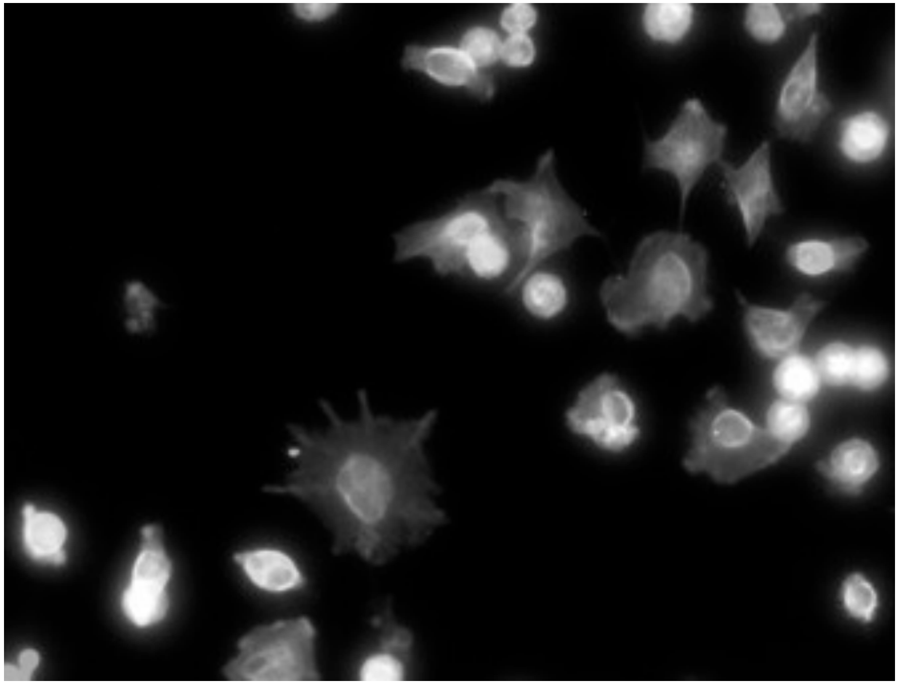
`x>0.5`

`rotate(x, angle=30)`

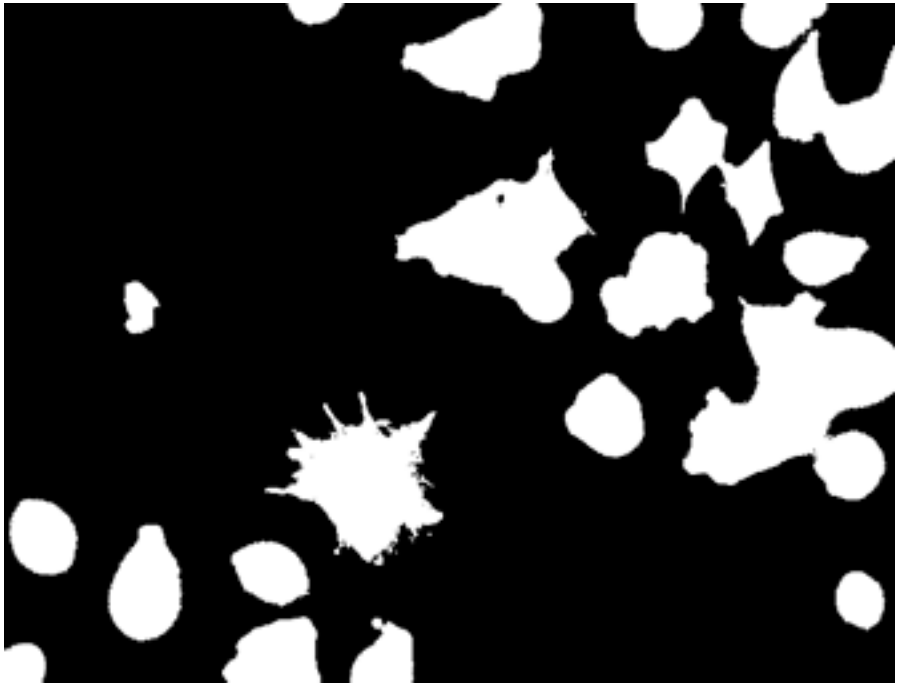
Thresholding

Global thresholding

Primitive building block for object segmentation



`x`

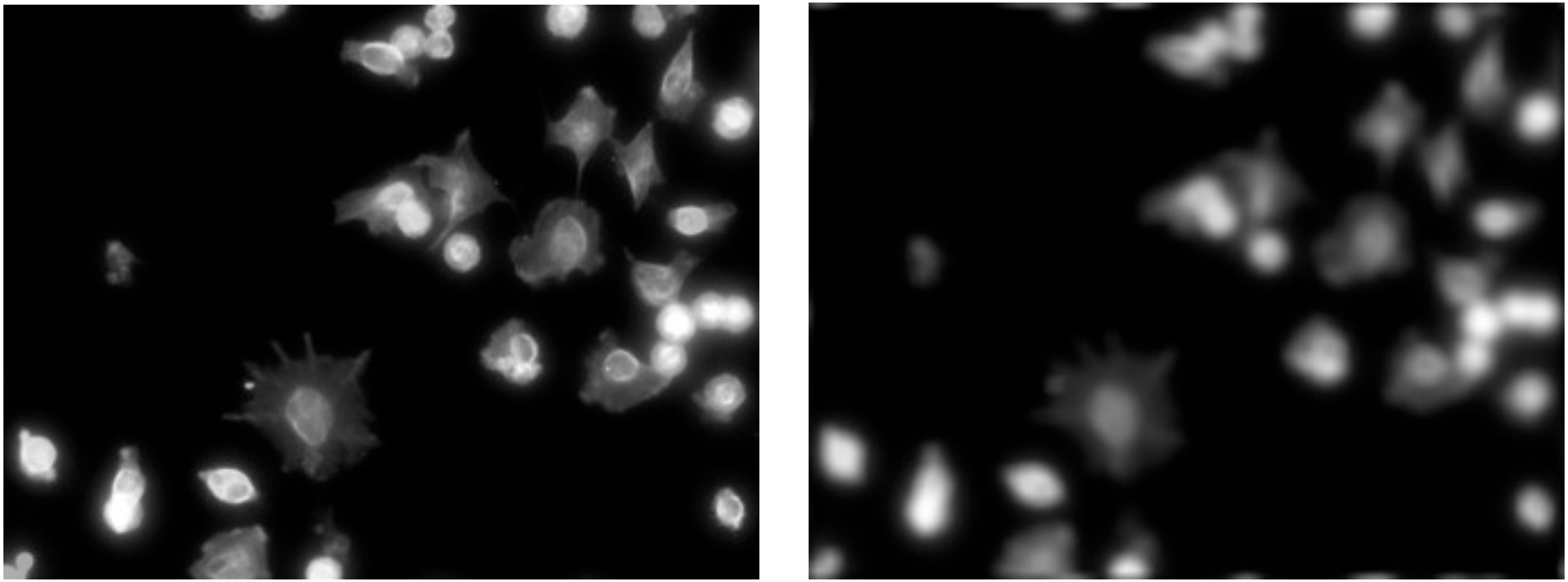


`x>0.3`

Linear filter

Fast 2D convolution with filter2()

Low-pass filter: smooth images, remove artefacts



x

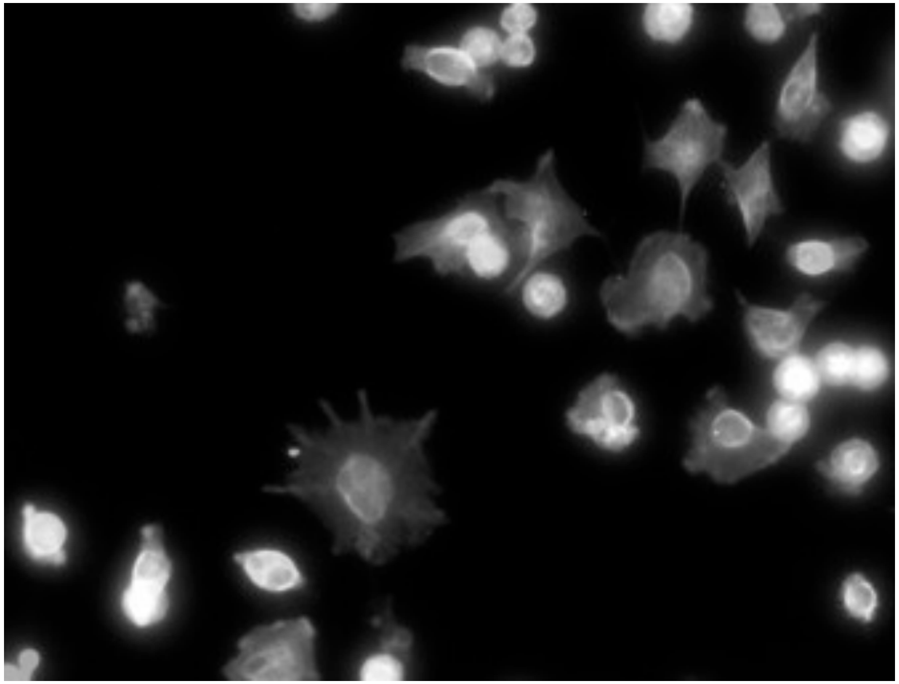
```
f = array(1, dim=c(9, 9))
f = f/sum(f)
y = filter2(x, f)
```

$$x * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Linear filter

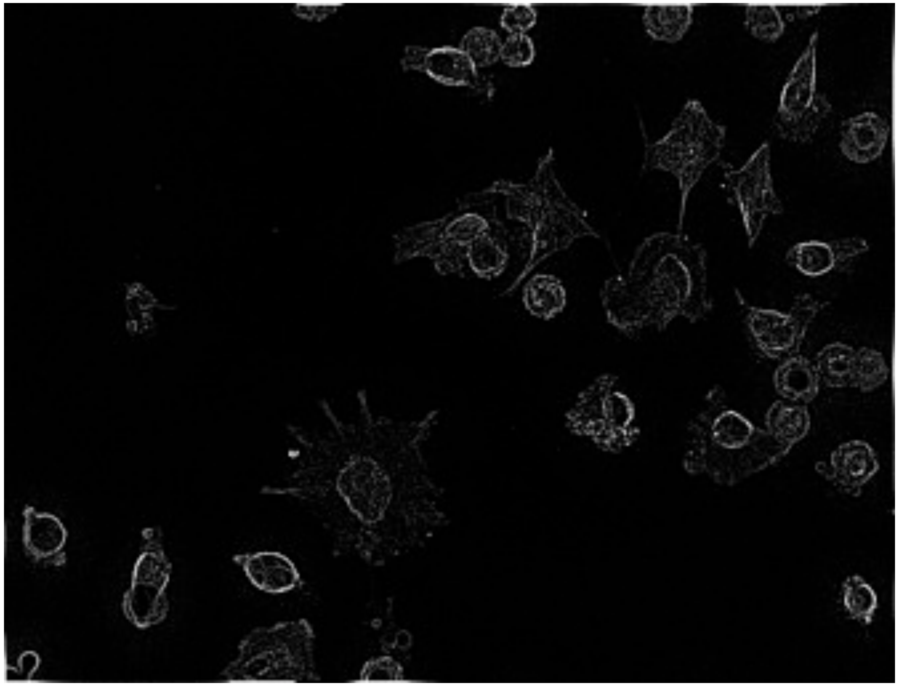
Fast 2D convolution with filter2()

High-pass filter: detect cell edges



x

```
f = array(1, dim=c(9, 9))
f[3, 3] = -8
y = filter2(x, f)
```



$$x * \begin{bmatrix} 1 & -1 & 1 \\ 1 & -8 & 1 \\ 1 & -1 & 1 \end{bmatrix}$$

Example: segmentation of nuclei

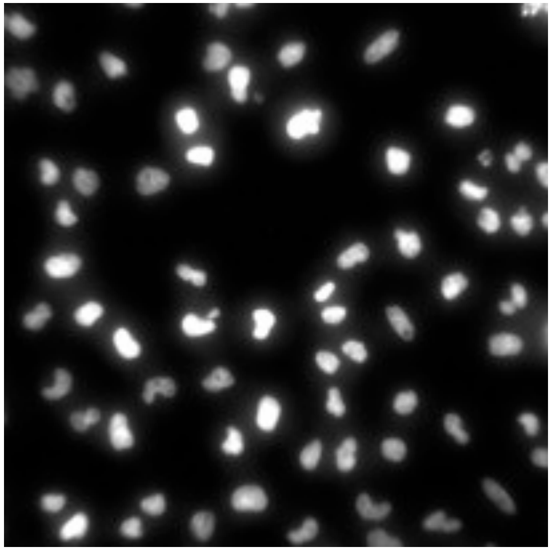
Global thesholding + labeling

Function `bwlabel`

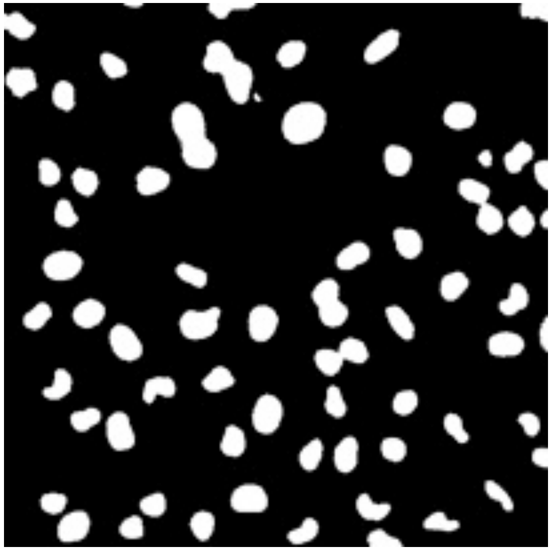
Labels connected sets (objects) in a binary image

The pixels of each connected object are set to a unique integer value

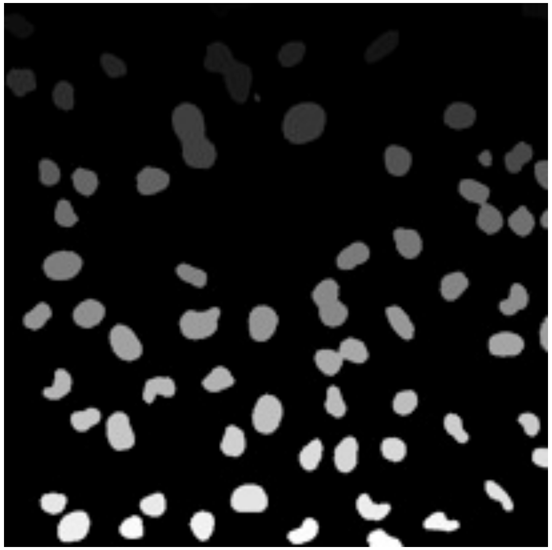
`max(bwlabel(x))` **gives the number of objects,**
`table(bwlabel(x))` **their sizes**



`x`



`x>0.2`

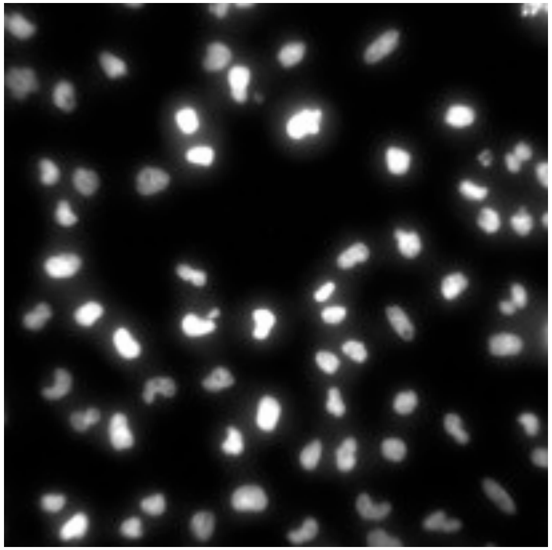


`bwlabel(x>0.2)`

Better segmentation

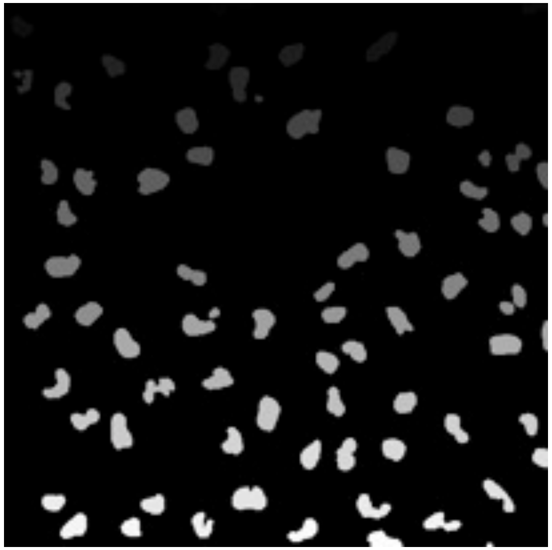
Adaptive thresholding + mathematical opening + holes filling

```
xb =  thresh(x[,,1], 10, 10, 0.05)

kern = makeBrush(5, shape='disc')

xb = dilate(erode(xb, kern), kern)
```



x

xb



xl

Nucleus morphology quantification

Function `getFeatures`

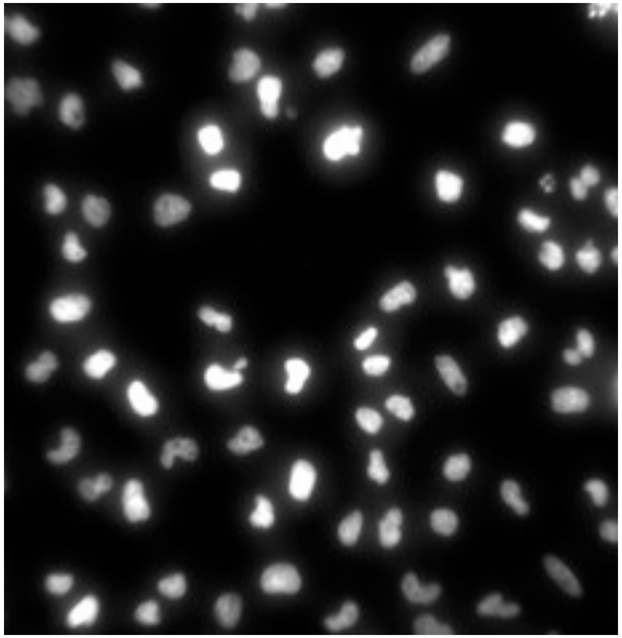
Extracts object features: geometric, image moment based (Zernike moments), texture based (Haralick)

Direct interpretation (e.g.: DNA content) or for classification/ clustering

41 features

76 nuclei

```
          g.x        g.y  g.s g.p     g.pdm    g.pdsd    g.effr   g.acirc
 [1,] 123.1391   3.288660  194  67  9.241719  4.165079  7.858252 0.417525
 [2,] 206.7460   9.442248  961 153 20.513190  7.755419 17.489877 0.291363
 [3,] 502.9589   7.616438  219  60  8.286918  1.954156  8.349243 0.155251
 [4,]  20.1919  22.358418 1568 157 22.219461  3.139197 22.340768 0.116709
 [5,] 344.7959  45.501992 2259 233 35.158966 15.285795 26.815332 0.501106
 [6,] 188.2611  50.451863 2711 249 28.732680  6.560911 29.375808 0.168941
 [7,] 269.7996  46.404036 2131 180 26.419631  5.529232 26.044546 0.193805
 [8,] 106.6127  58.364243 1348 143 21.662879  6.555683 20.714288 0.264836
 [9,] 218.5582  77.299007 1913 215 25.724580  6.706719 24.676442 0.243073
[10,]  19.1766  81.840147 1908 209 26.303760  7.864686 24.644173 0.304507
[11,]   6.3558  62.017647  340  68 10.314127  2.397136 10.403142 0.188235
[12,]  58.9873  86.034128 2139 214 27.463158  6.525559 26.093387 0.207106
[13,] 245.1087  94.387405 1048 123 18.280901  2.894758 18.264412 0.112595
[14,] 411.2741 109.198678 2572 225 28.660816  7.914664 28.612812 0.224727
[15,] 167.8151 107.966014 1942 160 24.671533  2.534342 24.862779 0.084963
[16,] 281.7084 121.609892 2871 209 31.577270  6.470767 30.230245 0.128874
[17,] 479.2334 143.098241 1649 183 23.913630  6.116630 22.910543 0.248635
[18,] 186.5930 146.693122 2079 199 27.280908  6.757808 25.724818 0.195286
[19,] 356.7303 148.253418 3145 285 34.746206 11.297632 31.639921 0.313513
[20,] 449.2436 147.798319  119  37  5.873578  1.563250  6.154582 0.243697
...
```



Installation

`EBImage` requires the following softwares/libraries to be installed:

- gtk+-2.0
- ImageMagick
- pkg-config (on Mac OS X and Linux)

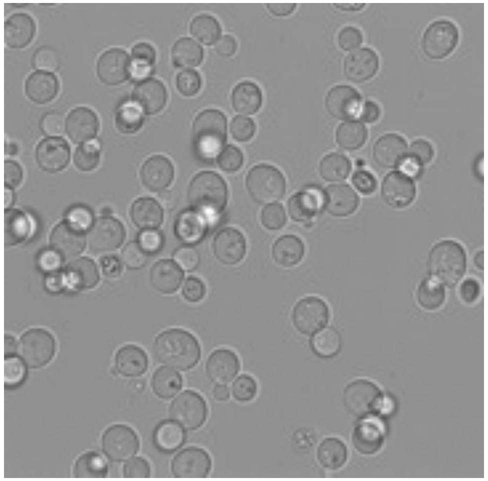
This has been cumbersome for some users … we are planning to remove these dependencies in exchange for one on Qt (which we would share with several other R packages)

Summary: EBImage

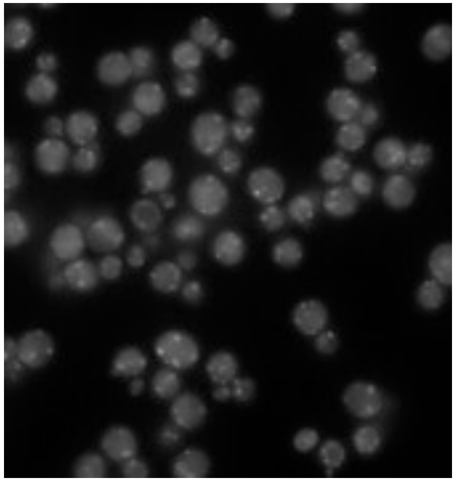
Powerful and fast package to compute with images in R

Aim: Automation of basic tasks such as image transformation, segmentation (object identification) and quantitative feature extraction

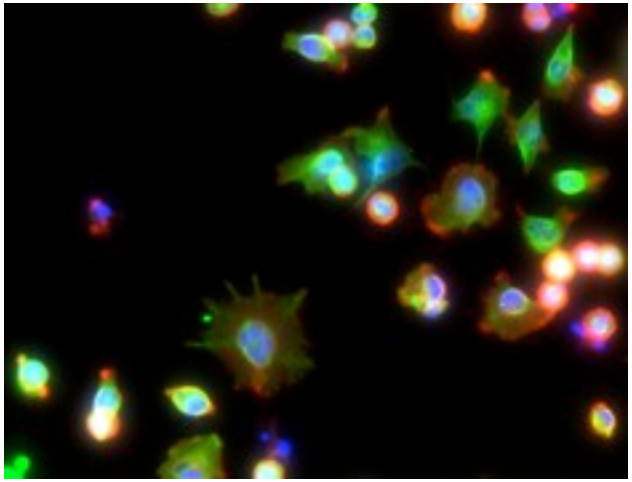
Design: manipulate images just like arrays in R (and in fact, use much of the implementation)



Yeast, brightfield



Yeast, GFP-tagged protein



HeLa, Höchst +Actin+Tubulin

An example application workflow

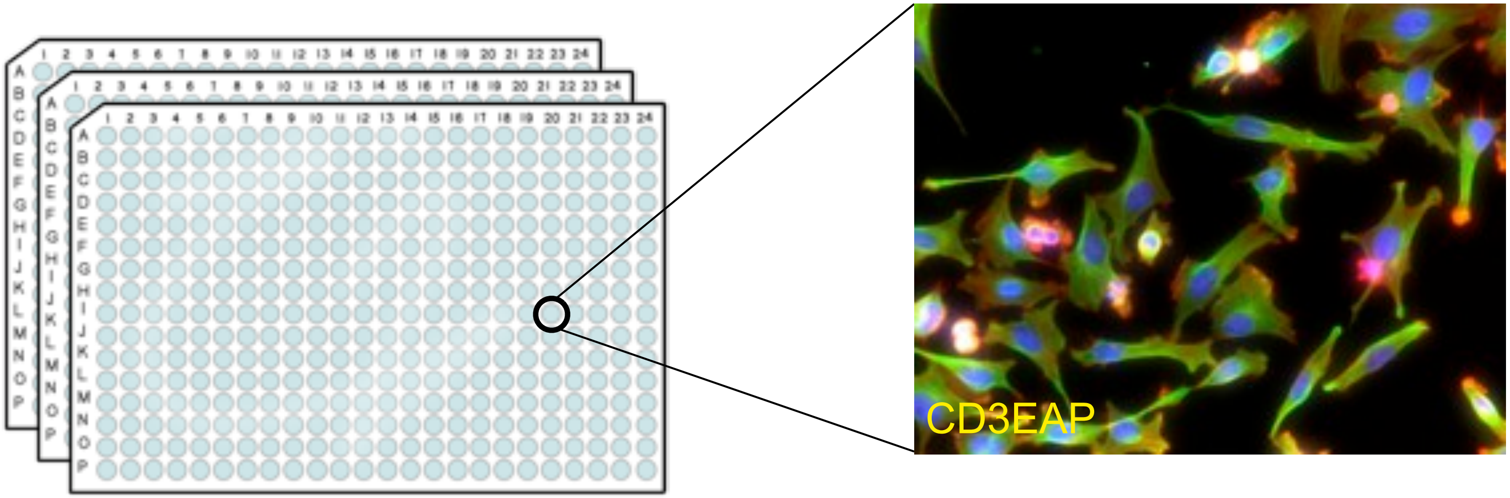
RNAi induced cell morphology phenotypes in human cells

with F. Fuchs, C. Budjan, Michael Boutros (DKFZ)

Genomewide RNAi library (Dharmacon, 22k siRNA-pools)

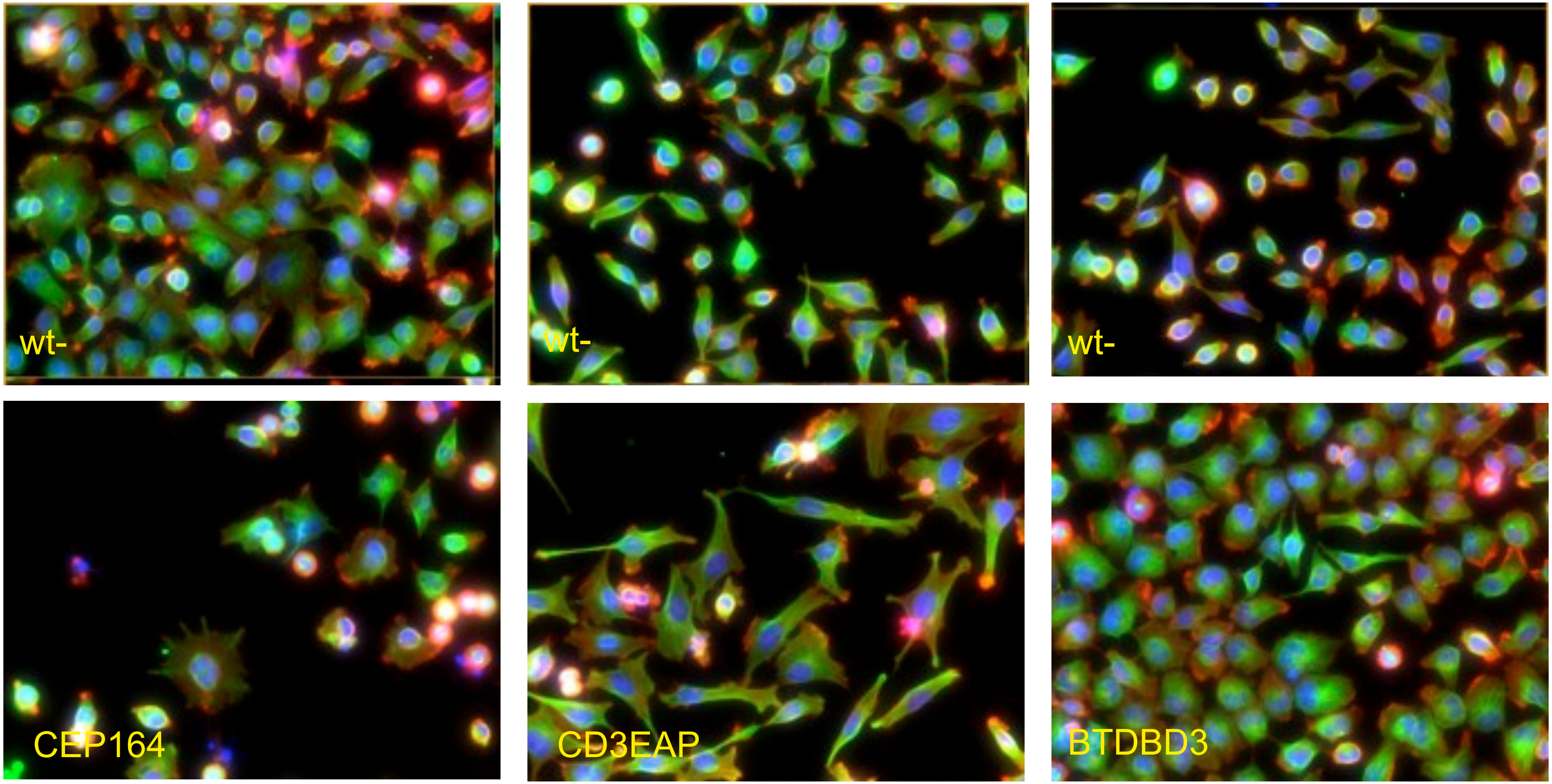
HeLa cells, incubated 48h, then fixed and stained

Microscopy readout: DNA (DAPI), tubulin (Alexa), actin (TRITC)



Molecular Systems Biology 6:370 (2010)
www.cellmorph.org

siRNA perturbation phenotypes observed by automated microscopy



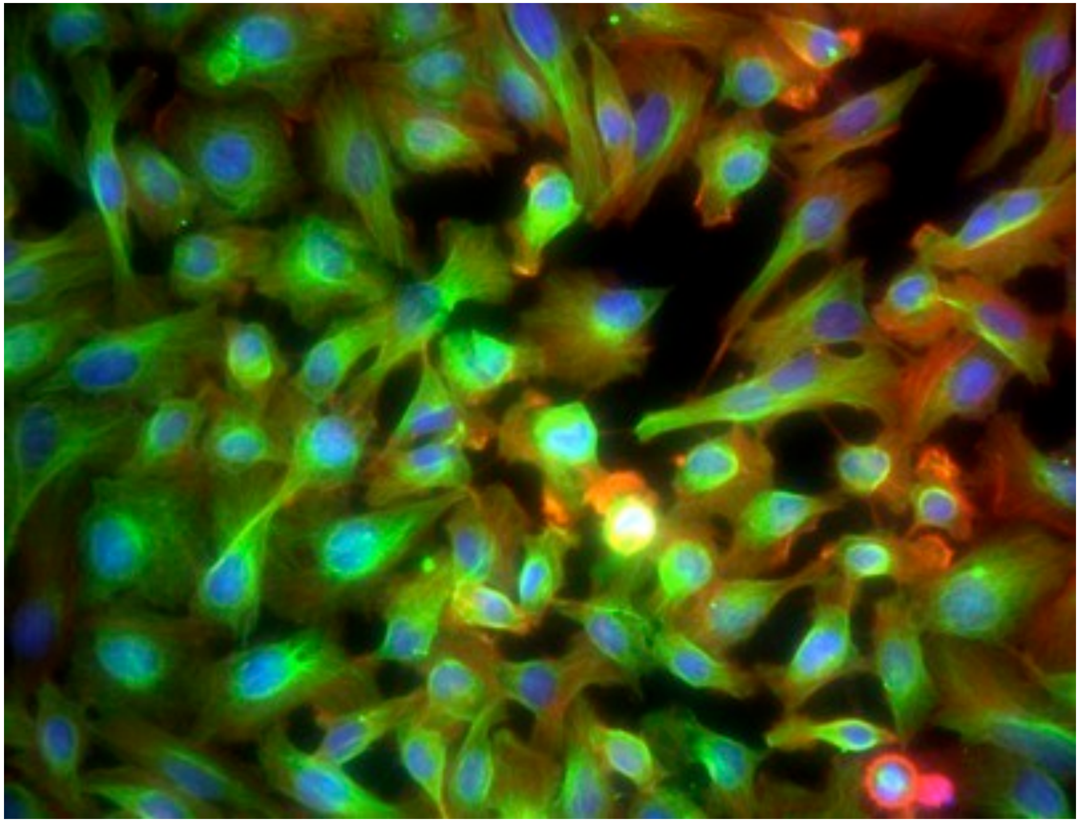
22839 wells, 4 images per well
each with DNA, tubulin, actin, 1344 x 1024 pixel at 12 bit

Identifying (segmenting) the nuclei

Nuclei are extracted from the DNA channel H

Adaptive local thresholding: Nmask = $(H * w) > \sigma H$

Connected set labelling + morphological opening



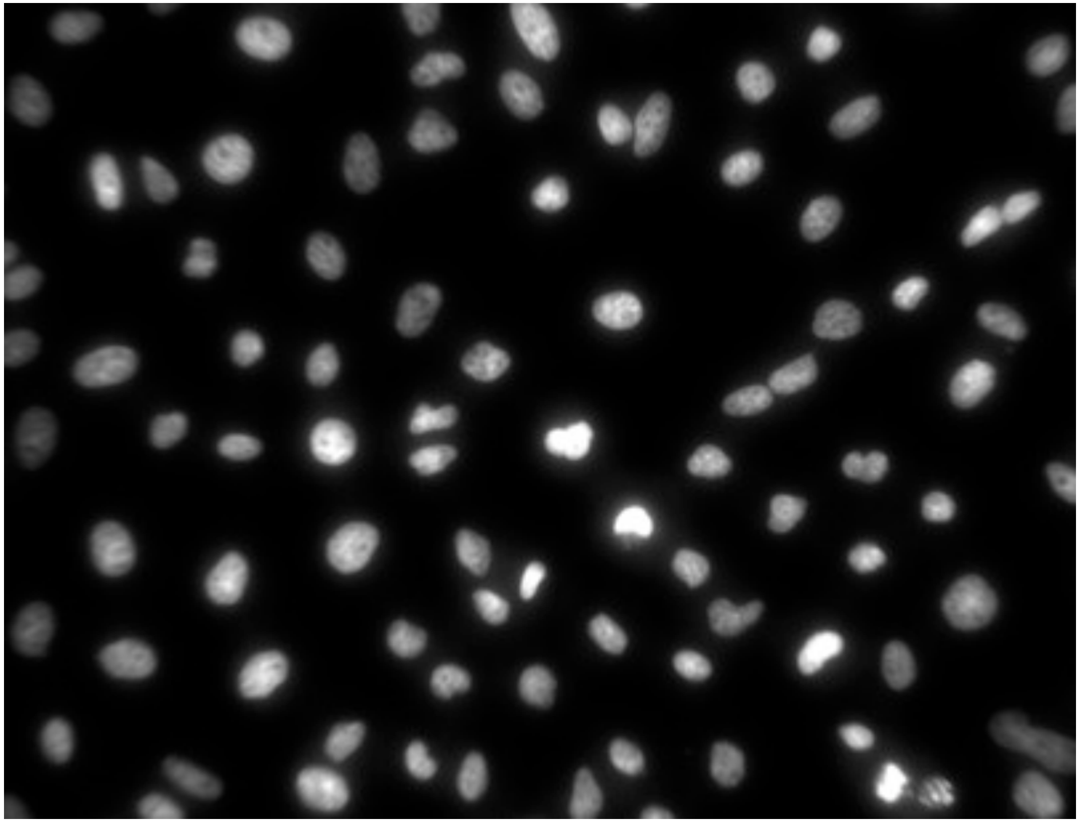
PLK3

Identifying (segmenting) the nuclei

Nuclei are extracted from the DNA channel H

Adaptive local thresholding: Nmask = $(H * w) > \sigma H$

Connected set labelling + morphological opening



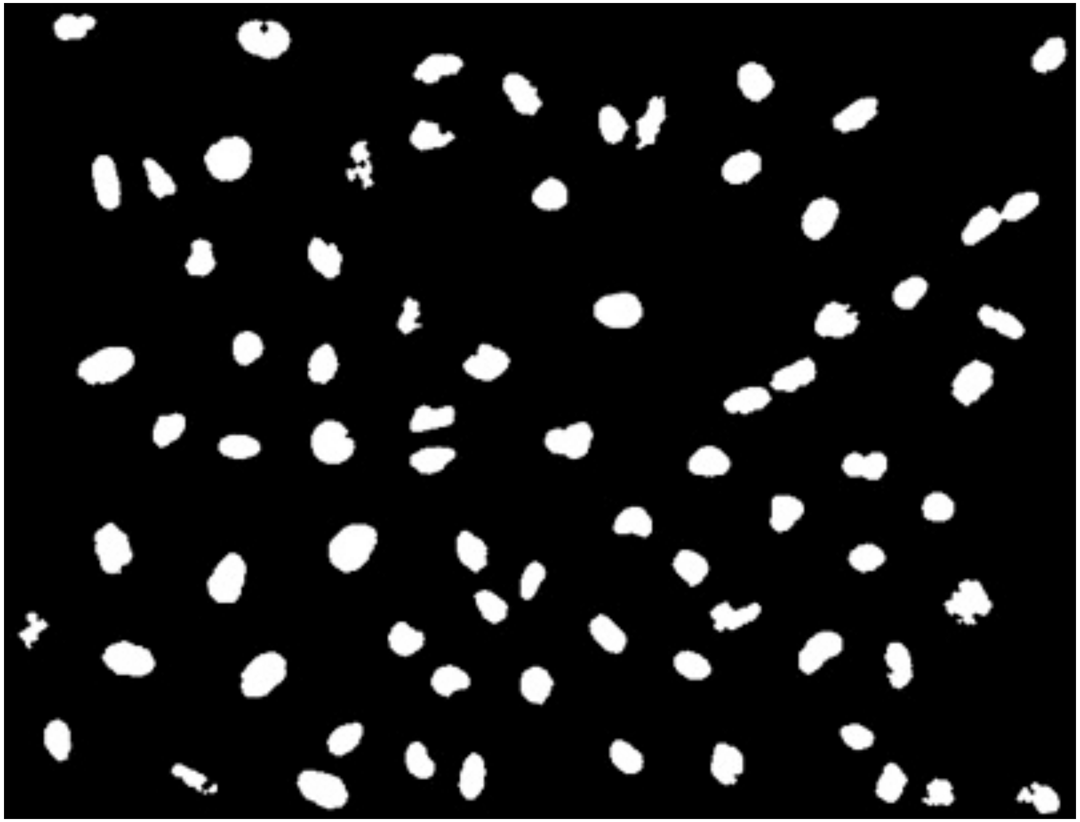
PLK3

Identifying (segmenting) the nuclei

Nuclei are extracted from the DNA channel H

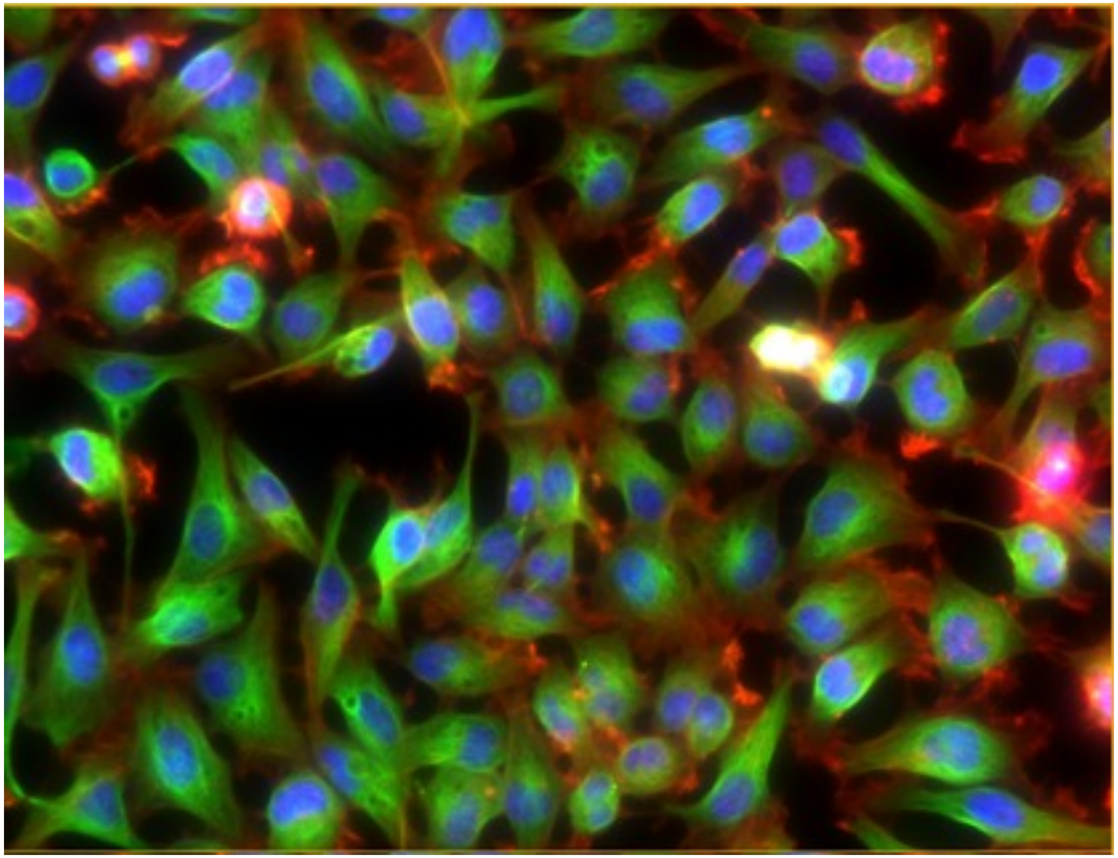
Adaptive local thresholding: Nmask = (H * w) > σH

Connected set labelling + morphological opening



PLK3

Segmentation of cells

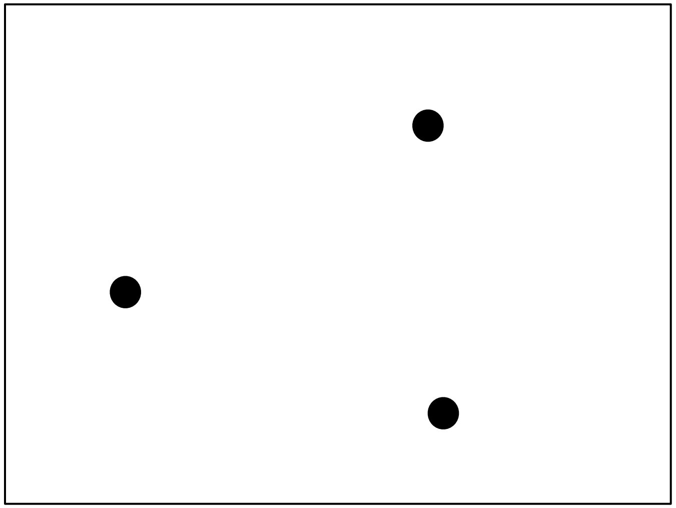


Nuclei are relatively easy.

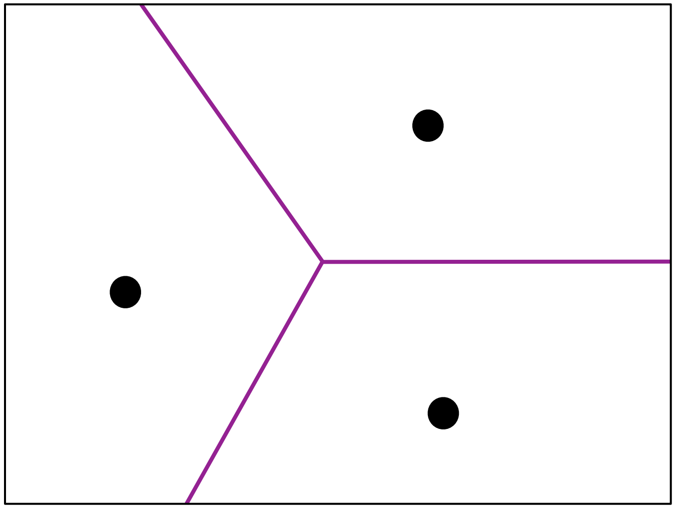
But cells touch.

How do you draw reasonable boundaries between cells' cytoplasmata?

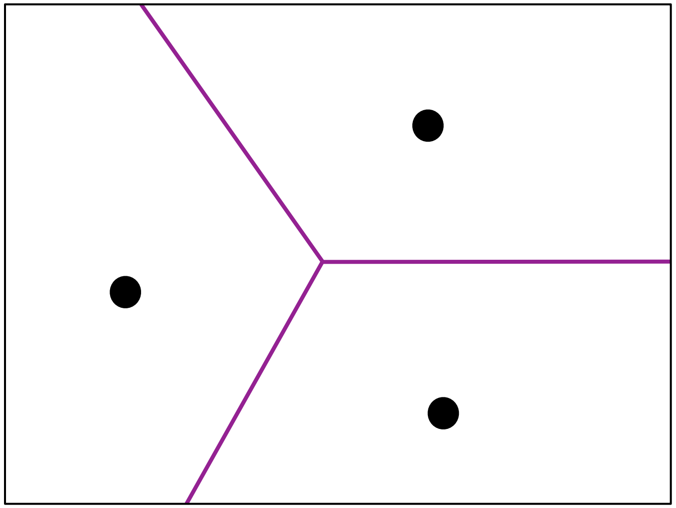
Voronoi segmentation

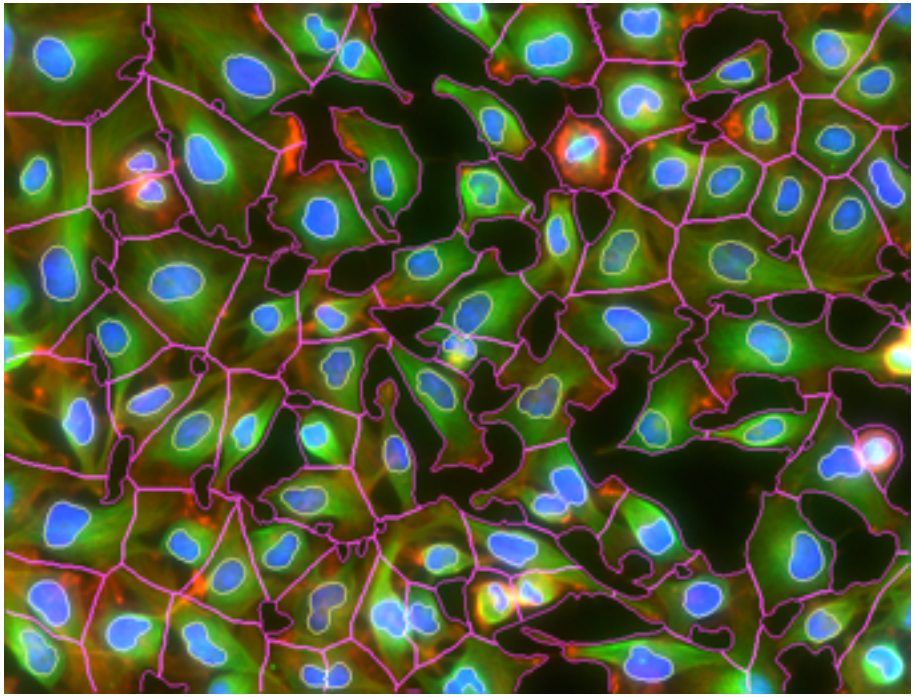


Voronoi segmentation

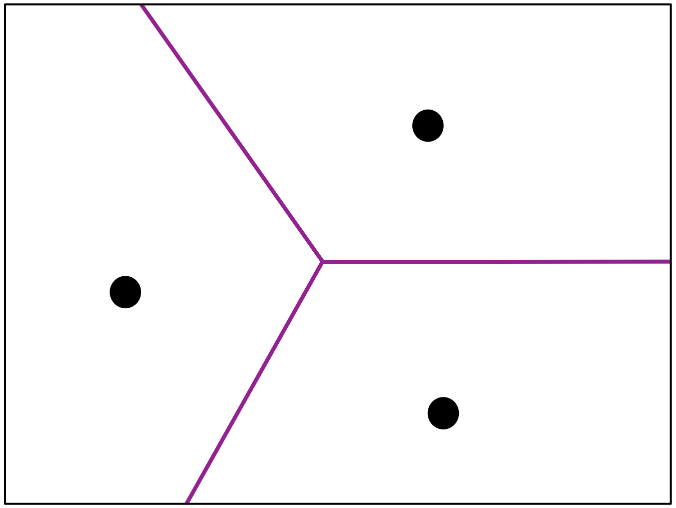


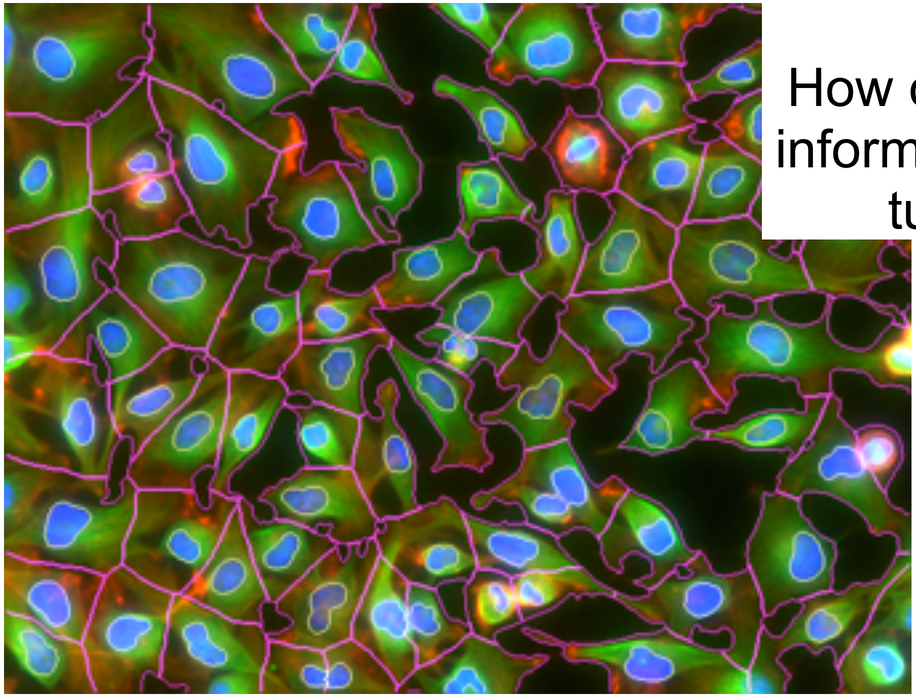
Voronoi segmentation





Voronoi segmentation



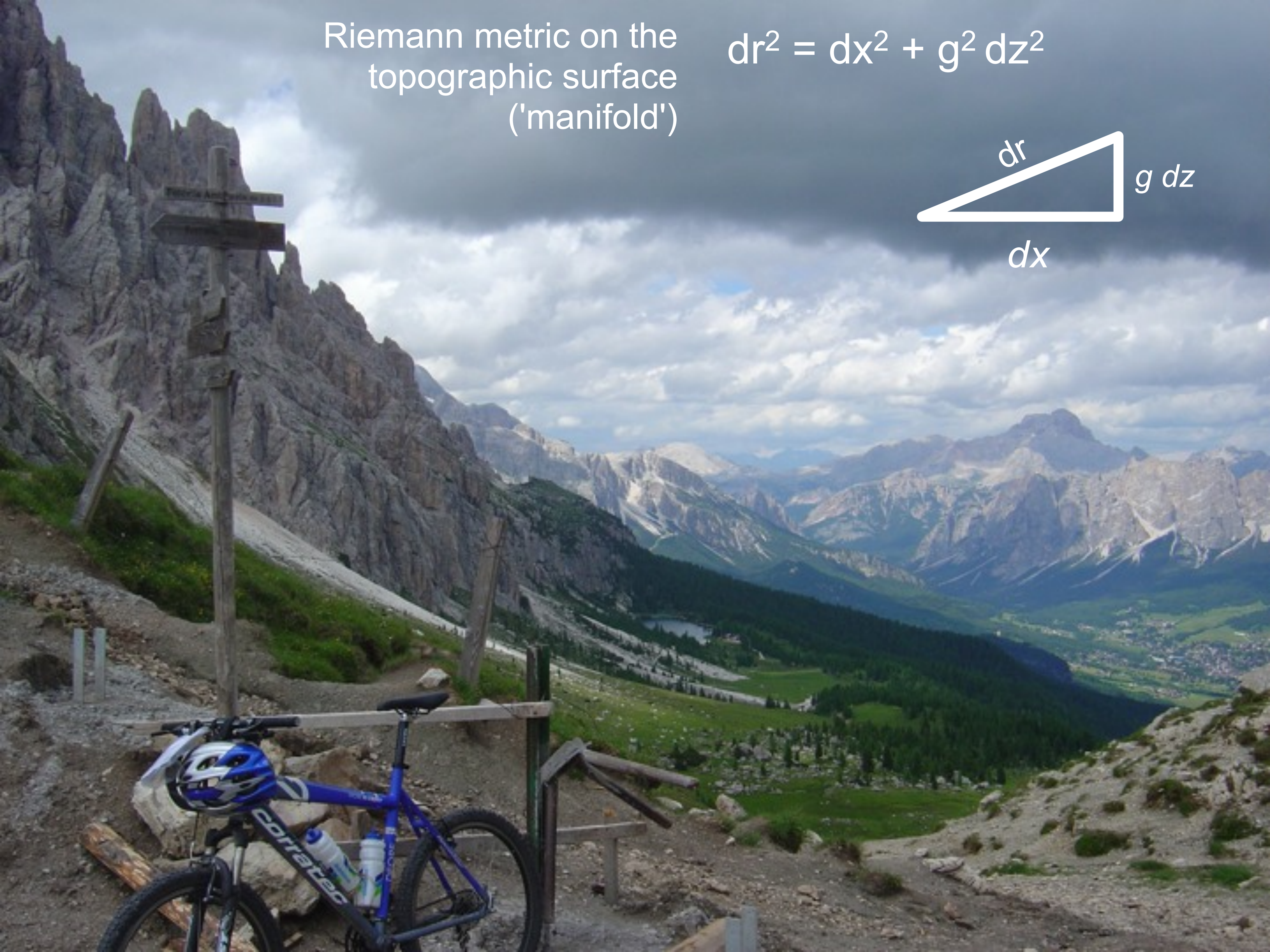


But we only used the nuclei. The boundaries are artificially straight.

How can we better use the information in the actin and tubulin channels?

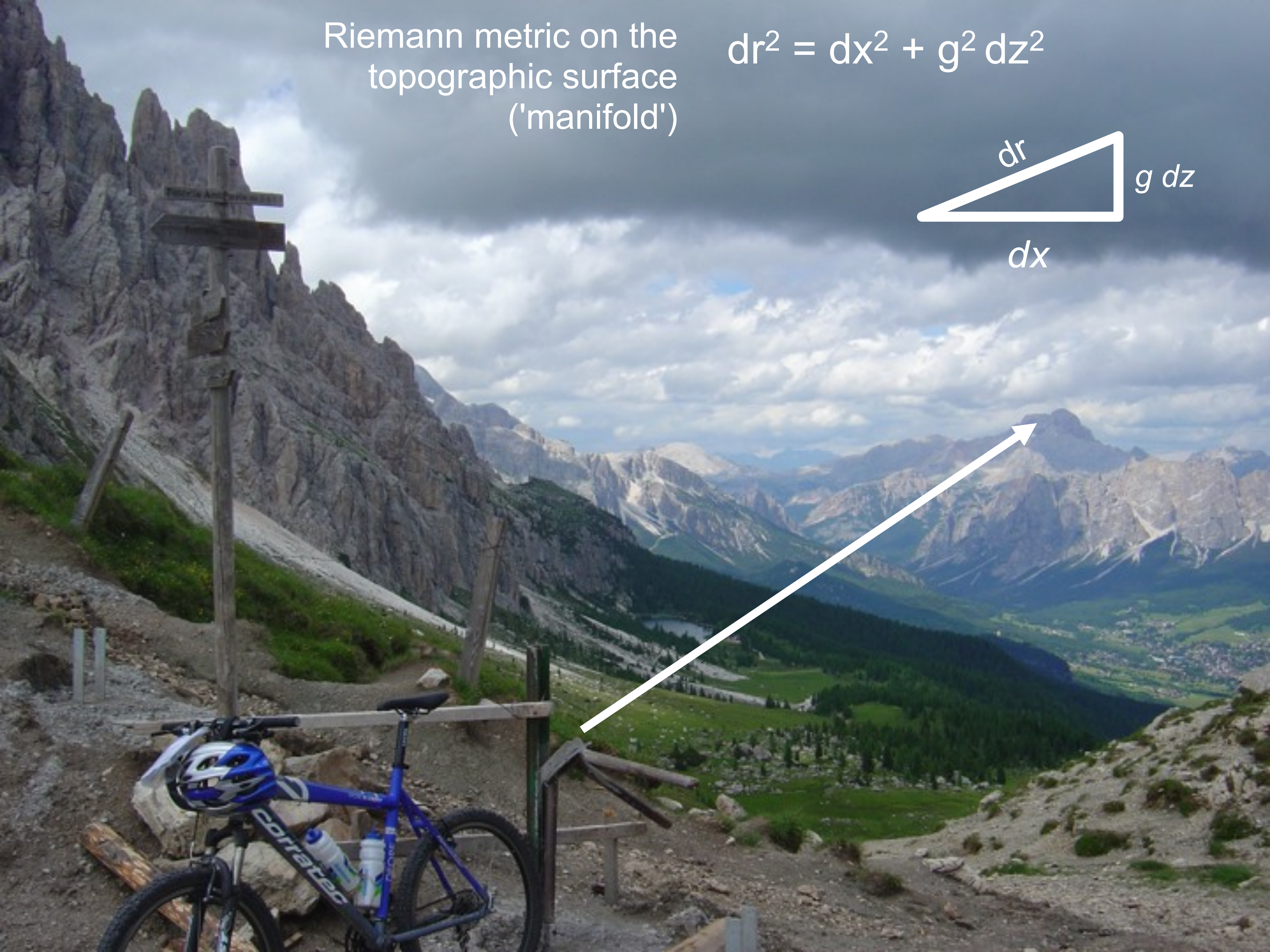
Riemann metric on the topographic surface ('manifold')

$$dr^2 = dx^2 + g^2\,dz^2$$



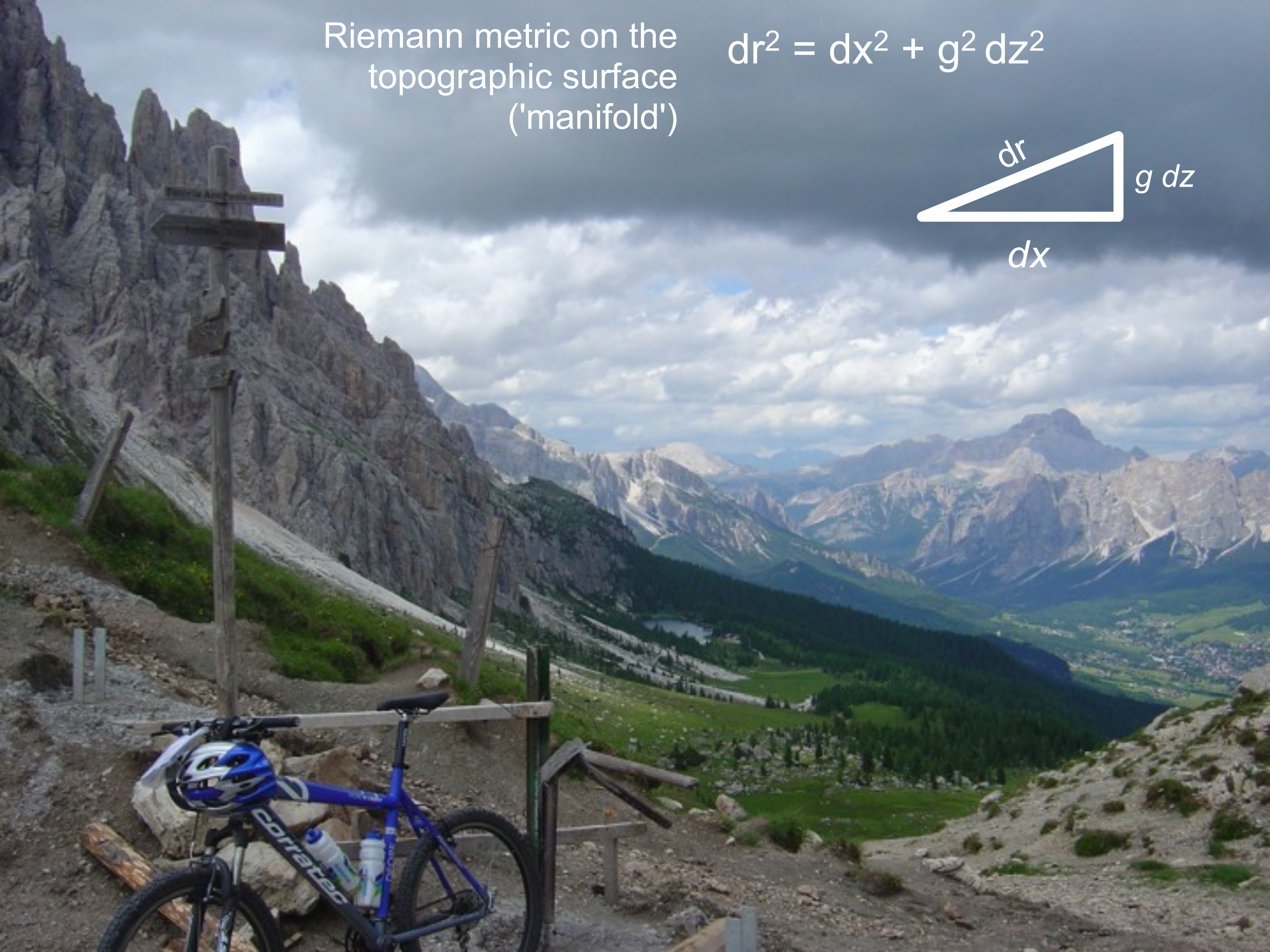
Riemann metric on the topographic surface ('manifold')

$$dr^2 = dx^2 + g^2\, dz^2$$



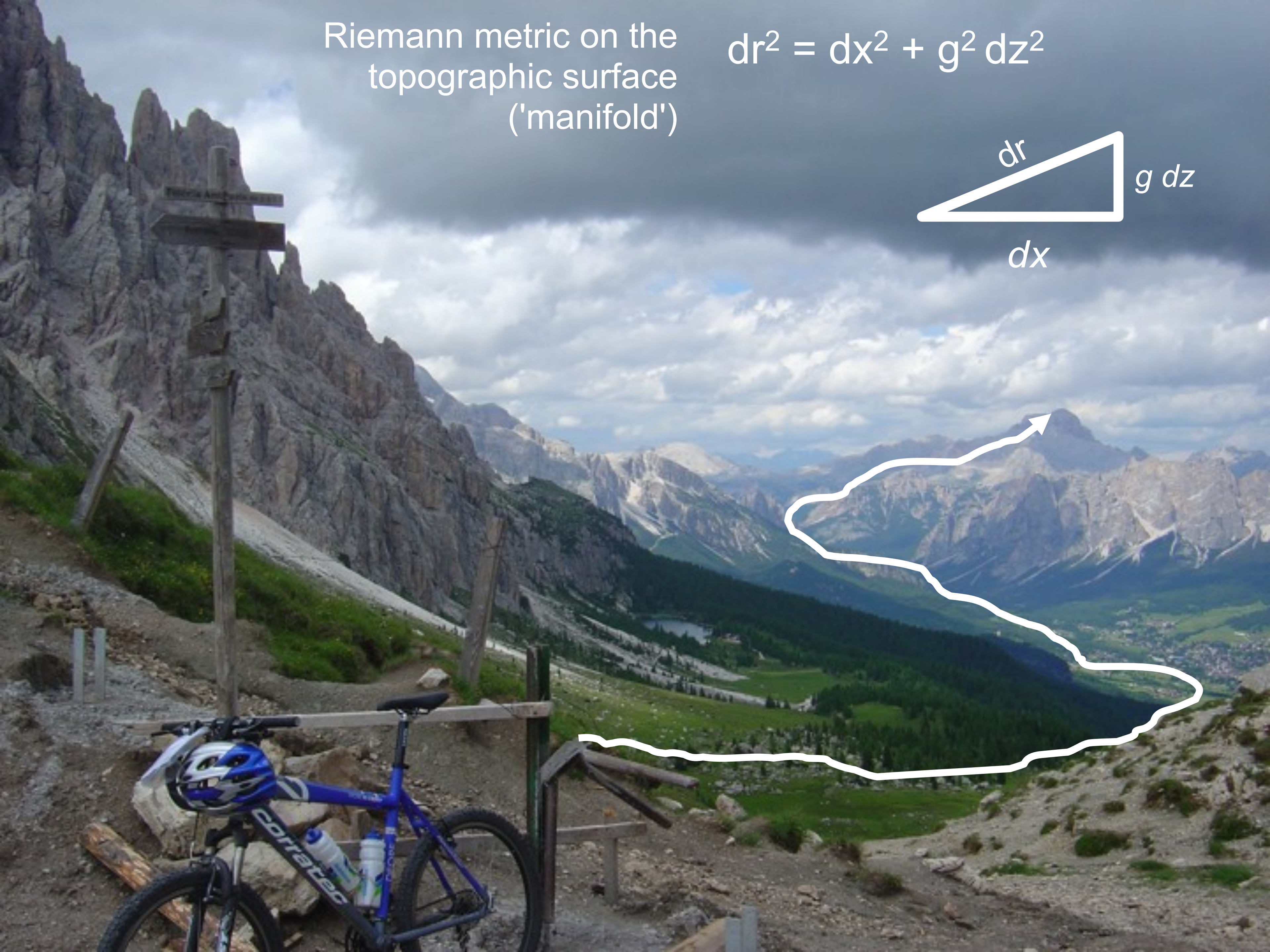
Riemann metric on the topographic surface ('manifold')

$$dr^2 = dx^2 + g^2\,dz^2$$

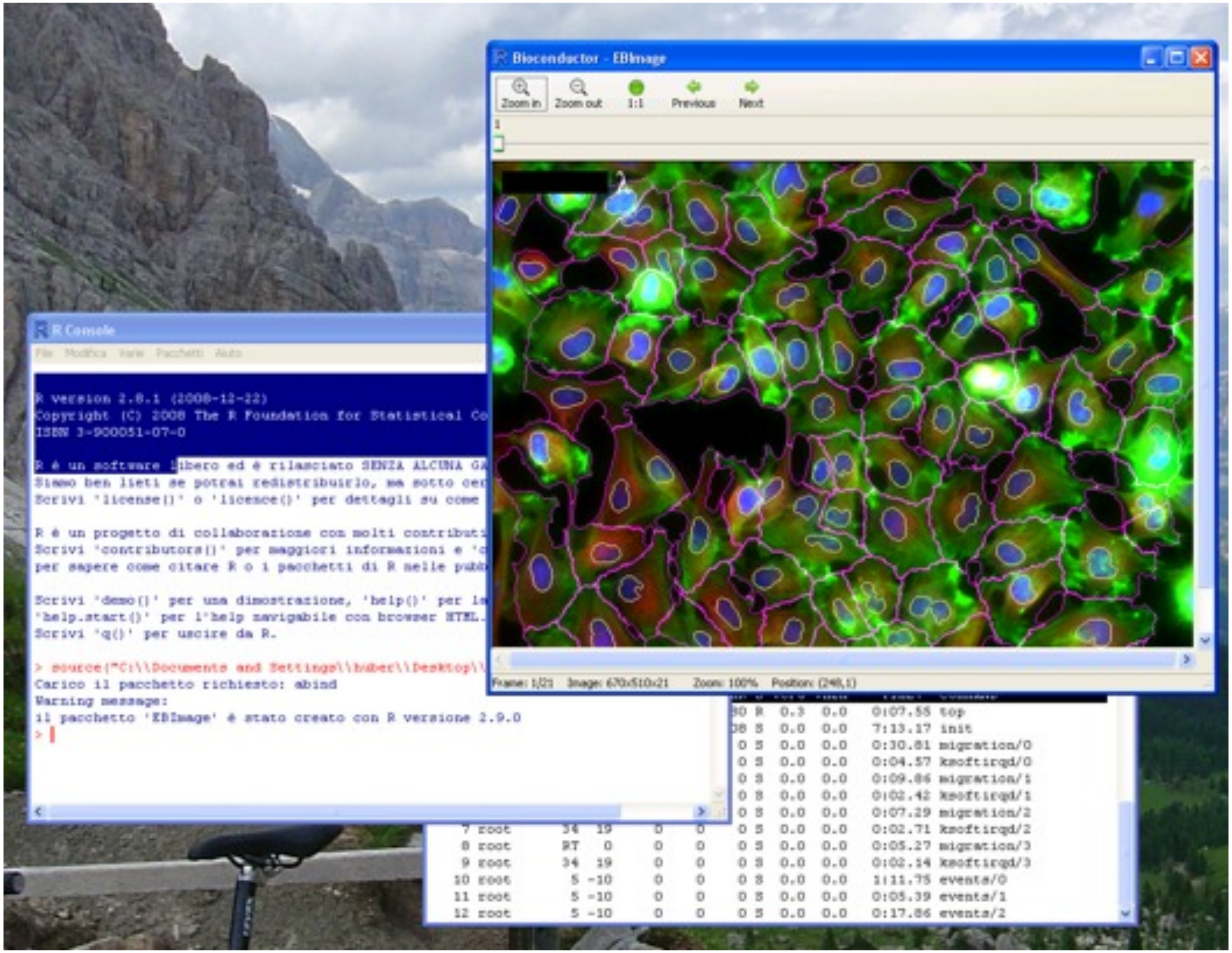


Riemann metric on the topographic surface ('manifold')

$$dr^2 = dx^2 + g^2\,dz^2$$



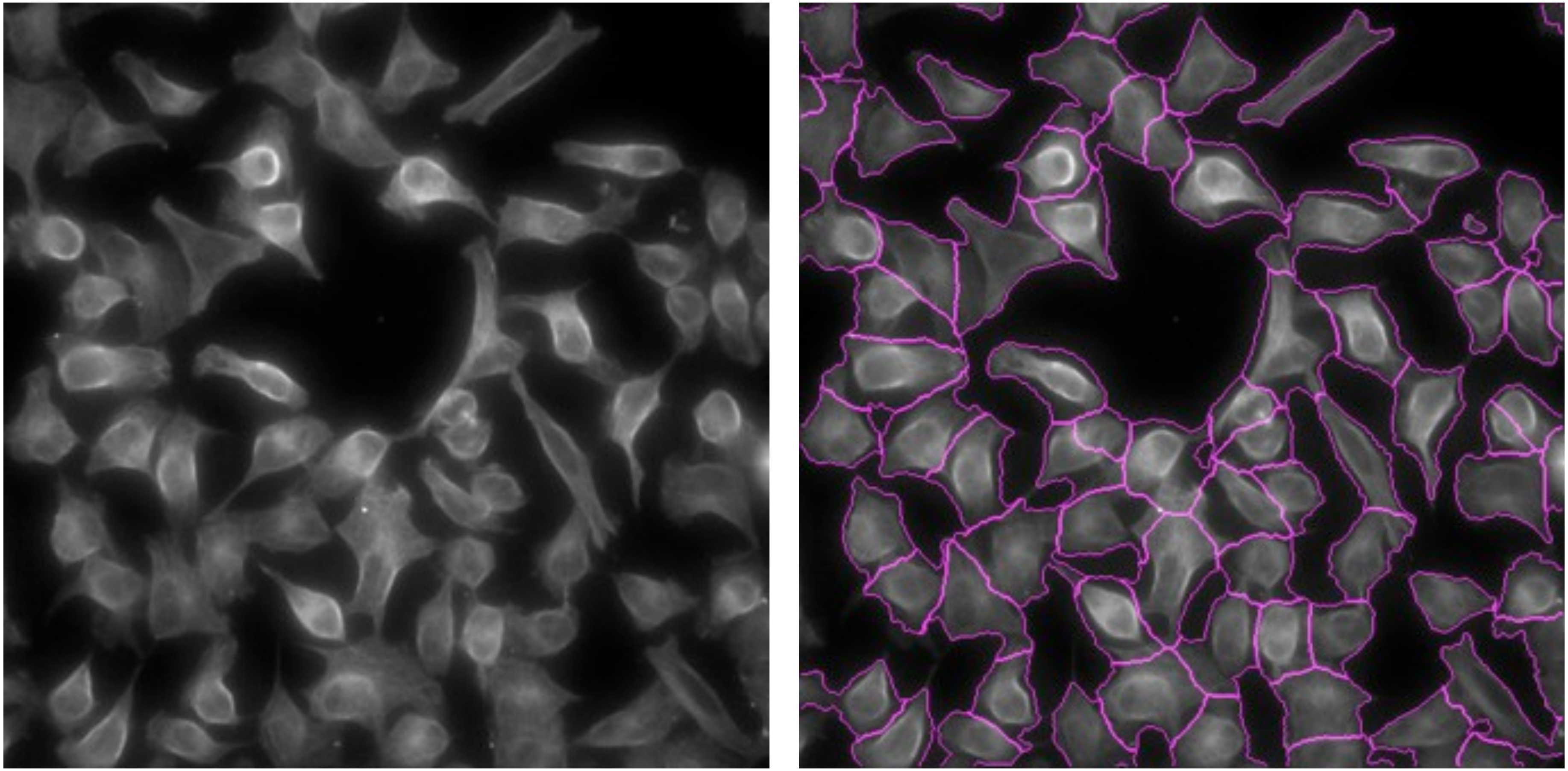
Voronoi segmentation on the image manifold



```
R version 2.8.1 (2008-12-22)
Copyright (C) 2008 The R Foundation for Statistical Co
ISBN 3-900051-07-0

R è un software libero ed è rilasciato SENZA ALCUNA GA
Siamo ben lieti se potrai redistribuirlo, ma sotto cer
Scrivi 'license()' o 'licence()' per dettagli su come

R è un progetto di collaborazione con molti contribut
Scrivi 'contributors()' per maggiori informazioni e 'c
per sapere come citare R o i pacchetti di R nelle pubb

Scrivi 'demo()' per una dimostrazione, 'help()' per l
'help.start()' per l'help navigabile con browser HTML.
Scrivi 'q()' per uscire da R.

> source("C:\\Documents and Settings\\huber\\Desktop\\
Carico il pacchetto richiesto: abind
Warning message:
il pacchetto 'EBImage' è stato creato con R versione 2.9.0
> 
```

```
                                         80 R  0.3  0.0   0:07.55 top
                                         08 S  0.0  0.0   7:13.17 init
                                          0 S  0.0  0.0   0:30.81 migration/0
                                          0 S  0.0  0.0   0:04.57 ksoftirqd/0
                                          0 S  0.0  0.0   0:09.86 migration/1
                                          0 S  0.0  0.0   0:02.42 ksoftirqd/1
                                          0 S  0.0  0.0   0:07.29 migration/2
   7 root       34  19     0     0        0 S  0.0  0.0   0:02.71 ksoftirqd/2
   8 root       RT   0     0     0        0 S  0.0  0.0   0:05.27 migration/3
   9 root       34  19     0     0        0 S  0.0  0.0   0:02.14 ksoftirqd/3
  10 root        5 -10     0     0        0 S  0.0  0.0   1:11.75 events/0
  11 root        5 -10     0     0        0 S  0.0  0.0   0:05.39 events/1
  12 root        5 -10     0     0        0 S  0.0  0.0   0:17.86 events/2
```

Segmentation result

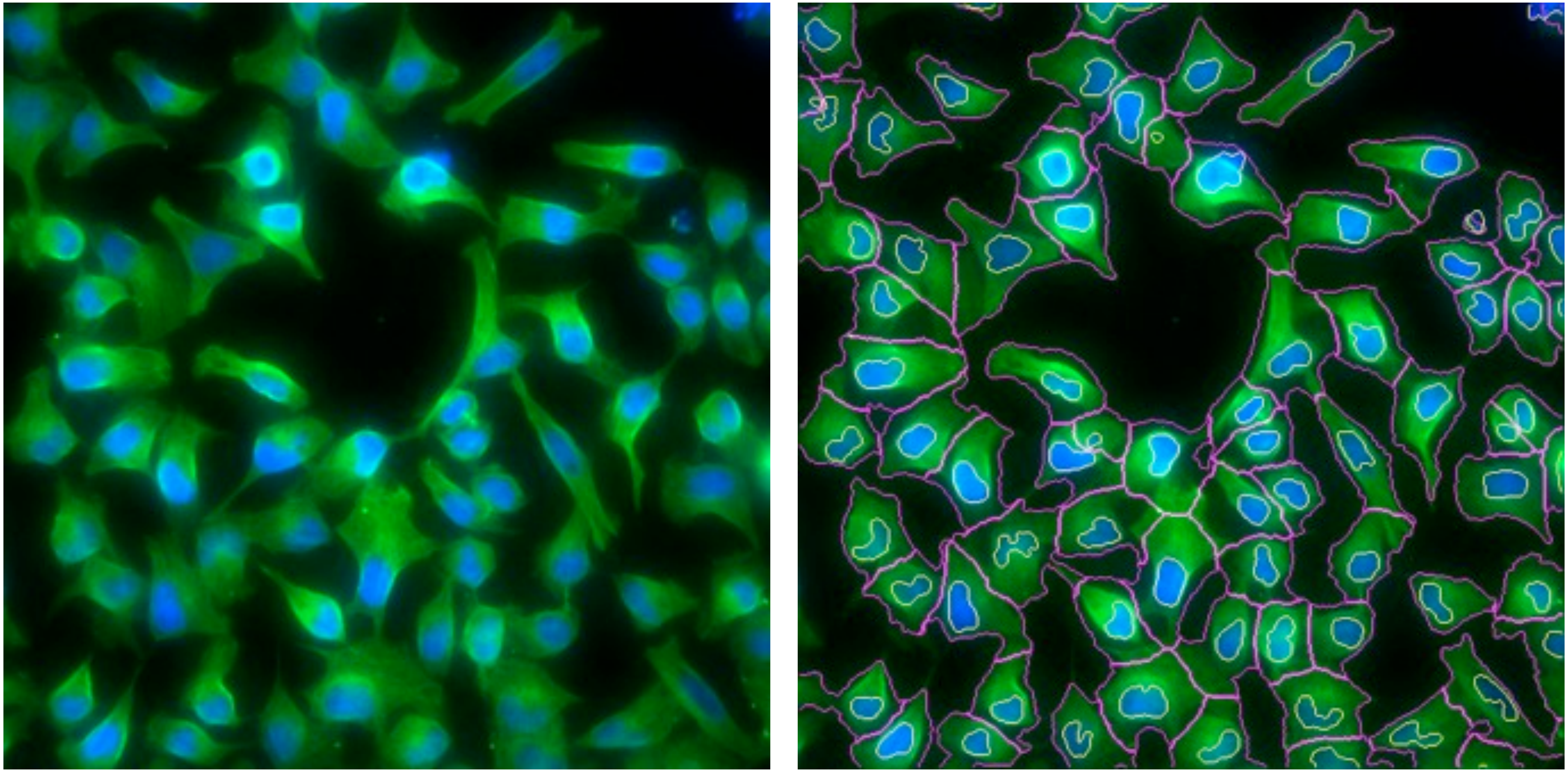


Fully automatic on all 88k images

Detailed resolution of boundaries also for adjacent cells

Would not deal with overlapping cells (multilayer, tissue)

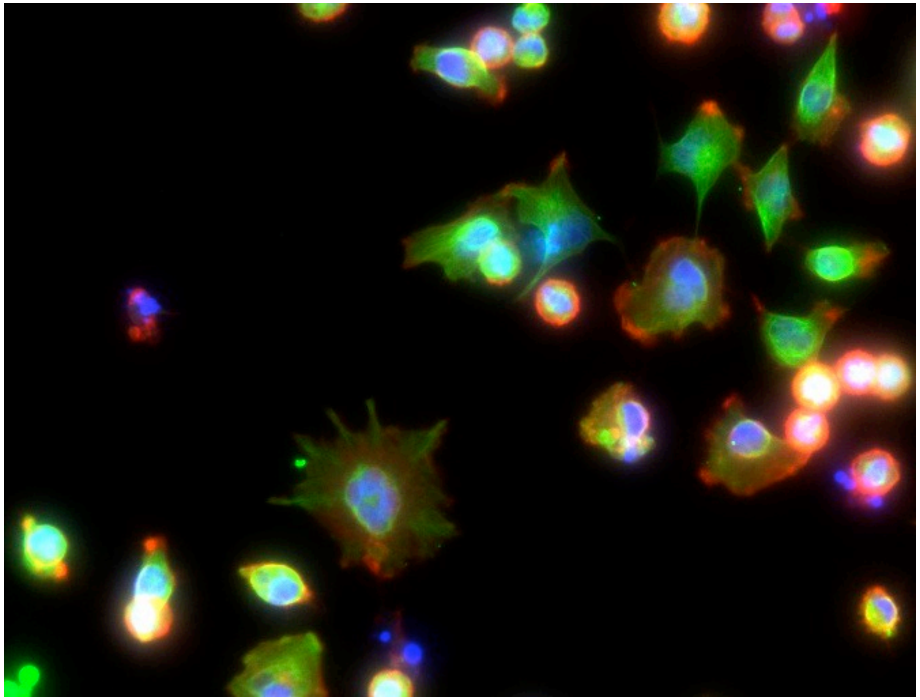
Combining channels and segmentation masks

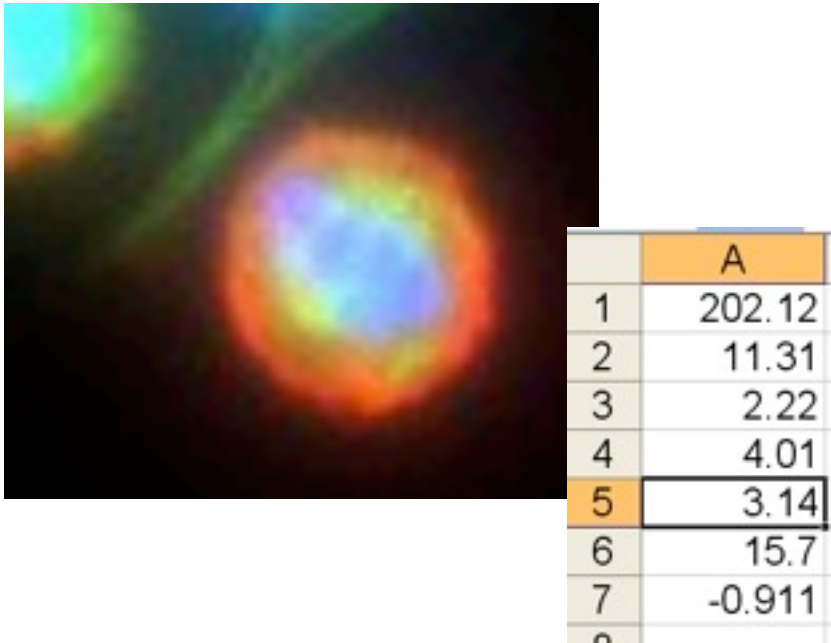


Extracting quantitative cell descriptors

translation and rotation invariant descriptors

- geometry (intensity, size, perimeter, eccentricity…)
- texture (Haralick, Zernike moments…) on each channel
- relative positions, joint distribution moments





	A
1	202.12
2	11.31
3	2.22
4	4.01
5	3.14
6	15.7
7	-0.911

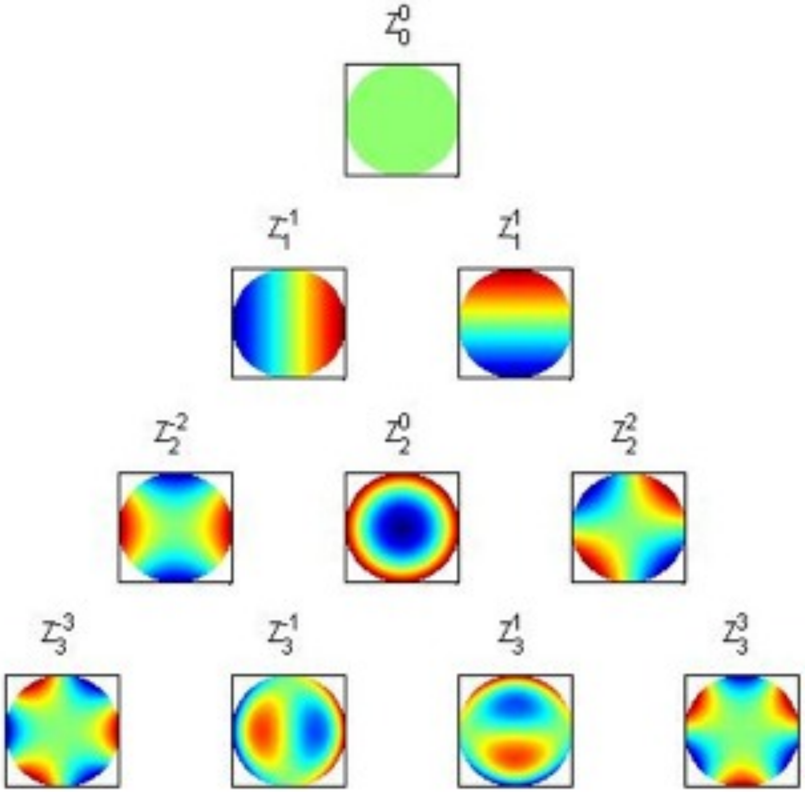
Zernike Moments

$$A_{mn} = \frac{m+1}{\pi} \int_{\substack{\text{unit}\\ \text{circle}}} \underset{\text{Zernike basis}}{\underline{e^{-in\theta} Z_{mn}(r)}}\ \underset{\text{Image}}{\underline{f(r,\theta)}}\ d\theta dr$$

$|n| \leq m$, $m-|n|$ even

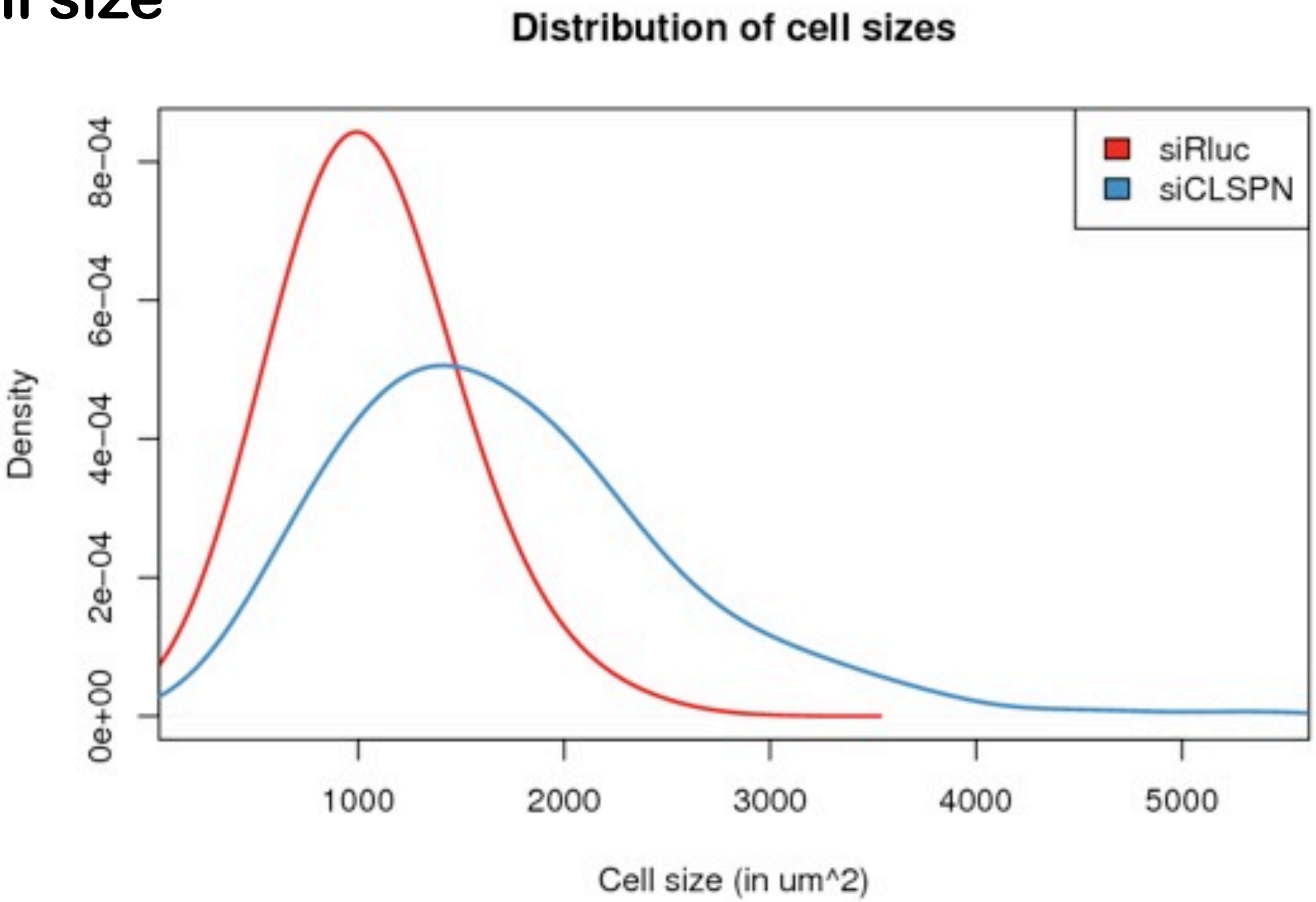
$|A_{mn}|$ rotation invariant

careful: f a discrete image, pixelisation of the circle



Features

Cell size

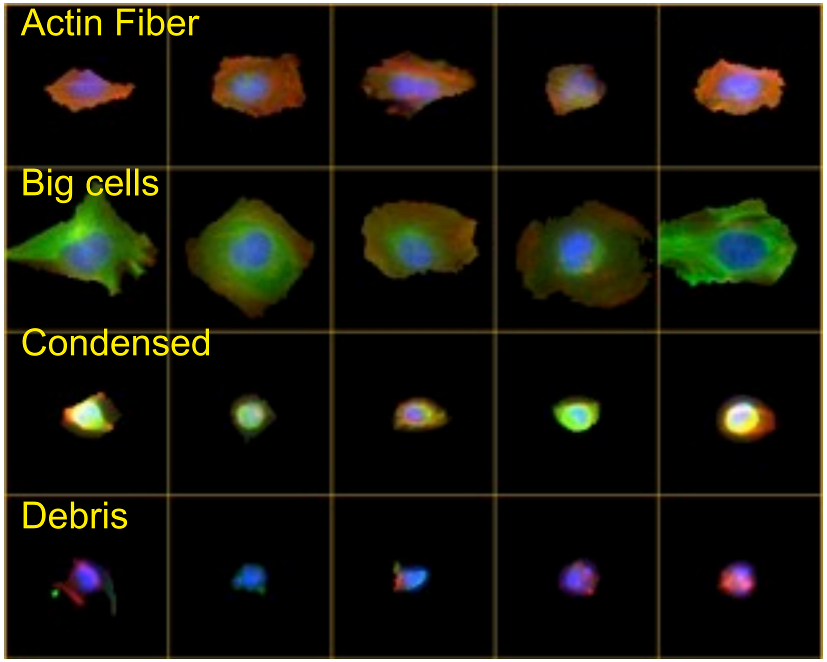


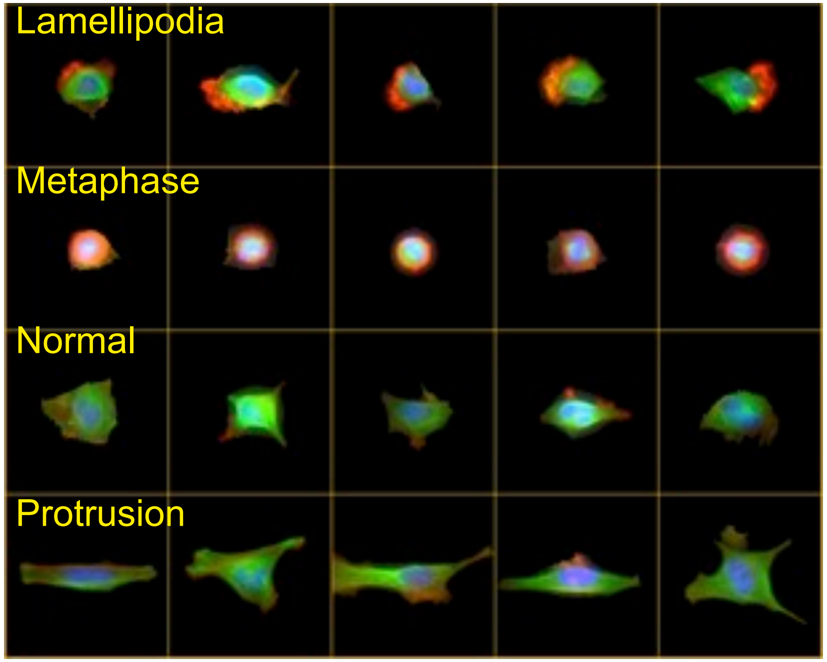
Cell classification

using the numeric descriptors

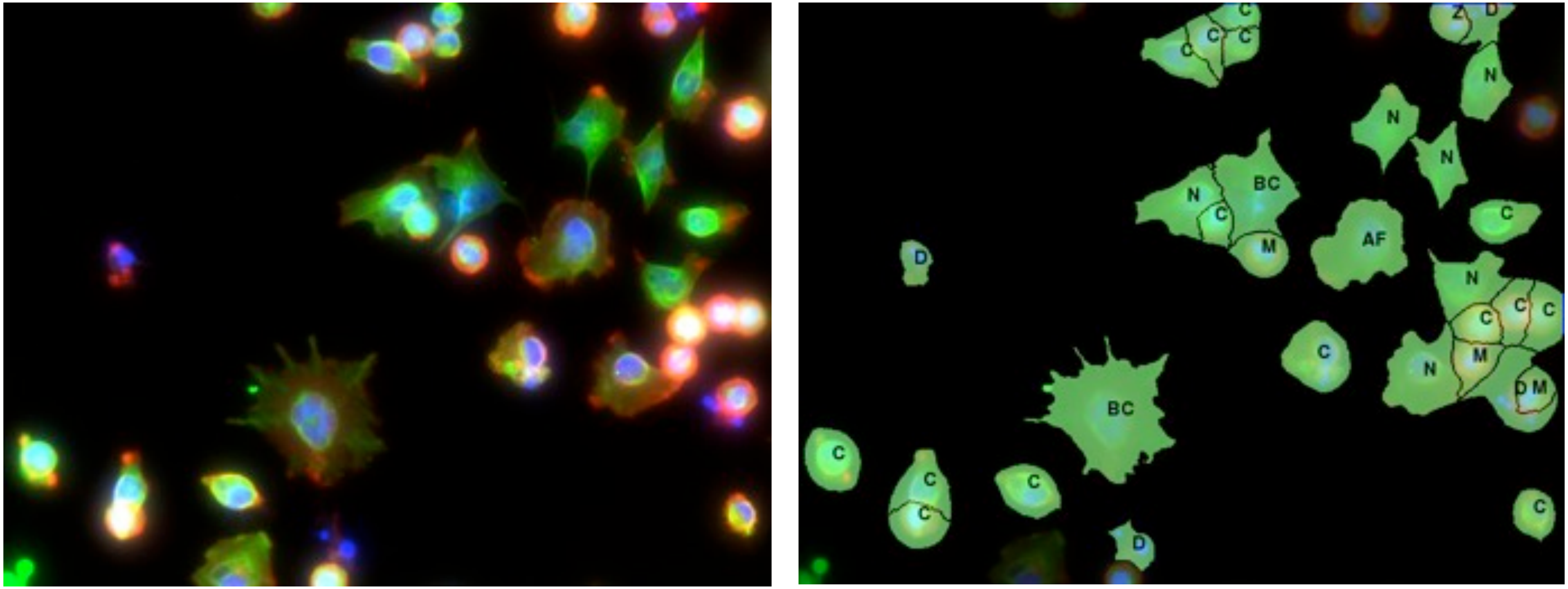
supervised learning, SVM

8 classes and a training set of ~3000 cells:

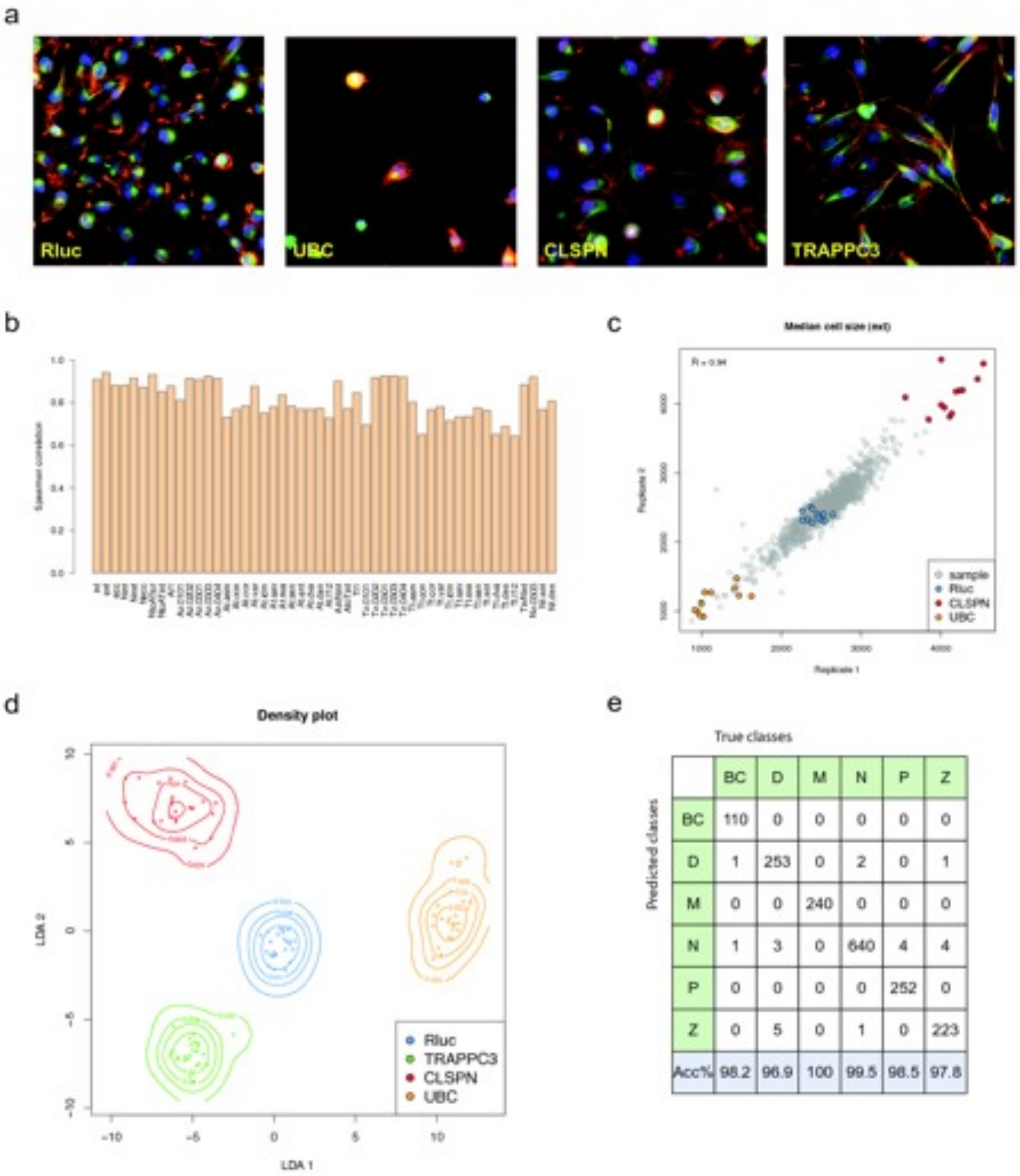




Cell classification

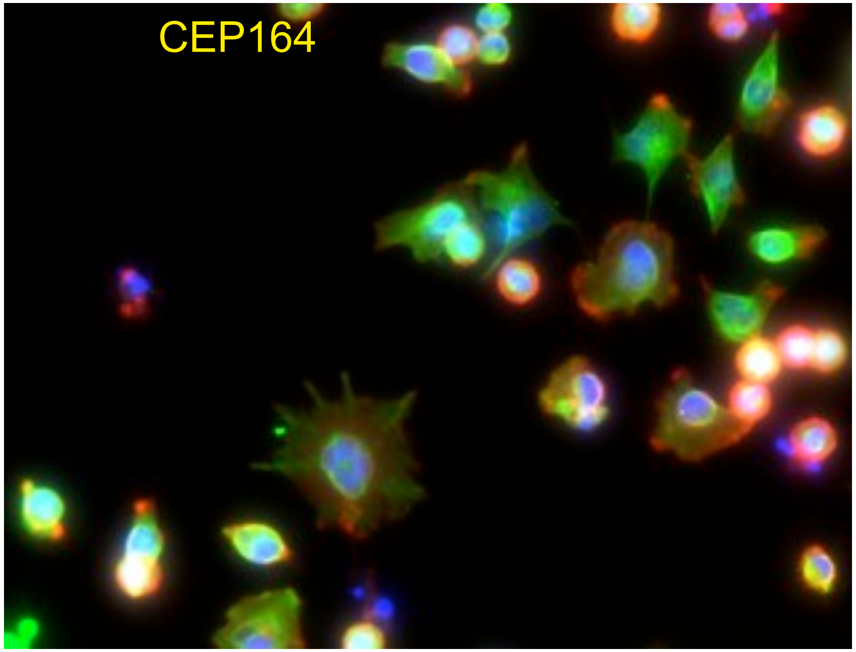


Cell classification



	BC	D	M	N	P	Z
BC	110	0	0	0	0	0
D	1	253	0	2	0	1
M	0	0	240	0	0	0
N	1	3	0	640	4	4
P	0	0	0	0	252	0
Z	0	5	0	1	0	223
Acc%	98.2	96.9	100	99.5	98.5	97.8

Each siRNA is characterized by its "phenotypic profile"



number of cells	128
average intensity	1054.8
average nuclear intensity	1225.6
average cell size	842.3
average nuclear size	278.7
average eccentricy	0.649
avg. nuclear / cell size	2.91
# AF (actin fibers)	2
# BC (big)	7
# M (mitotic)	15
# LA (lamellipodia)	0
# P (with protrusions)	17
# Z (telophase)	2

How do you measure

distance and **similarity**

in13-dimensional phenotypic

profile space?

Similarity depends on the choice and weighting of descriptors

Distance metric learning

Standard distances are not satisfying

Weighted Euclidean: $d(x, y) = \mathrm{sqrt}(\Sigma_i w_i(x_i - y_i)^2)$

General Euclidean: $d(x, y)^2 = (x-y)^t A(x-y)$

(Mahalanobis: $A = \Sigma^{-1}$)

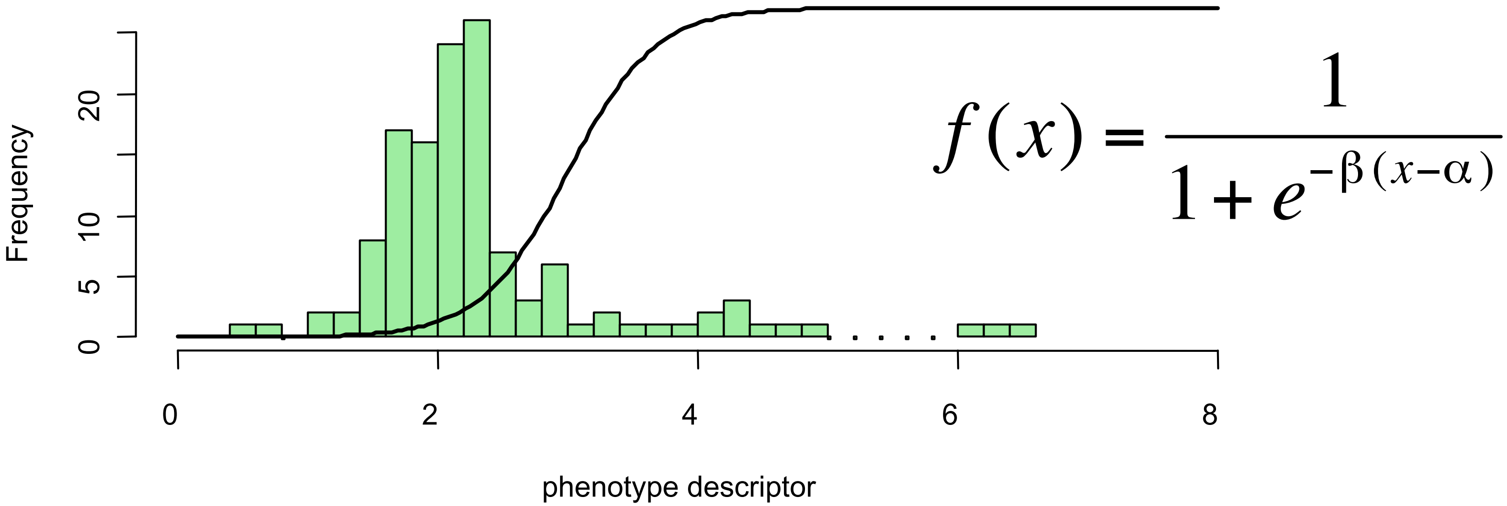
Cause: x_i contains the genetic effects as well as (unknown) experimental noise

Distance metric learning: learning a parametric distance

Using training set of gene pairs that are supposed to be “similar” and ”dissimilar”

Optimise parameters such that similar pairs have small distances, dissimilar pairs large.

Parametric sigmoid transformation of each phenotype descriptor into a score∈[0,1]



$$f(x) = \frac{1}{1 + e^{-\beta(x-\alpha)}}$$

2 parameters α, β for each of the 13 descriptors: measure its scale and interesting range

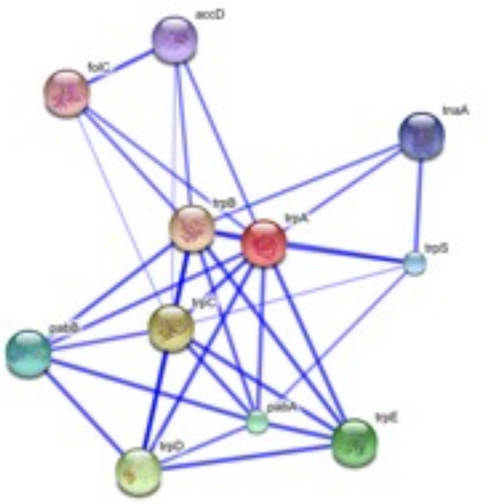
phenotypic distance: L_1 distance between two transformed phenotypic profile vectors

How can we fit the best transformation parameters?

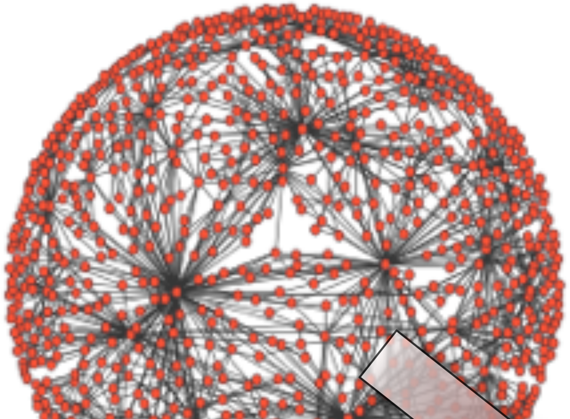
STRING is database of pairwise gene-gene associations.

Distance between gene pairs linked by STRING should on average (i.e. statistically) be lower than between random genes.

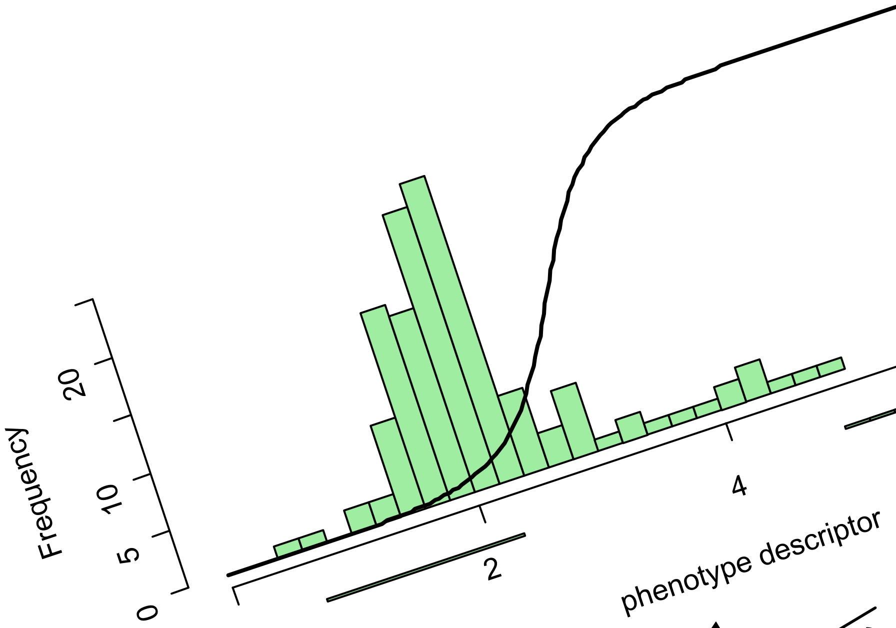
Solve a high-dimensional optimisation problem to obtain the best set of αs, βs



Metric learning



Training: network: 100,000s of ostensible protein-protein similarities

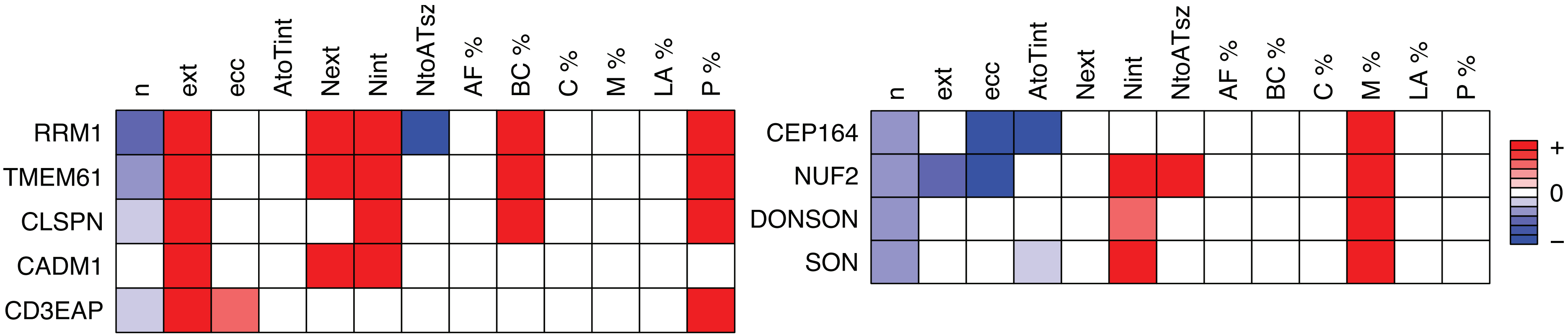


$$f(x)=\frac{1}{1+e^{-\beta(x-\alpha)}}$$

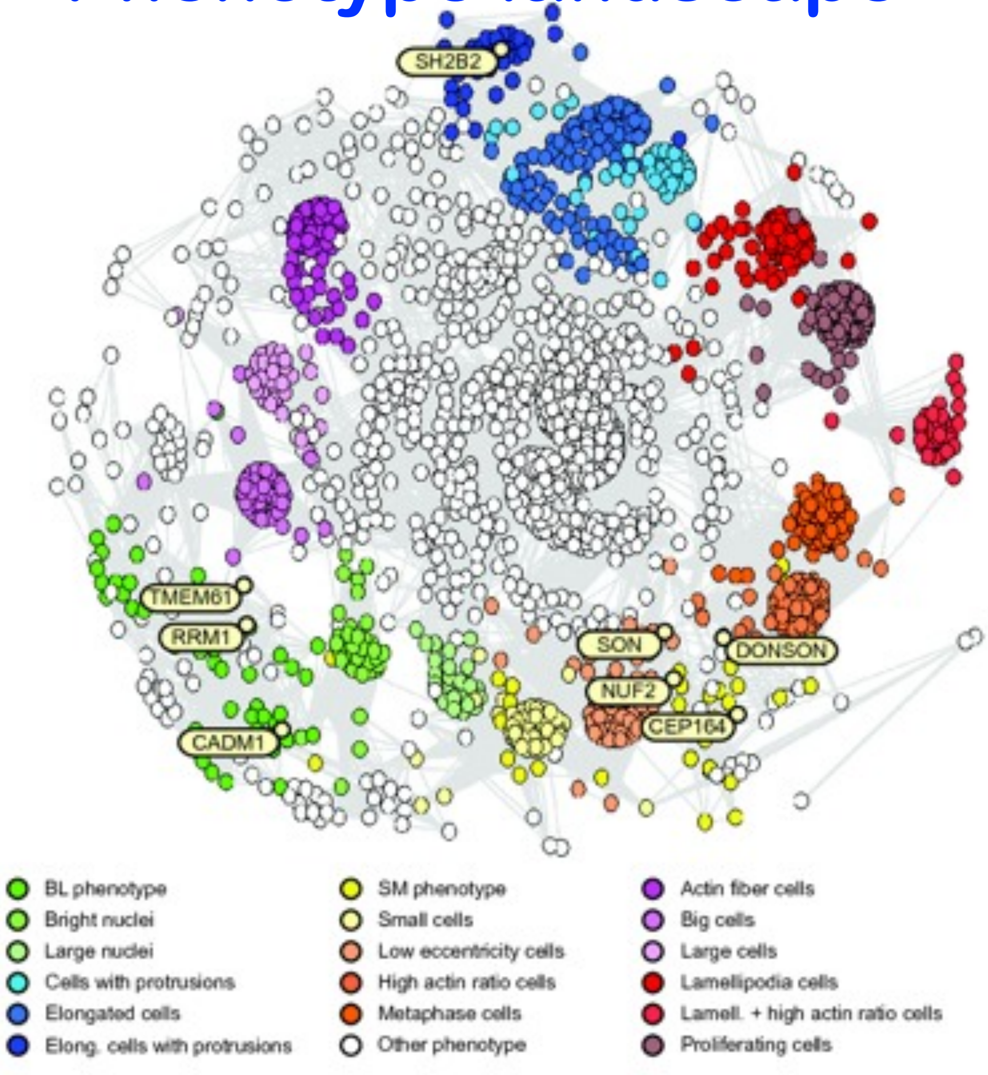
Learned: 13 x 2 transformation

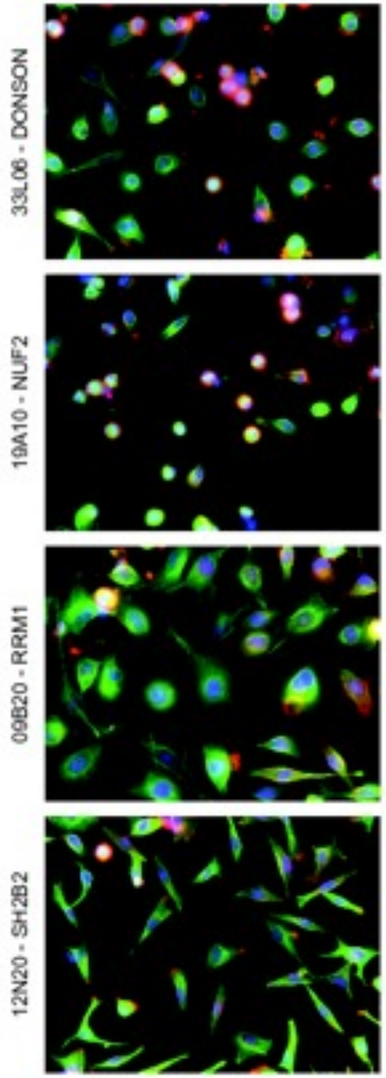
siRNA phenoprints

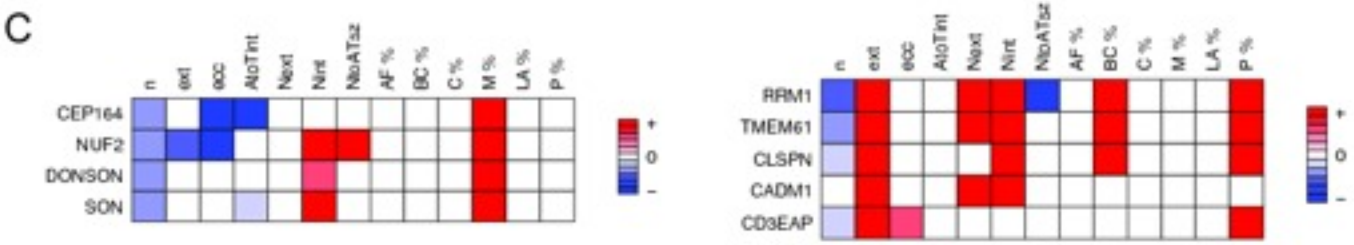
Among the 22839 siRNA pools, 1891 with non-null phenoprints.



Phenotype landscape

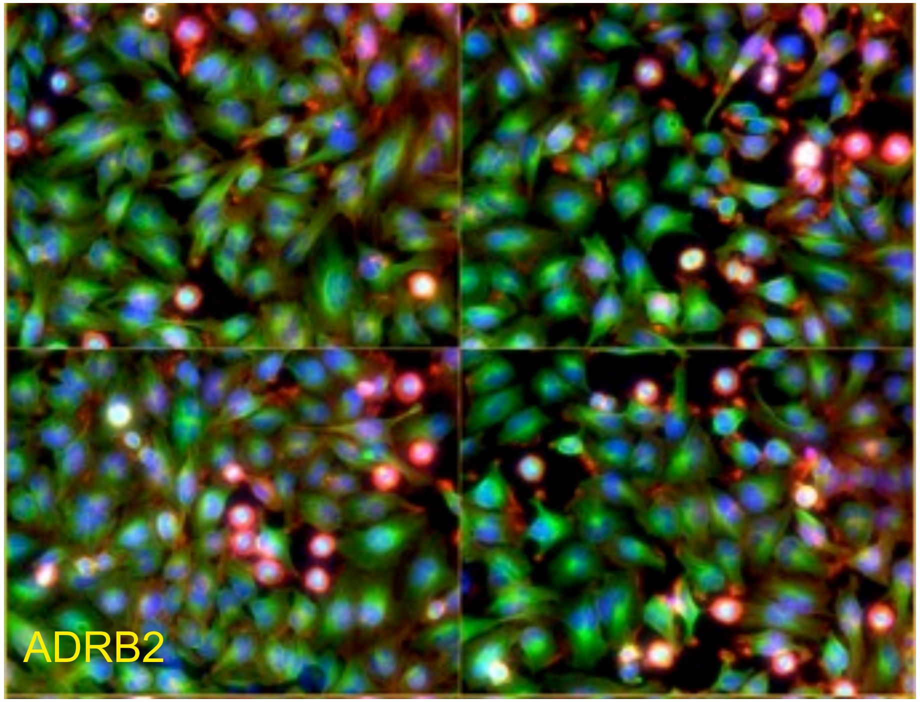




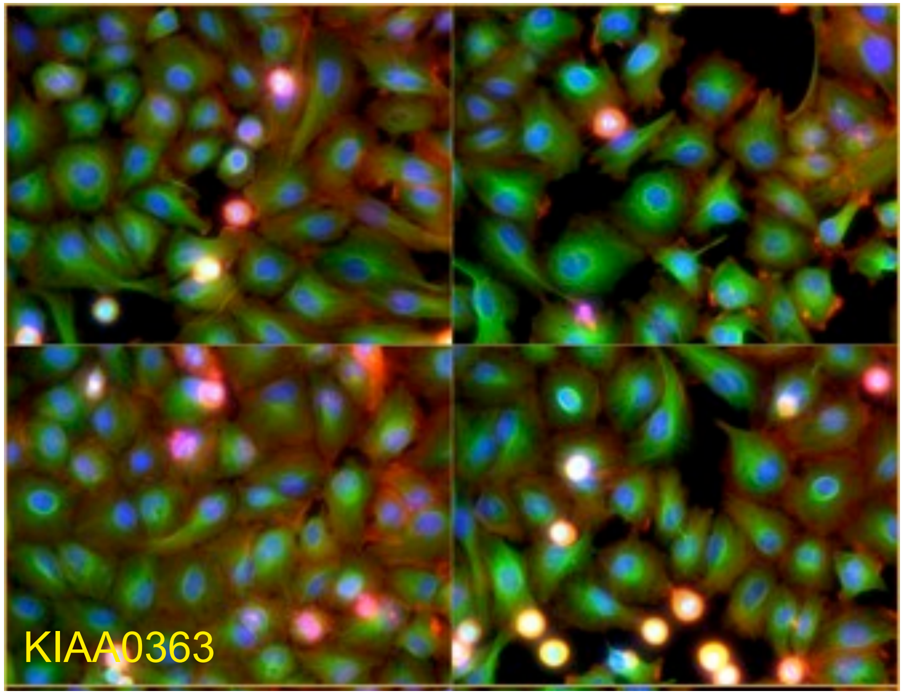


Example phenotypes

Many binucleated

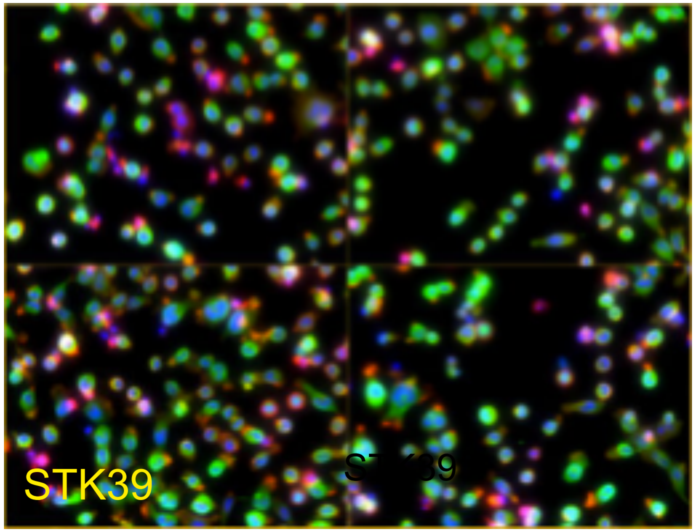


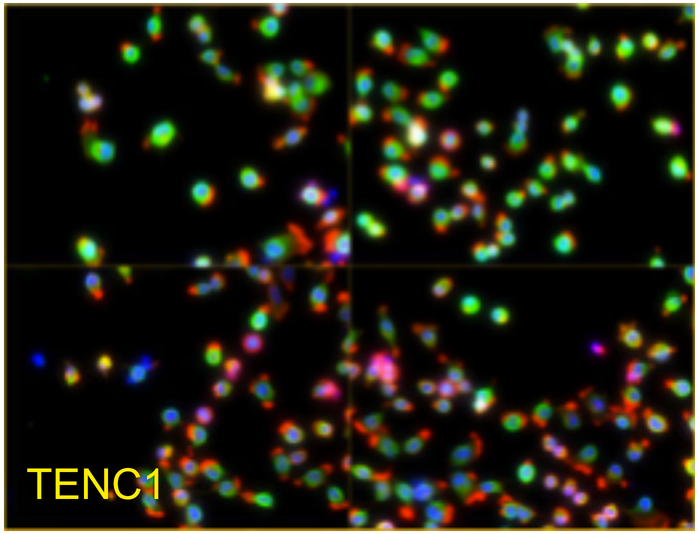
Many large



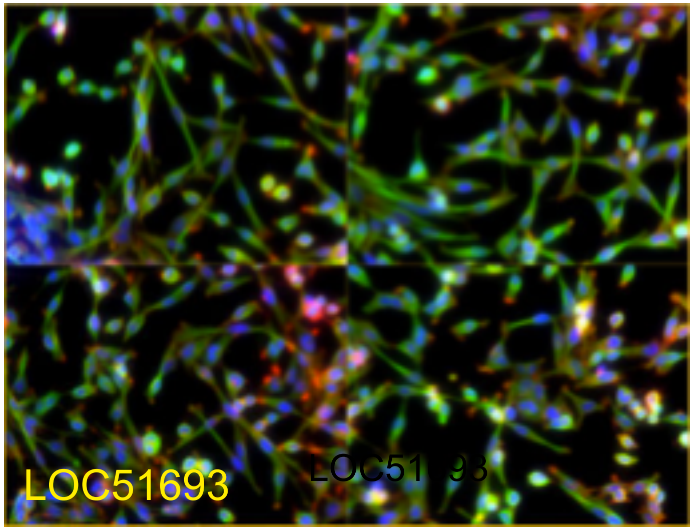
Example phenotypes

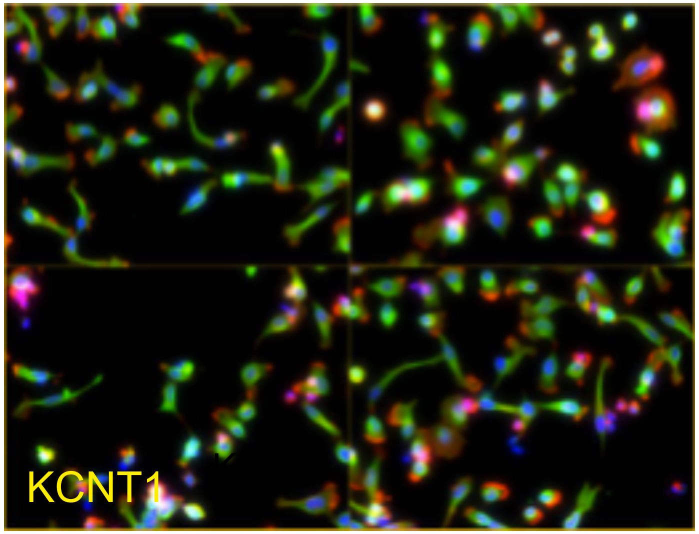
Condensed





Elongated





Follow Up

Such a map is a resource to generate hypotheses.

We characterized previously "untouched" genes from neighbourhoods with many genes in DNA damage response and genomic integrity.

Image-based phenotyping turned out to be a powerful method for functional discovery.

Greg Pau, Florian Fuchs, Dominique Kranz, Christoph Budjan, Oleg Sklyar, Thomas Horn, Michael Boutros

Some conclusions & outlook

Automated phenotype quantification of cellular populations

Multiparametric imaging

~92000 images: 660 h CPU time (but trivially parallelised - one night on the cluster)

Coming soon: `imageHTS`

Data and workflow management for automated analysis of cell-based imaging screens, based on `cellHTS2` and `EBImage`

Distributed and hierarchical (well, cell, features) web data access

EBImage demo

A nuclei and cell segmentation workflow demo

Vignette of the `imageHTS` package:

http://www-huber.embl.de/users/whuber/pub