A flexible image segmentation pipeline for heterogeneous multiplexed tissue images based on pixel classification

Vito Zanotelli & Bernd Bodenmiller

January 14, 2019

Abstract

Measuring objects and their intensities in images is basic step in many quantitative tissue image analysis workflows. We present a flexible and scalable image processing pipeline tailored to highly multiplexed images. This pipeline allows the single cell and image structure segmentation of hundreds of images. It is based on supervised pixel classification using Ilastik to the distill the segmentation relevant information from the multiplexed images in a semi-supervised, automated fashion, followed by standard image segmentation using CellProfiler. We provide a helper python package as well as customized CellProfiler modules that allow for a straight forward application of this workflow. As the pipeline is entirely build on open source tool it can be easily adapted to more specific problems and forms a solid basis for quantitative multiplexed tissue image analysis.

1 Introduction

Image segmentation, i.e. division of images into meaningful regions, is commonly used for quantitative image analysis [2, 8]. Tissue level comparisons often involve segmentation of the images in macrostructures, such as tumor and stroma, and calculating intensity levels and distributions in such structures [8]. Cytometry type tissue analysis aim to segment the images into pixels belonging to the same cell, with the goal to ultimately identify cellular phenotypes and celltypes [2]. Classically these approaches are mostly based on a single, hand selected nuclear marker that is thresholded to identify cell centers. If available a single membrane marker is used to expand the cell centers to full cells masks, often using watershed type algorithms. However nowadays several approaches have become available that allow a highly multiplexed measurement of molecular markers, allowing images with as many as 40 markers. In such images several markers have potentially information about the nuclear, cytoplasmic or membrane information of a pixel. Correspondingly it was already shown that a segmentation based on an optimized selection of a linear combination of these markers outperforms any single hand selected nuclear and membrane channel for segmentation [10]. However this approach does not utilize the interdependencies as well as texture information of markers present in such images. Supervised classification of pixels into relevant classes, such as nuclear-like and background-like, has been already proposed to be used to integrate the information encoded in the texture of high resolution but low dimensional images into probability maps that facilitate segmentation [5, 12, 7, 13]. We argue that this approach should also be particularly suited to integrate the channel information and textures found in multiplexed images. A flexible classification algorithm such as implemented in the Ilastik open source software should allows for a flexible pixel classification, that can be used to identify nuclear as well as membrane or cytoplasmic pixels over a wide range of cell types and phenotypes, after expert guided supervised training. Segmenting the resulting probability maps, indicative of the class association of the trained pixels, should allow for a robust cell identification by using segmentation routines implemented in software such as CellProfiler (Fig. 1).

We used this idea to build a flexible and scalable image processing pipeline to segment highly multiplexed images (Fig. 2). We developed the approach based on the multiplexed imaging technique imaging mass cytometry [3]. IMC allows the measurement off more than 40 markers at a resolution of 1 um in tissue sections, by exploiting a metal labeled antibody stain with a laser ablation coupled induced coupled plasma mass spectrometer. IMC raw data contains the pixel data in a flow-cytometry like pixel data file structure. Thus we build a python based converter package to convert the raw data formats into a standardized ome.tiff format [4]. This standardized format

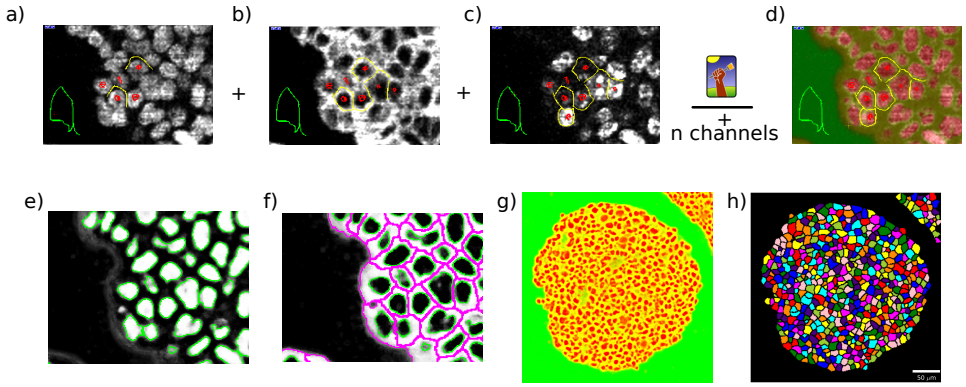

Figure 1: Ilastik is used to classify pixels according to nuclear (red), membrane/cytoplasm (yellow) and background (green) (d) using information from various channels (a-c). This achieves an integration of the class information from all available channels in a semi-supervised manner. CellProfiler is used to segment the class probability maps (white) to get nuclear (e) and cell level (f) segmentation masks. (g) shows an example probability map with (h) the corresponding segmentation. These masks are used to extract single cell information, such as mean marker levels and neighbourhood graphs using CellProfiler.

is the basis for the further pipeline. IMCtools can convert ome.tiff into formats and image stacks that can be directly used in CellProfiler and for Ilastik pixel classification. To build an optimized pipeline, existing CellProfiler modules were adapted and new modules written to facilitate the handling of the highly multiplexed image data and to allow for measurement of channel intensities as well as texture features of whole multiplexed image stacks. The resulting single cell masks and data can then be directly used in data analysis scripts as well as visualization tools such as HistoCAT [9].

2 Material and Methods

In the following section we give a detailed user guide for the proposed pipeline. Example configuration files and the mentioned CellProfiler pipelines can be found on the following Github page: `https://github.com/BodenmillerGroup/ImcSegmentationPipeline`. Specifically the ipython notebook found in `scripts/imc_preprocessing.ipynb` contains a detailed walk-through of an example analysis.

2.1 Pipeline overview

The developed image pipeline consists of the general steps:

- Installation of the required software
- Conversion of IMC data into a common file format
- Preparation of the pipeline metadata
- Generation of the analysis stacks
- Preparation of the input stacks for Ilastik
- Iterative training of the Ilastik pixel classifier
- Single cell segmentation using CellProfiler
- Multiplexed image measurements using CellProfiler
- Further analysis of the single cell data

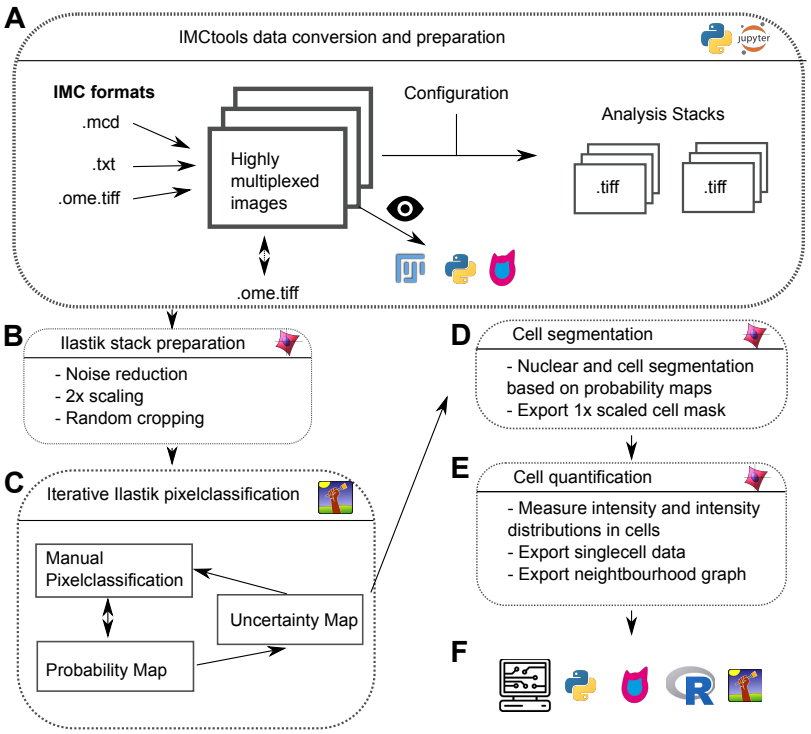

Figure 2: A schema of the proposed workflow. A) The imctools python package is used to convert raw data into ome.tiff and prepares analysis stacks. Various tools can be used for image visualization. B) The Ilastik analysis stack is preprocessed using CellProfiler C) Iterative Ilastik pixel classification is used to generate probability maps. D) The probability maps are segmented to single cell masks using CellProfiler. E) CellProfiler is used to generate single cell data in a standardized format. F) The single cell data can be analyzed using various tools.

2.2 Installation of the required software

The installation procedure is detailed in the ipython notebook `scripts/imc_preprocessing`. The procedure relies on using the *conda* package manager to generate reproducible environments.

2.3 Conversion of IMC data into a common file format

IMC data commonly comes as a vendor controlled .mcd or .txt file. To make the following pipeline generally applicable to multiplexed imaging data and independent of the vendor format, the raw files are first converted into an ome.tiff format [4].

For IMC data this one multiplane tiff file per acquisition. Each channel needs to have the channel label attribute as well as the fluor attribute set. For IMC data the metal name followed by the isotopic mass are used with the form: (IsotopeShortname)(Mass), e.g. Ir191 for Iridium isotope 191.

2.4 Generation of the analysis stacks

In the next step the converted .ome.tiff files are converted in a stack format suitable for further analysis, e.g. using CellProfiler. In a basic pipeline two stacks will be prepared: a 'Full' stack, containing all the channels chosen for CellProfiler quantification as well as the 'Ilastik' stack, containing all the channels selected for the Ilastik pixel classification. It is straight forward to modify this step to generate additional stacks, e.g. for additional tissue structure segmentations.

2.5 Preparation of input stacks for Ilastik

This step will allow a preprocessing of the images used for pixel classification using CellProfiler (example pipeline 1_ilastik_preprocessing). Common steps include removing outlier pixels (custom module Smooth Multichannel, 'Remove single hot pixels') as well as scaling the images two fold. In our experience scaling the images two fold facilitates the manual pixel classification using Ilastik considerable when using low resolution images such as produced by IMC. Additionally we recommend to crop random section from the image and only use these for training. Often the tissue structures on an image are rather similar, thus cropping of the random section will allows to train the pixel on smaller images, reducing the computational requirements of the classification. The preprocessed images are saved in an Ilastik compatible format.

2.6 Pixel classification using Ilastik

To train the Ilastik pixel classifier, an instance of Ilastik is opened and a new pixel-classification workflow is generated [13]. Then the image crops exported for Ilastik classification should be loaded as training data. As a next step in Ilastik, the features fore classification are selected. We recommend generally to select features generously here (all features between 1 and 10 pixels), if the computational resources allow it, and use the Ilastik 'suggest feature' feature selection after some initial training. In the next step the pixel classification is performed. For this step 3 classes: nuclear, cytoplasmic/membrane and background pixels are created. The classifier can be trained by generating a training set by manually drawing pixels for the respective classes. A training is most effective, if it contains a diverse collection of pixels and not to many very similar pixels. Thus we recommend to use a small brush size (e.g. 1 pixel) and label sparsely, as nearby pixels are often nearly identical. The channels can be changed in the lower left corner of the classification window in the input-channel layer. As the channels can have widely different intensities, the 'window leveling' tool should be used to adjust the visualization. To maximize the training efficiency we recommend to first draw some obvious pixels, e.g. from known nuclear and membrane channels. Then the 'live update' should be activated and the 'Uncertainty' checked. Pixels with high uncertainty provide the highest value for new training data and thus they should be preferably manually classified. Classification should be done until the uncertainty looks low (=transparent) except for the class borders, e.g. around nuclei. Once the uncertainties for an image crop look good, other image crops should be checked. To systematically check the uncertainty for all images we also recommend to use the 'Prediction Export' function to export the uncertainties as easily browsable .PNG images.

Once the segmentation on the image crops look sufficiently certain, the 'Batch processing' function of Ilastik can be used to convert the uncropped, scaled Ilastik stacks into probabilities.

As an additional step we recommend to use the imctools script 'probabilities2uncertainties' to convert these probability maps into big uncertainty maps for visual inspection. If there are regions not contained in the crops whose classification should be improved, e.g. FIJI can be used to manually crop these regions and add them to the Ilastik input data.

2.7 Single Cell Segmentation of probability maps using CellProfiler

To segment the probability maps a CellProfiler pipeline is used (pipeline 2_segment_ilastik). First the probability map stack is split into nuclear, cytoplasm/membrane and background. Then using the image-math module an image with the sum of nuclear+cytoplasmic signal is generated. The 'IdentifyPrimaryObjects' module is used to segment the nuclear masks to identify nuclei. Afterwards the 'IdentifySecondaryObjects' module is run with the nuclei and the nuclear+cytoplasma image to identify cells. As the segmentation is done on 2x scaled probability images, as a next step the module 'rescale objects' is used to rescale the masks to 1x resolution. This module removes pixels which are ambiguous after rescaling as they are combinations of pixels from different objects at 2x resolution. If wanted 'Identify secondary objects' can be used again on the cells to fill the gaps generated by this approach. Finally the masks are saved to be used in the next step. Alternatively the masks and images can also be directly exported for HistoCAT analysis.

2.8 Quantification of single cell features using CellProfiler

The quantification of channels per object is done by using the CellProfiler pipeline '3_measure_mask'. In this pipeline the mask from the previous step as well as the full analysis stack, containing all the channels to be analysed, are loaded. Additional image stacks, such as the probability stack, can be loaded as well. Further filtering outlier pixels can be considered as a preprocessing step for the analysis stack. If an experiment to assess the spillover matrix of the antibody conjugates was performed, at this step also the compensation can be applied to the images using the 'CorrectSpilloverApply' module. Then the custom modules 'MeasureObjectIntensity Multichannel' as well as 'MeasureImageIntensity Multichannel' can be used to get object as image level statistics. The standard module 'MeasureObjectNeighbors' can be used to identify the neightbourhood graph of the objects. Finally all these measurements can be saved.

2.9 Further analysis of the single cell data

The downstream analysis of the generated single cell data can be highly diverse. E.g. the exported data can be loaded in R or python scripts for statistical analysis or can be visualized in tools such as HistoCAT [9].

3 Results

3.1 Overview

Segmenting heterogeneous tissue images is challenging, as often nuclear, cytoplasm and membrane markers are very differentially expressed in different tissue parts. Using highly multiplexed images such as Imaging Mass Cytometry images combined with a broad selection of clearly localized markers can useful for this task, given that the multivariate nature of the data can be accounted for during the segmentation step. We propose that segmentation of probability masks based on supervised pixel classification is well suited for this task (Fig. 1). Using the excellent Ilastik framework, the user can browse the available channels to classify a training set of pixels into different classes, such as nuclear, cytoplasmic/membranous and background pixels in the case of single cell segmentation. This training data is then used to train a supervised, random forest based learning algorithm is learning the pixel classes on the basis of all the channels as well as derived variants capturing gradient as well as texture information [13]. After an iterative training process of the classifiers, probability maps for the trained classes can be exported. These probability maps provide a highly integrated view of the image information contained in the interdependencies, texture and gradients of the multiplexed image channels in respect to the classes of interest. For single cell segmentation the nuclear probability as well as the cytoplasm/membrane probabilities can be directly used by classical image segmentation algorithms, that usually expect a nuclear

as well as a cytoplasmic/membranous channel for segmentation. These pixel probabilities are strongly normalized, having values between 0 and 1, and are largely independent of actual channel intensities. Further the random forest based training is flexible enough to learn complex marker relationships in a robust way, even despite the large expression heterogeneity of markers present in tissues. Due to this standardized nature of the pixel probability maps, the segmentation can be run in a largely unsupervised manner, making the approach suitable for a high throughput setting.

3.2 Pipeline

We implemented the above idea into an efficient, high throughput compatible pipeline suited to analyse heterogeneous, multiplexed tissue image data (Fig. 2). The approximate processing times per image are indicated for a 500x500 pixel image with 30 channels on a single computing core. A detailed step by step description can be found in the Material and Methods. As a first step of the pipeline the input images are converted from vendor specific formats into a standardized ome.tiff image [4], making the pipeline suitable for a wide range of multiplexed imaging data ($\sim$seconds). For IMC data we developed the *imctools* python packages to convert existing .txt and .mcd IMC images formats into ome.tiffs. These ome.tiffs can either viewed in FIJI, using the imctools package as a FIJI plugin or converted by *imctools* for visualization in the HistoCAT or HistoCAT++ toolboxes. Based on configuration files, the images are split into a stack containing all channels that should be analyzed and a 'segmentation' stack containing the planes informative for segmentation ($\sim$seconds per image) using *imctools*. As a rule of thumb we recommend that all channels with clear localized markers should be used for segmentation. The 'segmentation' stack is then preprocessed using CellProfiler and exported for Ilastik pixel classification. For low-resolution IMC images, as part of the preprocessing the images are scaled two fold. This allows an easier classification process as the images appear smoother. To make the classification process more scalable, the random sections of fixed size are cropped for classification during the preprocessing step ($\sim$minutes).

For pixel classification the Ilastik software is used. The preprocessed and cropped segmentation images are loaded and a selection of derived features, quantifying gradients and texture of channels are calculated. For single cell segmentation of tissues, three classes are trained: nuclear pixels, cytoplasmic/membrane pixels as well as background pixels. Based on the appearance of individual marker channels, an expert user can interactively train the random forest based classifier. We suggest to do an iterative classification, trying to minimize the estimated uncertainty of the pixel classification by visually inspecting the uncertainty maps. Depending on the heterogeneity of tissue and the information content of the measured channels, this classification can take several hours for large datasets containing hundreds of heterogeneous images. The trained classifier can then be applied to the dataset in a batch mode ($\sim$seconds-minutes).

Using CellProfiler the resulting probability maps are then segmented in two steps into a cell mask, by first identifying the nuclear mask and then expanding it to the cellular mask ($\sim$seconds-minutes)

The resulting mask can then be used to retrieve per-cell or object information from a stack of channels using a CellProfiler pipeline. To streamline the analysis with CellProfiler, existing CellProfiler modules were modified to allow an efficient measurement of large image stacks. Additionally CellProfiler can be used to extract the neighbourhood graph of the cells. ($\sim$minutes).

The single cell data can then be exported as standardized text files that can be analysed using custom scripts or specialized software such as HistoCAT [9].

The presented workflow takes an accumulated processing time in the order of 5-10 minutes per image. Except for the manual pixel classification step, the processing of the pipeline can be fully automatized and can be run in a parallelized fashion, scaling linearly with the number of images. The manual pixel classification step is the major bottleneck of the approach and the iterative training can take hours. However this step uniquely allows the expert user to intuitively train a classifier to automatically integrate the complex and heterogeneous information contained in the multichannel images into normalized images that are well suited for automatized analysis, e.g. using a watershed based segmentation in CellProfiler.

Being fully based on open source software, this scalable pipeline provides an easy extendable basis for a semi-automatized, high throughput analysis of multiplexed tissue images, taking full advantage of the multivariate nature of the data for image segmentation.

4 Discussion

The presented workflow allows for a high throughput semi-supervised image analysis of highly multiplexed images. The development version of this workflow has already been used for the HistoCAT publication, where it was used to segment 49 IMC images and was shown to yield biologically reasonable segmentation results [9]. Internally the approach was already used for datasets with more than 800 images, showing the scalability of the approach.

While supervised pixel classification provides an excellent framework to integrate complex image pixel information into biological relevant classes, it has also several drawbacks. Needing manual input, the classification might have a significant expert bias. This was quantified as part of the HistoCAT paper and the results showed no strong dependency of the analysis results based on segmentations of different users [9].

In particular low resolution imaging techniques such as IMC, which has an xy resolution of 1um and a routinely used cut thickness of 5 um, a pixel might contain parts of nuclear as well as cytoplasmic regions. These overlaps are often not obvious from a single channel, but can e.g. seen by looking at nuclear as well as cytoplasmic channels simultaneously. As a result the probability map representation of the nuclear class is often smaller than one would judge by the nuclear signal alone. While this has favorable properties for declumping of nuclei during the segmentation, nuclear segmentation will thus often lead to nuclear masks that do not contain the complete nuclear signal. We argue that this is mostly due to the physically overlapping of pixel classes and thus is rather a problem of the low resolution used than this approach. We thus suggest that the analysis should be mainly done on the expanded cell level.

Another issue specific to the low IMC image resolution is that it only allows subcellular resolution by discriminating between nuclear and cytoplasmic/membrane pixels, but no separate distinction between cytoplasm and membrane. Correspondingly the separation between neighbouring cells is notoriously difficult. We partially address this problem by doing the pixel classification at a linear interpolated two fold upsampled resolution, which also makes the manual pixel classification easier. When downsampling the segmentation objects two the original resolution we initially remove pixels that would belong to two different cells at two fold resolution, leaving an empty border between neighbouring cells. Due to the lack of membrane specific membrane separation between neighbouring cells we argue that these pixels would be just randomly assigned to one or the other cells. Depending on the problem it might thus be reasonable to work with the mask with a gap or close it using another expansion step. Once the quality and resolution of IMC images improved we highly recommend to correspondingly acknowledge the membrane structures specifically to alleviate these problems.

A natural limitation of training a pixel classifier, is that the classifier can not be easily transferred to new datasets, except if they have the same markers measured. However given that the same markers used for classification are measured, we observe a transferability of a classifier from one dataset to another. However we recommend to calculate the uncertainty maps of the predicted images and screen for regions of high uncertainty, which should then be trained specifically. We speculate that in the future, provided a variable enough training set and the use of a different classification implementation, a classifier trained on a single large dataset might be reused on new datasets with only minor needs for retraining, making the pipeline even more automatized.

Being solely based on mature open source software, extending the pipeline is easy, mainly due to the modular structure of CellProfiler and Ilastik, which forms the core of the pipeline. Obvious extensions of the approach is to combine it with tissue level segmentation masks. For example Ilastik can be used to classify the tissue into stroma and tumor areas, similarly than used in the AQUA approach [8]. Measuring the resulting probability masks with the cellular segmentation masks with CellProfiler allows to further integrate this level of information.

Being build on CellProfiler, the output of this pipeline is segmentation masks and single cell information in the standardized CellProfiler output format. This forms a solid basis for more complex analysis e.g. with the HistoCAT software for multiplexed image analysis or custom R and python scripts, or to import the data in a standardized database.

4.1 Comparison to existing workflows

Classically single cell segmentation is done by using a nuclear marker to identify the cell center and then applying watershed or similar algorithms to identify the outline of a cell using a membrane

marker [2]. Correspondingly recent publications of multiplexed imaging approaches perform a simple segmentation based on a nuclear and often another membrane channel, ignoring the additional information encoded in the other acquired channels [6, 1]. A notable exception is the approach suggested by Schueffler et al [11]. They identify multiple membrane channels in multiplexed images based on the Spearman correlation with a bait, user defined membrane channel. Then they create a new 'meta' membrane channel, calculating a weighted sum of the channels. The weighting of the channels is optimized by optimizing an 'segmentation score', taking into account the overlap of the masks with a user defined nuclear as well as membrane channel as well as the expected number of cells. While taking information from more than one channel into account, this approach a) largely depends on the validity of the segmentation score, b) a single, good bait membrane channel that is expressed in most of the cells of interest, c) focuses heavily on the membrane channel identification. d) does not incorporate signal gradient and texture features. Observing a better scalability and visual performance, a real quantitative comparison of the approaches is challenging and corresponding experiments are currently being performed. Conceptually we argue that the supervised machine learning based approach is more flexible and makes better use of the segmentation relevant information incorporated in the channels. Notably not only information about the membrane but also about nuclear identity of the pixels is extracted from the masks. For example in cases were the nuclear signal is very weak, the classifier will still identify a nucleus based on the absence of non-nuclear markers.

5 Conclusion

We present a modular, scalable and flexible segmentation pipeline particularly suited for highly multiplexed images largely based on the combination of the CellProfiler and Ilastik software. Enabling an intuitive expert based classification with the flexible machine learning algorithm, allows to distill pixel class information using all the available channel data and results in standardized probability maps. Altogether the presented workflow allows a high throughput processing of hundreds of multiplexed tissue images and thus forms a solid basis for a standardized, open source data analysis.

References

[1] Michael Angelo, Sean C Bendall, Rachel Finck, Matthew B Hale, Chuck Hitzman, Alexander D Borowsky, Richard M Levenson, John B Lowe, Scot D Liu, Shuchun Zhao, Yasodha Natkunam, and Garry P Nolan. Multiplexed ion beam imaging of human breast tumors. 20(4):436–42.

[2] Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David a Guertin, Joo Han Chang, Robert a Lindquist, Jason Moffat, Polina Golland, and David M Sabatini. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. 7(10):R100.

[3] Charlotte Giesen, Hao a O Wang, Denis Schapiro, Nevena Zivanovic, Andrea Jacobs, Bodo Hattendorf, Peter J Schüffler, Daniel Grolimund, Joachim M Buhmann, Simone Brandt, Zsuzsanna Varga, Peter J Wild, Detlef Günther, and Bernd Bodenmiller. Highly multiplexed imaging of tumor tissues with subcellular resolution by mass cytometry. 11(4):417–22.

[4] Ilya G Goldberg, Chris Allan, Jean-Marie Burel, Doug Creager, Andrea Falconi, Harry Hochheiser, Josiah Johnston, Jeff Mellen, Peter K Sorger, and Jason R Swedlow. The open microscopy environment (OME) data model and XML file: open tools for informatics and quantitative analysis in biological imaging. 6(5):R47.

[5] Ranveer Joyseeree, Oscar Alfonso Jiménez del Toro, and Henning Müller. Using probability maps for multi–organ automatic segmentation. In *International MICCAI Workshop on Medical Computer Vision*, pages 222–228. Springer.

[6] Jia-Ren Lin, Mohammad Fallahi-Sichani, and Peter K Sorger. Highly multiplexed imaging of single cells using CycIF, a high-throughput cyclic immunofluorescence method. 6:1–7.

[7] David J. Logan, Jing Shan, Sangeeta N. Bhatia, and Anne E. Carpenter. Quantifying co-cultured cell phenotypes in high-throughput using pixel-based classification. 96:6–11.

[8] A. McCabe, M. Dolled-Filhart, R. L. Camp, and D. L. Rimm. Automated quantitative analysis (AQUA) of in situ protein expression, antibody concentration, and prognosis. 97(24):1808–1815.

[9] Denis Schapiro, Hartland W Jackson, Swetha Raghuraman, Jana R Fischer, Vito R T Zanotelli, Daniel Schulz, Charlotte Giesen, Raúl Catena, Zsuzsanna Varga, and Bernd Bodenmiller. histoCAT: analysis of cell phenotypes and interactions in multiplex image cytometry data. 14(9):873–876.

[10] P. J. Schüffler, D. Schapiro, C. Giesen, H.A.O. Wang, B. Bodenmiller, and J. M Buhmann. Single cell segmentation with watersheds on highly multiplexed images. In *12th European Congress on Digital Pathology*.

[11] Peter J. Schüffler, Denis Schapiro, Charlotte Giesen, Hao A. O. Wang, Bernd Bodenmiller, and Joachim M. Buhmann. Automatic single cell segmentation on highly multiplexed tissue images: Automatic single cell segmentation on highly multiplexed tissue images. 87(10):936–942.

[12] Christoph Sommer, Luca Fiaschi, Fred A. Hamprecht, and Daniel W. Gerlich. Learning-based mitotic cell detection in histopathological images. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2306–2309. IEEE.

[13] Christoph Sommer, Christoph Straehle, Ullrich Koethe, and Fred A. Hamprecht. Ilastik: Interactive learning and segmentation toolkit. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 230–233. IEEE.