**Course**: BINP37

**Credits**: 15 (10 weeks)

**Project Name**: Development of a 2D nuclei segmentation model for the detection and visualisation of poly-aneuploid cancer cells in pre-treatment breast cancer tissue images

**Student**: Elijah Ugoh

**Email**: ugohelijah@gmail.com

**Supervisor(s)**: Emma Hammarlund, Tissue Development and Evolution (TiDE) Group, Lund University Cancer Centre, BMC B11, Lund University, Sweden.

**Email**: [emma.hammarlund@med.lu.se](mailto:emma.hammarlund@med.lu.se)

Arthur Boffelli Castro, Tissue Development and Evolution (TiDE), BMC B11, Lund University

**Email**: [arthur.boffelli\_castro@med.lu.se](mailto:arthur.boffelli_castro@med.lu.se)

# Abstract

Poly-Aneuploid Cancer Cells (PACCs) are large cancer cells that are present in many different types of solid cancer tissues and have multiplied their aneuploid genome several times, becoming larger than normal-sized cancer cells. Recent studies have shown that, in vitro, they are capable of surviving toxic levels of therapies, remain in a prolonged senescent-like state, and subsequently propagate tumors with therapy-resistant progenies. Ongoing studies correlate the presence of PACCs in primary cancer tissue with poorer outcomes and a higher risk of tumor recurrence in breast cancer patients with the SweBCG-RT91 cohort. Currently, the manual analysis of PACCs in tumors is time-consuming and subjective. This project developed a machine learning model that can automate the detection and segmentation of PACCs in pre-treatment breast cancer tissue samples based on their abnormally-large nuclei, allowing for more detailed and accurate analysis of the cells. The project focused on training a new model on Tissue Micro-Array (TMA) images stained with EpCAM-DAB. The data was obtained from 1190 pre-treatment breast cancer patients. Supervised machine learning method was used to train the model with the StarDist, a Python-based deep-learning package for object detection. QuPath, a software for digital pathology and whole slide image analysis, was used for ground truth annotations of the images and testing the model after training. Compared to the pre-trained H&E model provided by StarDist, our custom-trained model is more sensitive at different detection probability thresholds and does not detect wrong large objects. The model produces reliably precise nuclei detection and segmentation, making it a more efficient option for analyzing PACCs in tumors. It can be adapted for use in existing pipelines for automatic detection of PACCs in breast cancer studies.

**Availability and Implementation:** This project was executed in Python, Groovy, and Bash terminal. The Notebook containing the entire training code and other useful resources is freely [available on GitHub](https://github.com/Elijah-Ugoh/Model-Training-For-Nuclei-Segmentation/blob/master/Model_Training_Script.md).

# Introduction

Breast cancer constitutes one of the major global health concerns today, and now accounts for the most commonly diagnosed cancer in the world (Wilkinson and Gathani, 2022). After decades of research, major advances have been made in breast cancer treatment. Wang and Wu (2023) concluded that surgery-based local and systemic treatments remain the standard care approach for early breast cancer. Chemotherapy-based systemic treatments are typically used for metastatic breast cancer while surgery is only used for early invasive breast cancer in selected patients. Most new cases will have a 10-year survival rate of approximately 85% or higher, if treated with surgery only (Sopik *et al.,* 2023). The emergence of targeted therapy and immunotherapy has also helped to improve the outcomes of treatments in early and metastatic breast cancer. However, the success of these therapies is still limited due to tumor recurrence. Tumors use a variety of molecular mechanisms to enhance resistance to therapy.

One mechanism thought to be responsible for tumor recurrence is the presence of cells in the poly‐aneuploid cancer cell (PACC) state, which has been linked to the resistance of chemotherapy and radiotherapy in prostate cancer (Carroll *et al.,* 2024).

Pienta and colleagues (2020) demonstrated that cells in the PACCs state play a pivotal role as potent reservoirs of heritable variation that facilitates rapid evolution, speciation, and adaptive response to environmental changes in cancer cells. These PACCs enable evolutionary rescue, development of therapy resistance, and metastatic progression, which can affect the patient’s prognosis and survival. By definition, PACCs are non-dividing aneuploid cells formed in response to stress. PACCs are cancer cells that have multiplied their aneuploid genome by endocycling, and become larger, often with a misshapen large nucleus, and have been shown to have increased metastatic potential (Mallin *et al*., 2023). These cells have previously been considered unimportant as they are believed to be destined to mitotic collapse (Pienta *et al.,* 2020), but in recent studies, PACCs are observed in vitro to increase in numbers during therapy, remain in a prolonged senescent-like state, and eventually divide and produce progeny (Carroll *et al.,* 2024). To better understand how the presence of PACCs in primary cancer tissue with a poorer outcome correlates with a higher risk of tumor recurrence in breast cancer patients, these tissues must be studied pre-treatment and post-treatment. However, the large amount of Tissue Micro-Array (TMA) images obtained from breast cancer patients makes a manual analysis time-consuming and the detection of PACCs subjective, since it largely depends on pathologists’ interpretation of large nuclei.

StarDist developed a nuclei detection pipeline and provides pretrained deep learning models that effectively detect and segment nuclei in fluorescent and brightfield (Hematoxylin and Eosin) stained tissue images (Schmidt *et al.,* 2018). StarDist is a deep learning-based nuclei and cell detection and segmentation program for 2D and 3D microscopy images, and can be used for cell/nuclei segmentation in densely-packed tissues. However, the pre-trained StarDist models are optimized for H&E (Hematoxylin and Eosin) and fluorescent-stained images. While the StarDist H&E model works when used on DAB (3,3’-diamobenzidine)-stained images, it is limited in accuracy and precision, and misses a significant number of nuclei. DAB staining is often used in cancer research because of its high specificity for binding tumor in cells, which is characterized by the presence of the Epithelial Cell Adhesion Molecule (EpCAM) biomarker (Patriarca *et al.,* 2012). This project, therefore, focused on training a unique deep-learning model for nuclei detection in breast cancer TMA images which have been stained with EpCAM-DAB using the StarDist deep-learning package in Python. StarDist works well for segmenting all kinds of blob-like objects, especially roundish objects like cells and nuclei that have a star-convex shape. It uses a framework trained with a convolutional neural network (CNN) that can predict every pixel in a polygon and segment the cells or nuclei at that position with high accuracy. The trained model can be applied for the automatic detection of PACCs in cancer tissue cores.

# Materials and methods

## Preliminary Data Exploration and Preparation

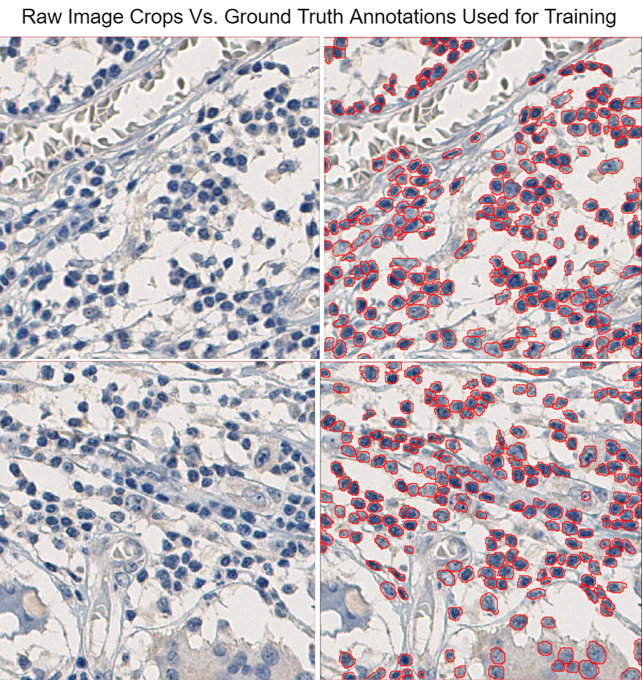
The data used in this project is brightfield tissue micro-array (TMA) images stained using EpCAM-DAB and obtained from 1190 pre-treatment breast cancer patients (Killander *et al*., 2016). These images were preliminarily examined by pathologists and confirmed to contain poly-aneuploid cancer cells. The data comprise a total of 20 high-resolution TMA images, with each microarray containing between 90-130 different tissues cores embedded at defined array coordinates. Data from all patients are randomly distributed in the TMA, with one or two cores per patient. The data used for training the model was extracted from these cores. But first, we split each TMA into its constituent tissue cores using QuPath (Bankhead *et al*., 2017), an open-source software for digital pathology image analysis. QuPath (v0.5.1) offers a user-friendly interface, with many tools provided within the software. However, it also allows scripting in Groovy language to reach even more functionalities. A total of 1391 tissue cores were extracted from 11 of the available TMA images. After removing 144 blank cores, 1247 cores remained for further analysis.

### Creating Crops/Training Data

After de-arraying the tissue cores, another Groovy script is used to automate the random selection of 20 tissue cores from the dataset, also within QuPath. This randomization ensures that the images selected for training the model are diverse and representative of the entire dataset, while the automation makes it easy to handle the large dataset. With 20 tissue cores randomly selected from all the de-arrayed TMA images, crops of these cores were further extracted for annotation as ground truth labels for training using the StarDist deep-learning package in Python. These crops are smaller sections or sub-regions of the cores and were generated randomly using QuPath. The minimum recommended dimension for StarDist is 128 X 128 pixels squared (Bankhead, 2022). We created 3 crops measuring 300 x 300 pixels squared in dimension from each core, to ensure that each crop contains enough cells/nuclei for annotation and captures different morphologies and regions in the tissue.

### Ground Truth Annotation

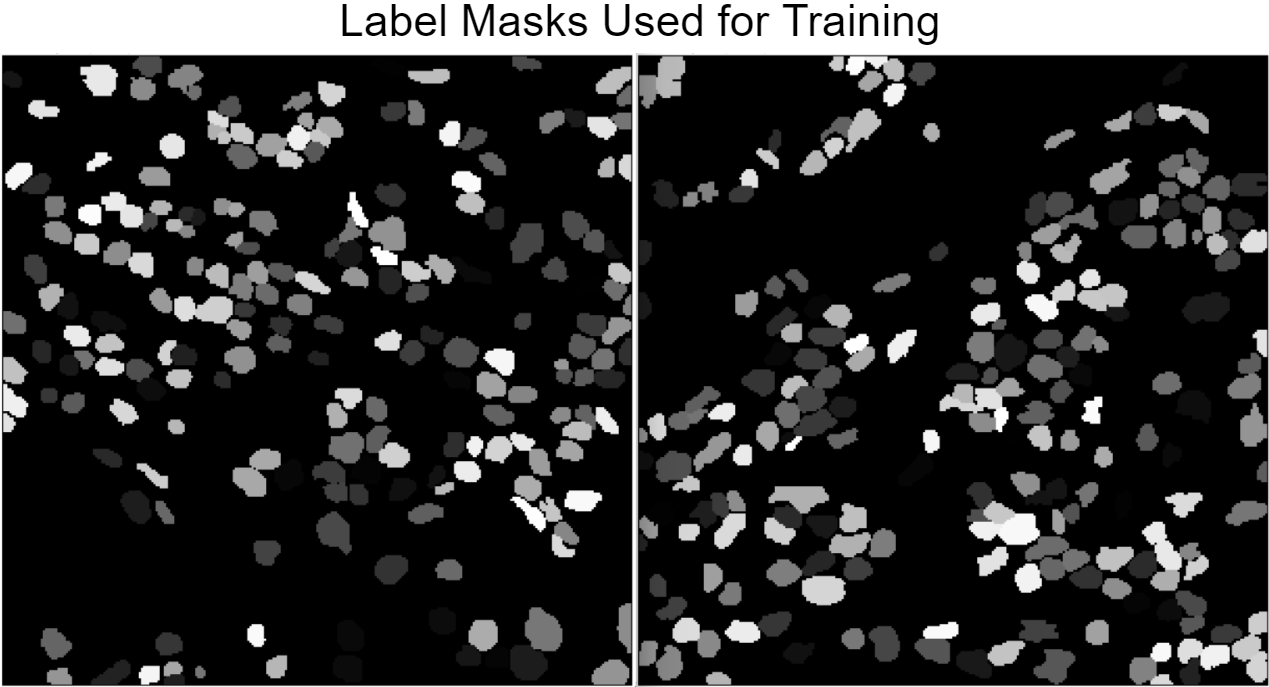
The ground truth labels were created by manually annotating the nuclei in all the crops as the regions of interest (ROI) (Fig.1). A Groovy script is used to export the annotations as masks (Fig. 2), which together with the unlabeled/raw images were used for training the nuclei detection model.



***Fig. 1. Raw images of crops (300 X 300 Pixels) obtained from the TMA cores (left) and annotations (ground truth labels) of crops****. Crops were generated randomly using QuPath. All annotations were created manually using QuPath and exported as masks. The masks are used alongside the original (unannotated images) for training.*

StarDist recommends a minimum of 10 crops for training. In this case, 51 crops (6 test images, 7 validation images, and 38 training images) are used to provide sufficient training and test data. This is an important step in a supervised learning procedure, where the model is trained to make accurate predictions using labelled data (ground truths) with a defined target (Bankhead, 2022).

Similar to other segmentation networks, the StarDist package uses the actual image and ground truth (masks) to learn a classification task. As a supervised machine learning, the label masks or ground truth tells the network the specific class every pixel in the image belongs to. This way, the network learns the exact features of the nuclei in each image based on its corresponding mask, while ignoring any other feature on the images or objects. The images and their corresponding masks were saved in separate folders, but with the exact same file names. This ensures accurate matching of each image to its corresponding label mask during training.



***Fig. 2. Corresponding masks of images used for training the model.*** *All masks were generated in QuPath after each complete annotation. In these samples, the image brightness has been adjusted to make the annotations visible. These masks were used for training alongside the raw images.*

The final step in the data preparation process is the splitting of the dataset into training and test images. This is a standard procedure in machine learning, and depending on the size of the dataset, the ratio of training-test data can range from 80:20 to 90:10. With the aid of a custom Python script, the dataset containing 51 images and their corresponding masks are split into 6 test and 45 train subsets using an 88:12 ratio. During the training, the StarDist pipeline automatically further splits the training dataset into training and validation images. The network uses the validation data to automatically improve the detection accuracy of the trained model during the learning process.

## Model Training Using StarDist

Considering that star-convex polygons are an ideal shape representation to accurately localize cell nuclei even in densely-packed tissues with overlapping nuclei (Schmidt *et al*. 2018), we opted for StarDist for training our model. StarDist supports detection in both 2D and 3D images when trained with 2D and 3D images, respectively.

### Programmes/Packages Installation

All annotations and predictions were run in QuPath (v0.5.1) (Bankhead *et al*., 2017). The Model training script is run solely in a Google Colab Notebook (0.0.1a2) as individual code blocks. The training data (raw images and labelled masks) are imported into the Colab Notebook, where the StarDist model training pipeline is executed in Python (3.10.12) (Schmidt *et al*., 2018; Weigert *et al*., 2020; Weigert & Schmidt, 2022). Before running the script, StarDist (0.9.1) and TensorFlow (pre-installed in Colab) were installed. TensorFlow is, a versatile, open-source machine learning (ML) library, typically used for building and deploying ML models. It also includes TensorBoard, which provides interactive visualisation while the model is being trained, allowing us to track and visualize metrics, such as epoch loss, the model graph (ops and layers), histograms of weights, biases as they change over time. The software Git (2.43.2) was used for the version control and GitHub for public sharing of all scripts and processes used in this project.

### Data Augmentation and Model Optimisation

The pipeline also employs data augmentation to artificially create more training data by transforming existing images and masks into different geometrical, yet plausible forms. The augmenter function applies random rotations, flips, and intensity changes to the images, which are typically sensible for (2D) microscopy images, thereby making the trained model able to detect varying instances or forms of the original image. With the validation images, the model is constantly tested for accuracy and optimized during training. This helps improve the model's performance and accuracy on unseen data. Finally, the optimized threshold values were saved to disk so that they are automatically loaded with the model weights. This ensures consistent and reliable performance of the model across different unseen datasets and applications.

### Conversion to ONNX and Prediction

The trained model is exported as a zip file from Colab and saved locally. Because of compatibility problems when running the trained StarDist model in QuPath or other image processing software like ImageJ/Fiji (Schindelin *et al.,* 2012), the StarDist model requires post-processing to produce an instance cell/nuclei segmentation. This problem is common when using TensorFlow 2.0 or later in the training. To resolve the issue, the trained model is loaded in a Python environment with TensorFlow v1.15.5 installed, and then exported it again. To use the trained model in QuPath, we further converted it to ONNX (Open Neural Network Exchange) format. This conversion allows for interoperability between different deep learning frameworks, in this case, enabling the model trained in TensorFlow to be run in other frameworks, that support ONNX, with minimal effort. To achieve this, tf2onnx (v1.16.1) was installed in the same Python environment, and then used to convert the trained model into ONNX format, which was later easily loaded into QuPath, using a StarDist extension. The model training was executed twice, with subsequently higher number of training data. This was done to ascertain the minimum amount of training data required to obtain optimum nuclei segmentation from the model and to avoid overfitting/overtraining (Charilaou & Battat, 2022).

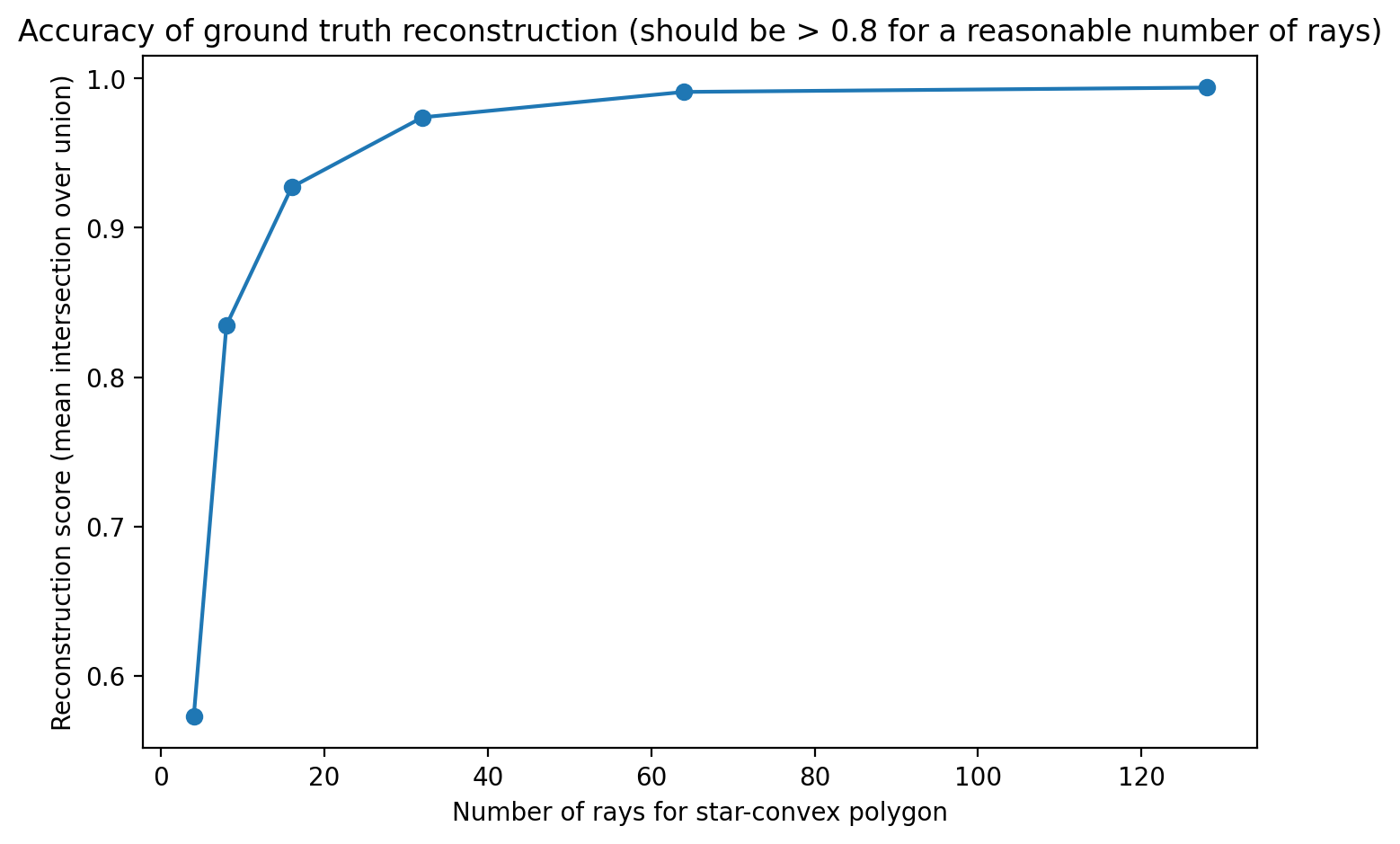
The model was tested on several cores not used during training using a Groovy script executed in QuPath for nuclei segmentation. Finally, the results were visually compared to previous analysis performed using the publicly available H&E pre-trained model.

# Results

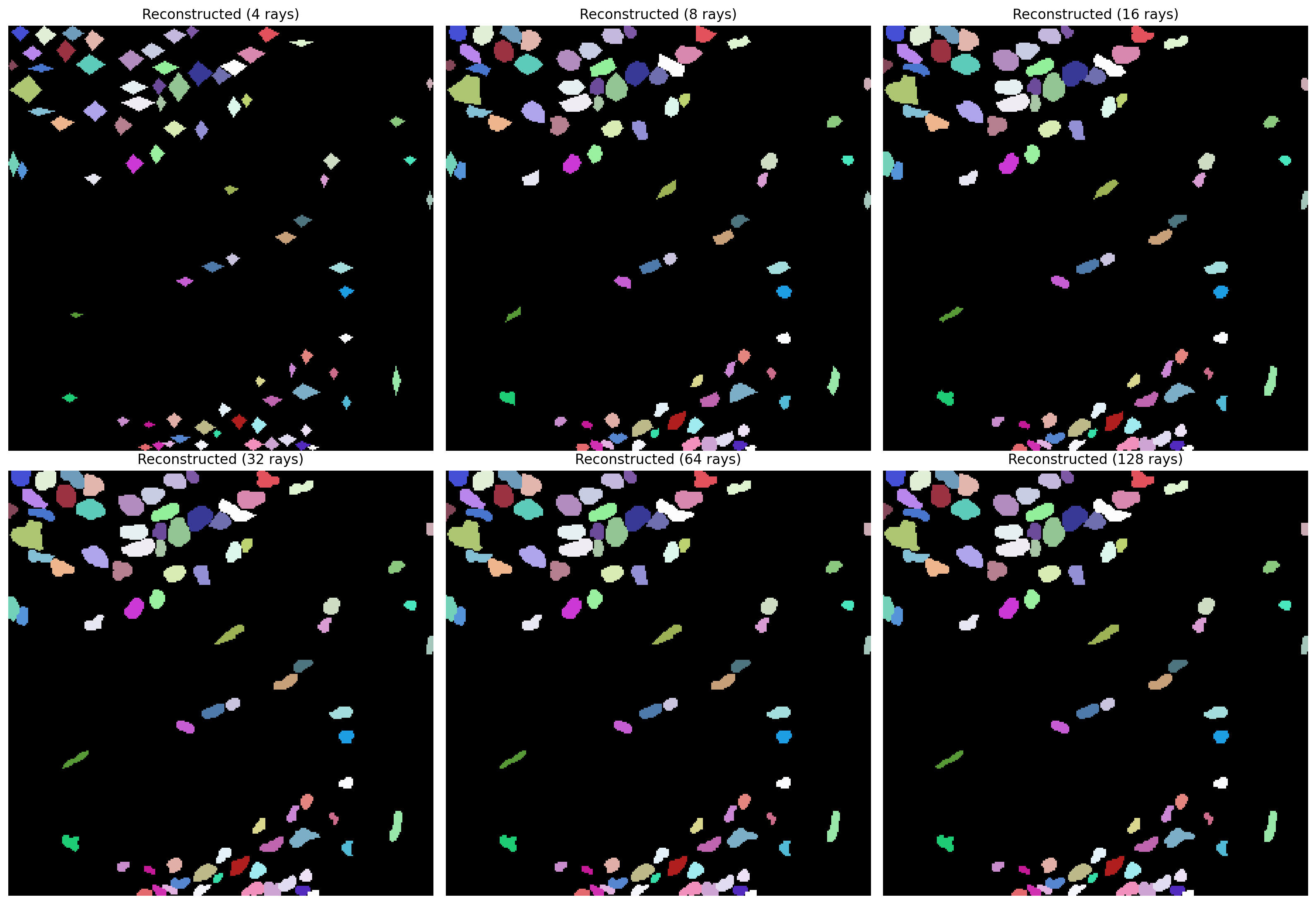
With only 10 annotated crops, the model’s segmentation accuracy was relatively high compared to the pre-trained H&E model provided by StarDist, but not sufficiently accurate for reliable downstream analysis. With 28 more annotations, the epoch loss during training was lower and the detection/segmentation accuracy was significantly better. Nuclei predictions were executed in both Fiji and QuPath.

## Data Fitness for StarDist

An important step before the training is to check if the data loaded into the model training pipeline is actually ideal for deep learning using the StarDist package. A fundamental principle used by StarDist is to train the model to predict the distances to an object’s (nucleus) boundary along a fixed set of rays and object probabilities. So, we evaluated the performance of the training data with the StarDist model using different numbers of rays (by reconstructing the ground truth labels and calculating the mean Intersection over Union scores for each configuration or reconstruction) (Fig.4). Intersection over Union (IoU) is a model performance metric that we used to evaluate the accuracy of both the ground truth annotations and segmentations the model will produce based on those annotations (Rahman & Wang, 2016). The code for this analysis is included in the StarDist data preparation script on GitHub. A plot of the reconstruction scores (IoU) against number of rays (Fig.3) shows the fitness of our data for training with the StarDist deep-learning.



***Fig. 3. Plot of reconstruction score (Intersection over Union).*** *This plot shows the relationship between different number of rays used in the StarDist model and the reconstruction score, which represents the mean Intersection over Union (IoU) between the original labels and the reconstructed labels. The IoU quantifies the overlap between the predicted bounding box and the ground truth bounding box or annotated region in the training data (ROI). Here, the accuracy of the ground truth reconstruction is greater than 0.8 for at least 16 rays in a star-convex polygon.*



***Fig. 4. Reconstruction of an example image from the training dataset using various number of rays.*** *The plots above show that the minimum reasonable number of arrays that yields a high mean IoU score between the original labels and the reconstructed labels is 16 rays. 32, 64, and 128 rays produce similar and more accurate results. Reconstructions with 4 and 8 rays are the least accurate.*

## Model Accuracy as a Function of IoU Thresholds

After training and optimizing the model, it was further evaluated against the validation dataset at various IoU thresholds to determine its performance in nuclei segmentation. The matching statistics (Fig.5. a) include critical metrics like the precision of detections, mean true score, accuracy of prediction, F1 score, panoptic quality, mean matched score, false positives (fp), true positives (tp), and false negative (fn) detections of the model. The closer the IoU score is to 1 (conventionally ranges from 0 to 1), the better the prediction. In practice, an IoU score of 0.5 is often used as a threshold for good object detection.



***Fig. 5. Evaluating the accuracy of the trained StarDist 2D model.*** *(a) Matching statistics is computed for the validation data at various IoU thresholds (ranging from 0.1 to 0.9) and plotted as a function of the IoU threshold, τ. (b) The numbers of true/false positives/negatives are plotted as a function of the IoU threshold, τ. The plots provide insights into how well the model will perform under different evaluation criteria.* *The intersection of tp and fp at an IoU threshold of about 0.72 indicates that tp = fp, and at this IoU threshold, the model is making as many correct positive predictions as it is making incorrect positive predictions.*

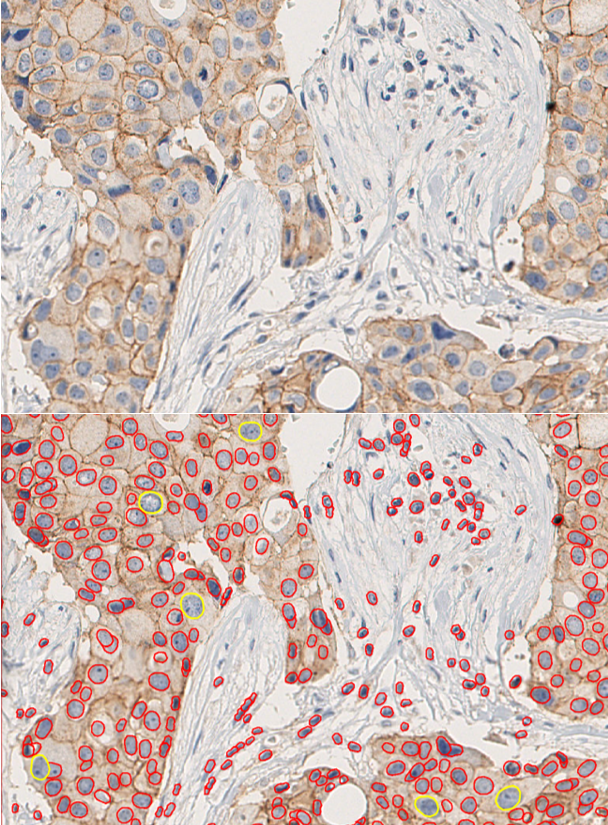
An IoU score (τ) of 0.5 or higher indicates that the predicted bounding box sufficiently overlaps with or captures the object of interest, even if it might not perfectly align with the ground truth. At τ = 0.5 (Fig.5. b), the segmentation precision of the trained model is 0.82, F1 score is 0.77, and its accuracy is 0.63, which is an acceptable performance level at this IoU threshold. From Fig.5. b, we can observe that the reliability of the model drops significantly at τ = 0.7 and beyond.

**Table 1: Breakdown of Model Matching Statistics and Values at IoU threshold (τ) of 0.5**

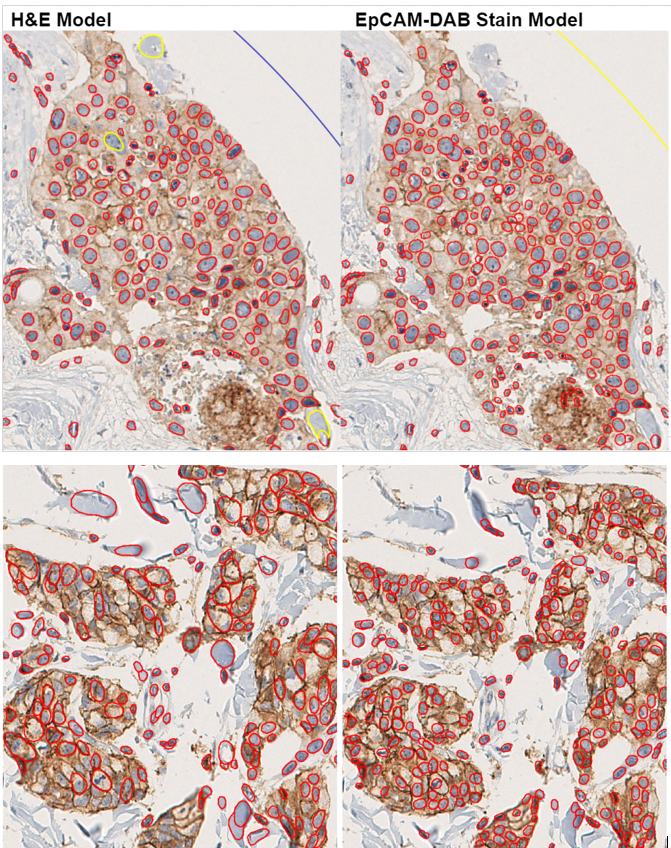
|  |  |  |
| --- | --- | --- |
| **Statistic** | **Meaning** | **Value** |
| **Threshold (τ)** | The threshold value used for determining true positives, false positives, and false negatives. Any IoU value greater than or equal to the threshold is considered a positive match. | 0.5 |
| **True Positives (tp)** | The number of correctly detected objects or instances by the model, where both the predicted and ground truth objects have sufficient overlap (IoU ≥ threshold) | 422 |
| **False Positives (fp)** | The number of objects or instances predicted by the model that do not have sufficient overlap (IoU < threshold) with any ground truth objects | 92 |
| **False Negatives (fn)** | The number of ground truth objects or instances that were not detected by the model | 159 |
| **Precision** | Measures the proportion of correctly predicted positive instances out of all instances predicted as positive by the model. | 0.8210 |
| **Recall** | Also known as sensitivity or true positive rate, it measures the proportion of correctly predicted positive instances out of all true positive instances | 0.7263 |
| **Accuracy** | Measures the overall correctness of the predictions made by the model | 0.6270 |
| **F1 Score** | The harmonic mean of precision and recall, providing a balance between the two metrics | 0.7708 |
| **n\_true** | The total number of ground truth objects or instances | 581 |
| **n\_pred** | The total number of objects or instances predicted by the model | 514 |
| **Mean True Score** | The mean IoU score computed for all ground truth instances | 0.5438 |
| **Mean Matched Score** | The mean IoU score computed for all matched (correctly detected) instances | 0.7487 |
| **Panoptic Quality** | A metric that combines both segmentation and detection accuracy, providing a unified measure of model performance | 0.5770 |

## Detection and Segmentation Results on Whole TMA Cores

By visual inspection, the trained model produces highly precise object segmentation in QuPath. A QuPath extension for StarDist models allows users to import the trained model into QuPath and use it to segment the nuclei in any EpCAM-DAB-stained tissue image, as well as detect any PACCs present. Users must first load the TMA images into QuPath, then run the model using a Groovy script to detect and segment all nuclei. The model works regardless of the image size. However, to achieve optimum segmentation, the pixel size must be adjusted accordingly in script depending on the image size. Where present, the poly-aneuploid cancer cell (PACC) nuclei are also detected (Fig. 6). The script also includes an adjustable probability threshold, which is used during the detection phase to determine whether or not a pixel or region in the input image is classified as a nucleus by the model. This threshold directly influences the number of nuclei detected by the model, affecting the trade-off between sensitivity and specificity in the detection process. The trained model also performs better that pre-trained H&E model provided by StarDist in a number of cases. At a detection probability threshold of 0.3, the trained model makes 3,989 detections while the H&E model makes 3,053 detections, indicating lower sensitivity. While both models make inaccurate predictions in some cases (depending on the set threshold and other parameters like the pixel size and normalization percentile), our trained model does not detect wrong PACC objects, which is observed in the H&E (Fig. 6 and Fig. 7).



***Fig. 6. Section of instance segmentation produced by the trained model on a sample EpCAM-DAB-stained tissue image.*** *The upper section of the image shows the raw image without any detections. The lower part shows the same image after the model has been run on it. The red marked regions are the regular cancer nuclei, while the yellow-marked nuclei are the PACCs (characterized by their larger sizes).*



***Fig. 7. Detection results produced by the StarDist H&E model and our custom-trained model on two sample tissue cores.*** *The H&E model detects wrong large objects, and in some cases, is not able to segment nuclei accurately. The trained model produces better segmentation, and though it also detects objects that are not nuclei, the objects do not fall into the PACCs category. This makes the trained also better at detecting PACCs. In the lower half of the images, there is a clear difference in the segmentation results.*

The measurement feature in QuPath allows the user to zoom in on each detected PACC and view the area (in µm^2) and several other parameters like detection count, length, circularity, and solidity. This data can also be saved or exported as a text file and used for further analysis.

# Discussion

In this project, EpCAM-DAB-stained images obtained from pre-treatment breasts cancer patients were used to train a StarDist deep-learning model for nuclei segmentation. The model can be used for detecting and segmenting nuclei, including poly-aneuploid cancer cells (PACCs), present in tissue cores, enabling easier and more accurate analysis of cells in PACC state in breast cancer studies. When run in QuPath or Fiji, the model allows the user to adjust the detection probability threshold to reduce/increase the sensitivity and specificity of the detections. Also, depending on the size of the input images, the user can adjust the pixel size (only in QuPath), to ensure accurate segmentation of the nuclei detected. While this project focuses on using QuPath to run the model, it also works with another pathological image analysis tool like Fiji. However, Fiji has a less friendly user interface, as it returns the predictions as a separate output (label image only), the ROI Manager, or both. QuPath shows all of these in one simple, easy-to-manipulate interface that’s ideal for both small and larger images. QuPath proved to be most flexible for accessing the precision and accuracy of nuclei detection and segmentation at a glance or in details.

There are other notable deep learning frameworks for nuclei detection and segmentation, such as Cellpose (Stringer *et al*., 2021). But, StarDist outperforms Cellpose on large datasets with densely-packed objects (Kleinberg *et al*., 2022). StarDist also does a better job at separating overlapping nuclei and outperforms Cellpose with a low signal-to-background ratio of intensity. However, Cellpose appears to be a great fit for 3D cell segmentation (Kleinberg *et al*., 2022).

While we used Google Colab Notebook, the model training code can also be run in Jupyter Notebook and other IDEs or initiated from the terminal. Colab works well for machine learning, provides access to GPUs, has a simple interactive interface with TensorFlow, which is a major dependency for training a StarDist model, pre-installed. A major drawback in this project is the time-consuming process of annotating the training data. Automating this process can significantly improve efficiency and reduce the completion time, allowing for more iterations for optimization of the model. However, as there isn’t a suitable automatic annotation method to achieve this yet, QuPath was used for manual annotation to ensure the model was trained on accurately annotated ground truth labels. To improve the future performance and application of the model, it can be trained on broader data from other types of cancer tissues. This way, it could be useful for broad-case applications in studying PACCs.

# Acknowledgements

I would like to thank my supervisors, Emma Hammarlund, for the opportunity to join their research group at the Lund Cancer Centre, and Arthur Boffelli Castro, for the opportunity to work on this research project, as well as for the feedback and guidance throughout the development of the model.

# References

Wilkinson, L., & Gathani, T. (2022). Understanding breast cancer as a global health concern. *The British Journal of Radiology, 95*(1130), 20211033. <https://doi.org/10.1259/bjr.20211033>

Wang, J., & Wu, S. G. (2023). Breast cancer: An overview of current therapeutic strategies, challenges, and perspectives. *Breast Cancer (Dove Medical Press)*, 15, 721–730. <https://doi.org/10.2147/BCTT.S432526>

Sopik, V., Lim, D., Sun, P., & Narod, S. A. (2023). Prognosis after local recurrence in patients with early-stage breast cancer treated without chemotherapy. *Current Oncology (Toronto, Ont.),* 30(4), 3829–3844. <https://doi.org/10.3390/curroncol30040290>

Mallin, M. M., Kim, N., Choudhury, M. I., Lee, S. J., An, S. S., Sun, S. X., Konstantopoulos, K., Pienta, K. J., & Amend, S. R. (2023). Cells in the polyaneuploid cancer cell (PACC) state have increased metastatic potential. *Clinical & Experimental Metastasis*, 40(4), 321–338. <https://doi.org/10.1007/s10585-023-10216-8>

Carroll, C., Manaprasertsak, A., Castro, A., van den Bos, H., Spierings, D., Wardenaar, R., Bukkuri, A., Engström, N., Baratchart, E., Yang, M., Biloglav, A., Cornwallis, C., Johansson, B., Hagerling, C., Arsenian-Henriksson, M., Paulsson, K., Amend, S., Mohlin, S., Foijer, F., & Hammarlund, E. (2024). Drug-resilient cancer cell phenotype is acquired via polyploidization associated with early stress response coupled to HIF2α transcriptional regulation. *Cancer Research Communications, 4*. <https://doi.org/10.1158/2767-9764.CRC-23-0396>

Pienta, K. J., Hammarlund, E. U., Axelrod, R., Brown, J. S., & Amend, S. R. (2020). Poly-aneuploid cancer cells promote evolvability, generating lethal cancer. *Evolutionary Applications*, 13(7), 1626–1634. <https://doi.org/10.1111/eva.12929>

Killander, F., Karlsson, P., Anderson, H., Mattsson, J., Holmberg, E., Lundstedt, D., Holmberg, L., & Malmström, P. (2016). No breast cancer subgroup can be spared postoperative radiotherapy after breast-conserving surgery: Fifteen-year results from the Swedish Breast Cancer Group randomised trial, SweBCG 91 RT*. European Journal of Cancer*, 67, 57–65. <https://doi.org/10.1016/j.ejca.2016.08.001>

Schmidt, U., Weigert, M., Broaddus, C., & Myers, G. (2018). Cell detection with star-convex polygons. In A. Frangi, J. Schnabel, C. Davatzikos, C. Alberola-López, & G. Fichtinger (Eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018* (11071). Lecture Notes in Computer Science. Springer, Cham. <https://doi.org/10.1007/978-3-030-00934-2_30>

Patriarca, C., Macchi, R. M., Marschner, A. K., & Mellstedt, H. (2012). Epithelial cell adhesion molecule expression (CD326) in cancer: a short review. *Cancer Treatment Reviews*, 38(1), 68–75. <https://doi.org/10.1016/j.ctrv.2011.04.002>

Bankhead, P., Loughrey, M. B., Fernández, J. A., Dombrowski, Y., McArt, D. G., Dunne, P. D., McQuaid, S., Gray, R. T., Murray, L. J., Coleman, H. G., James, J. A., Salto-Tellez, M., & Hamilton, P. W. (2017). QuPath: Open source software for digital pathology image analysis. *Scientific reports*, 7(1), 16878. <https://doi.org/10.1038/s41598-017-17204-5>

Bankhead, P. (2022). Developing image analysis methods for digital pathology. *The Journal of Pathology*, 257(4), 391-402. <https://doi.org/10.1002/path.5921>.

Stringer, C., Wang, T., Michaelos, M., & Pachitariu, M. (2021). Cellpose: a generalist algorithm for cellular segmentation. *Nature Methods*, 18(1), 100–106. <https://doi.org/10.1038/s41592-020-01018-x>

Kleinberg, G., Wang, S., Comellas, E., Monaghan, J. R., & Shefelbine, S. J. (2022). Usability of deep learning pipelines for 3D nuclei identification with Stardist and Cellpose. *Cells & Development*, 172, 203806. <https://doi.org/10.1016/j.cdev.2022.203806>

Weigert, M., Schmidt, U., Haase, R., Sugawara, K., & Myers, G. (2020). Star-convex polyhedra for 3D object detection and segmentation in microscopy. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 3655–3662. <https://doi.org/10.1109/WACV45572.2020.9093435>

Weigert, M., & Schmidt, U. (2022). Nuclei Instance Segmentation and Classification in Histopathology Images with Stardist*. 2022 IEEE International Symposium on Biomedical Imaging Challenges (ISBIC), Kolkata, India*, 1-4. <https://doi.org/10.1109/ISBIC56247.2022.9854534>

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J. Y., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P., & Cardona, A. (2012). Fiji: an open-source platform for biological-image analysis. *Nature Methods*, 9(7), 676–682. <https://doi.org/10.1038/nmeth.2019>

Charilaou, P., & Battat, R. (2022). Machine learning models and over-fitting considerations. *World Journal of Gastroenterology*, 28(5), 605–607. <https://doi.org/10.3748/wjg.v28.i5.605>

Rahman, M.A., & Wang, Yang. (2016). Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation. In: Bebis, G., et al. (Eds), *Advances in Visual Computing. ISVC 2016*. *Lecture Notes in Computer Science* (10072). Springer, Cham. <https://doi.org/10.1007/978-3-319-50835-1_22>