DATA+AI SUMMIT BY databricks

Distributed Cancer Cell Typing & Tumor Purity

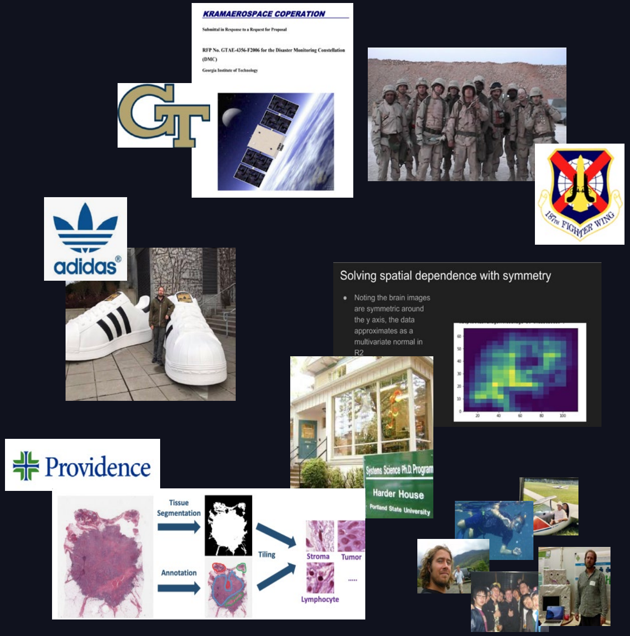
Robert Kramer
Principal Data Scientist - Providence Health
June 12 2024

©2024 Databricks Inc. — All rights reserved

Robert Kramer - About Me

Me

- Principal Data Scientist – Providence Health
- Diverse academic background from aerospace engineering to systems & complexity science
- Current projects include predicting surgery admissions & molecular pathology ML / AI development


KRAMAEROSPACE COPERATION
Submitted in Response to a Request for Proposal
RFP No. GTAE-4356-F2006 for the Disaster Monitoring Constellation (DMC)
Georgia Institute of Technology
GT
187th FIGHTER WING
adidas
Solving spatial dependence with symmetry
Noting the brain images are symmetric around the y axis, the data approximates as a multivariate normal in R2
Providence
Systems Science Ph.D. Program
Harder House
Portland State University
Tissue Segmentation
Annotation
Tiling
Stroma
Tumor
Lymphocyte

©2024 Databricks Inc. — All rights reserved

Providence Overview

122K
CAREGIVERS

38K
NURSES

34K
PHYSICIANS

$2.1B
COMMUNITY BENEFIT

51
HOSPITALS

1000
CLINICS

29M
TOTAL PATIENT VISITS

2.6M
COVERED LIVES

1700+
PUBLISHED RESEARCH STUDIES

1
HEALTH PLAN

18
SUPPORTING HOUSING FACILITIES

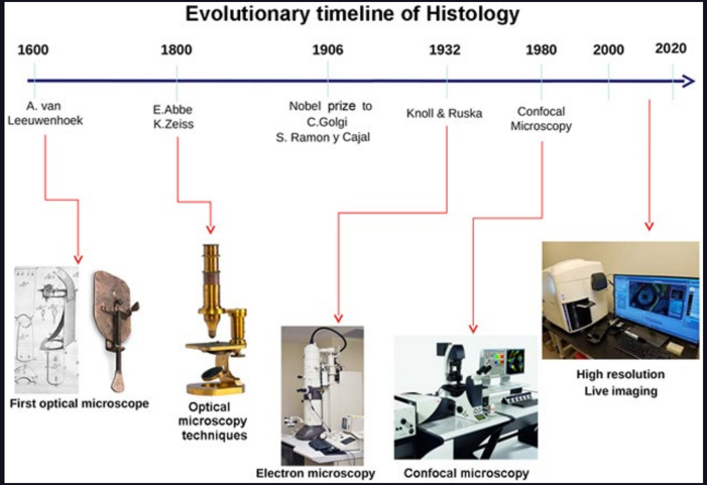
HIGH SCHOOL NURSING SCHOOLS & UNIVERSITY

©2024 Databricks Inc. — All rights reserved

Integrating AL/ML into Histology

Cell typing is the first step to understand the tumor microenvironment

- Tumor Purity = $\frac{\mathit{Tumor\ Cells}}{\mathit{Total\ Cells}}$
- **Consistency and Objectivity:** Traditional manual estimates of tumor purity are subjective and often inconsistent.
- **Quality Control:** Ensures accurate analysis of tumor samples.
- **Relating Tumor Environment to Genetic Markers:** Tumor purity is critical for understanding the tumor microenvironment, which is linked to genetic biomarkers and patient outcomes.



Mazzarini M, Falchi M, Bani D, Migliaccio AR. Evolution and new frontiers of histology in bio-medical research. Microsc Res Tech. 2021 Feb;84(2):217-237. doi: 10.1002/jemt.23579. Epub 2020 Sep 11. PMID: 32915487; PMCID: PMC8103384.

©2024 Databricks Inc. — All rights reserved

Going From 0 to 1

Cell Typing is the “model organism” of histology imaging AI at Providence

- **Foundational Design Patterns:** Key for future AI applications
- **Histologic Imaging Challenges:** Complex & data-intensive
- **MVP Approach:** Learn by doing to uncover & understand
- **Data Integration:** Links omics data with histology whole slide imaging
- **Tool Development:** Cell viewer, model monitoring, & feedback/annotation engine

©2024 Databricks Inc. — All rights reserved

Prov-GigaPath

Mastering our cell typing use case enables new model deployment

- World-leading computational pathology foundation model
- Deployment in Providence production env fundamentally similar
- Providence, Microsoft, & University of Washington collaboration
- Open Weights! Check it out in Nature, Github, or Huggingface

Prov-GigaPath achieves state-of-the-art performance, marks largest pretraining effort to date

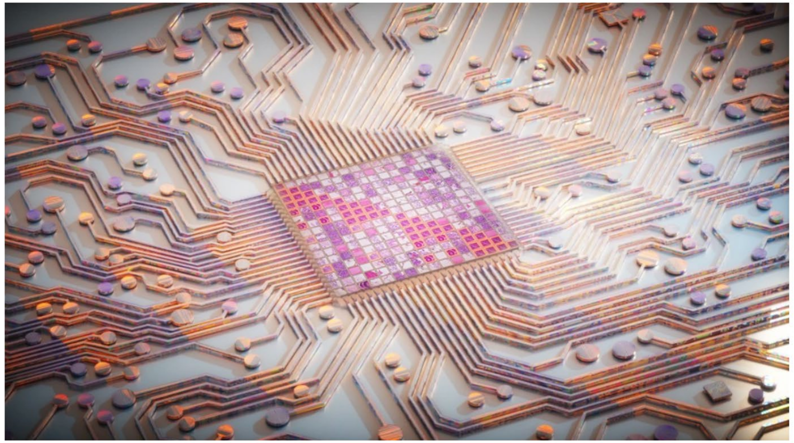


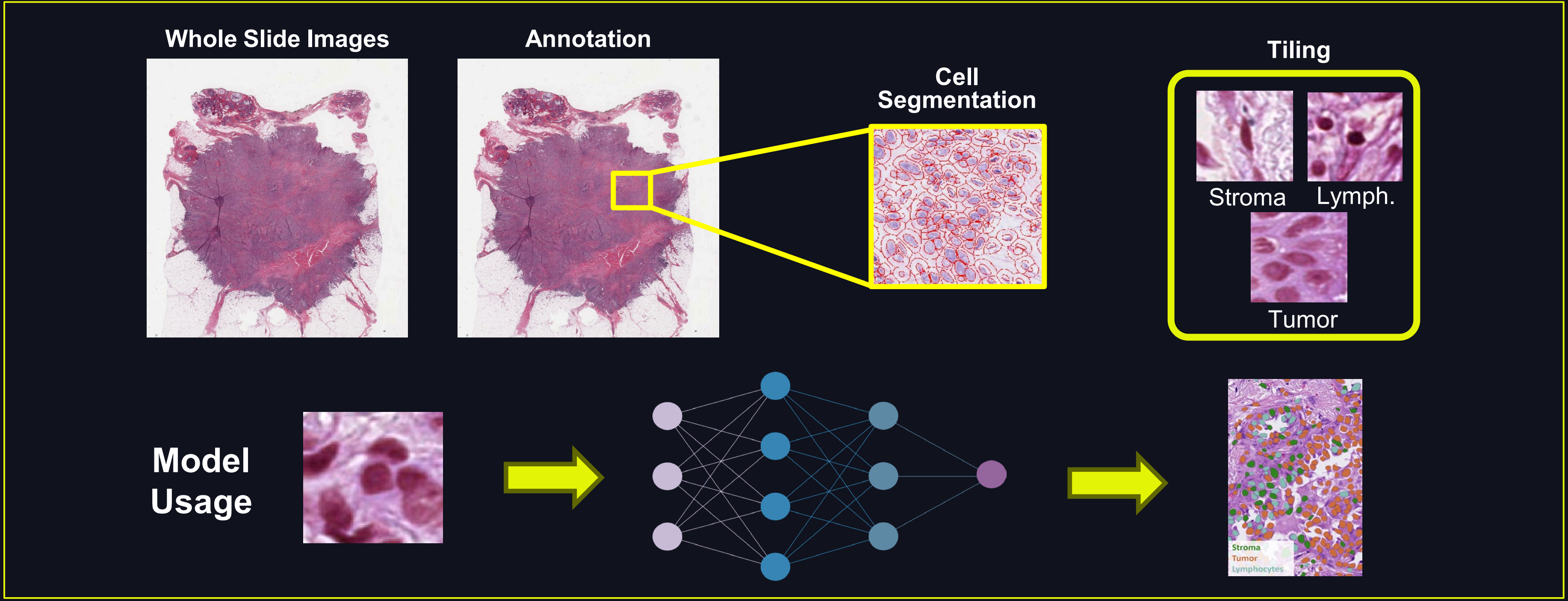
Image: Ella Maru Studio

"With the Potential to Transform Cancer Diagnostics, Providence Contributes to Innovative AI-Powered Digital Pathology Model." 2024. June 3, 2024. https://blog.providence.org/national-news/with-the-potential-to-transform-cancer-diagnostics-providence-contributes-to-innovative-ai-powered-digital-pathology-model.

©2024 Databricks Inc. — All rights reserved

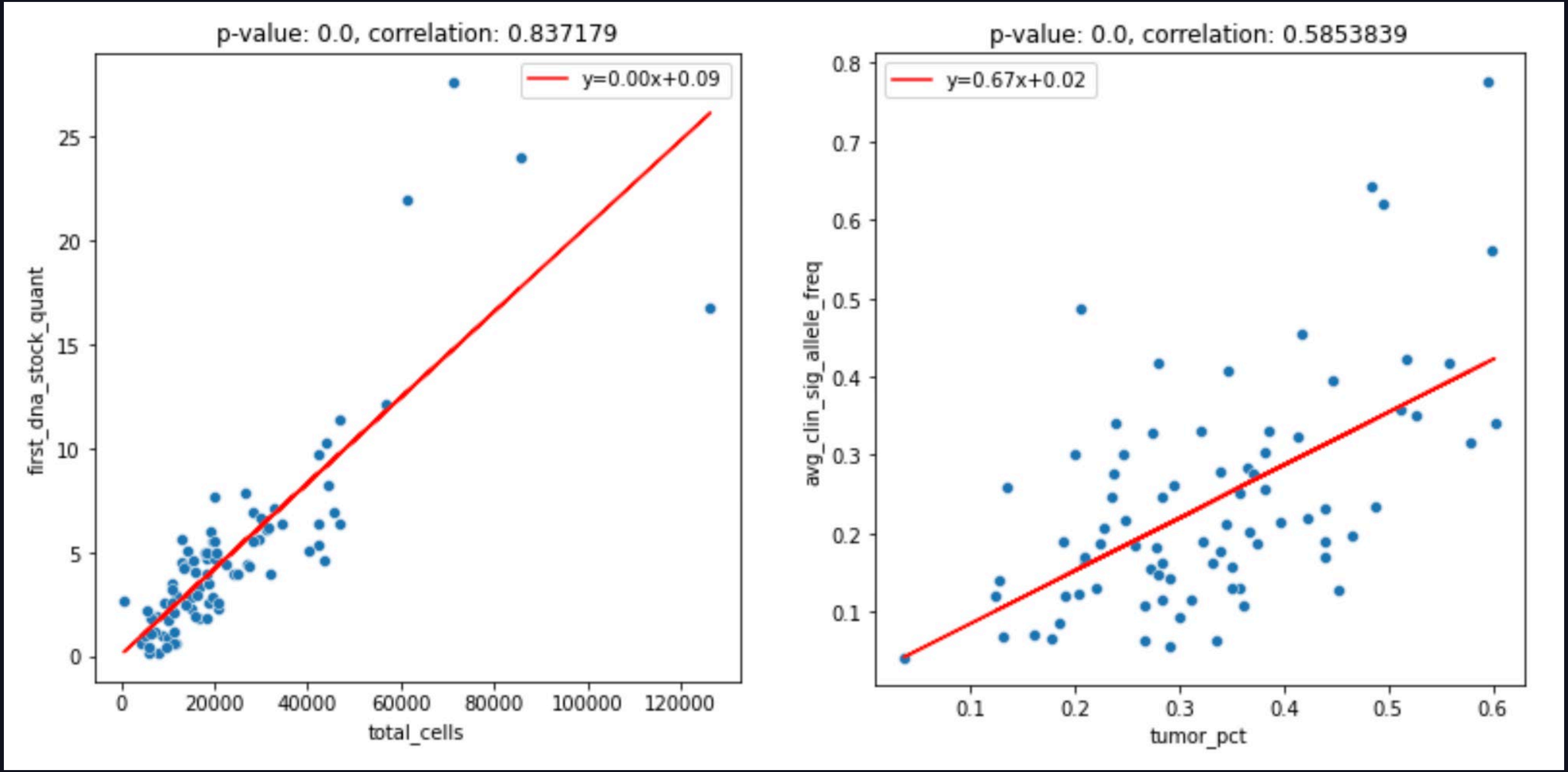
Cell Typing Model Overview

Histology imaging AI case study



©2024 Databricks Inc. — All rights reserved

Tumor % & Total Cells are Predictive



©2024 Databricks Inc. — All rights reserved

The Challenge of Scale

We have over 125k WSI's scanned from our Microsoft Research partnership

Data Volume:

- ~30k tiles & 2 GB per whole slide image (WSI)
- 125k+ historical image dataset
- 20 new cases per day

Infrastructure Challenges

- ~3hr to process 1 WSI
- OpenSlide inefficient with cloud storage
- Need effective executor VM caching strategies

©2024 Databricks Inc. — All rights reserved

The Bridge of Databricks

From AI research to production workflows

DISCOVER

Developed Cell Typing Model with researchers & presented at Association for Molecular Pathology

PARTNER

Databricks offered Digital Pathology Accelerator resources & consultation

BUILD

Adapt Python based model to Pyspark

Solve large scale image processing issues

DELIVER

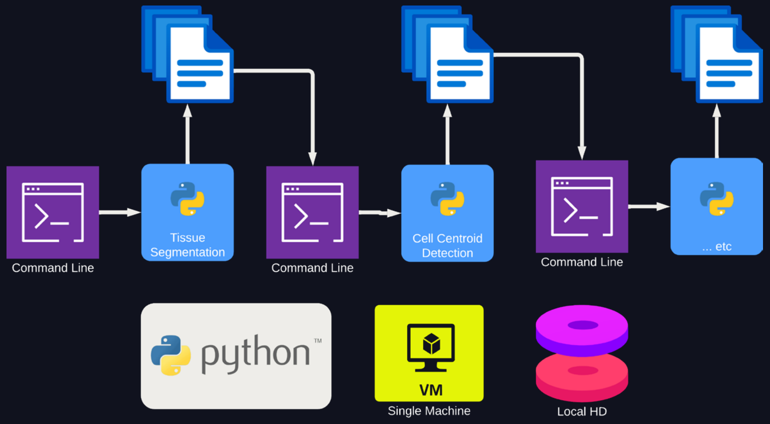
Easily integrate with DBX based genomic variant clinical workflow

©2024 Databricks Inc. — All rights reserved

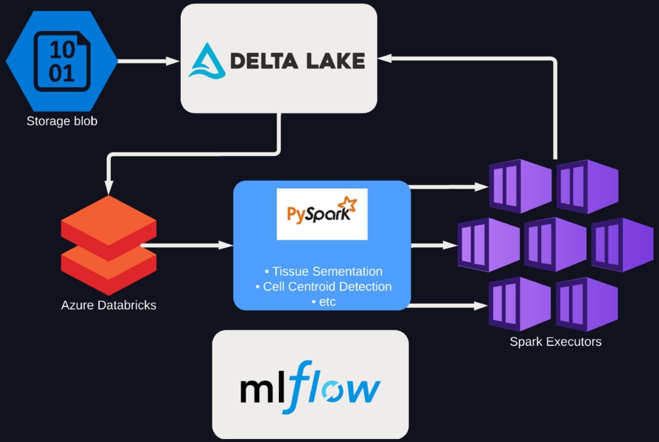
Research Model -> Production Model

Initial Python / Pytorch cell typing model developed for single VM

Research Single Machine

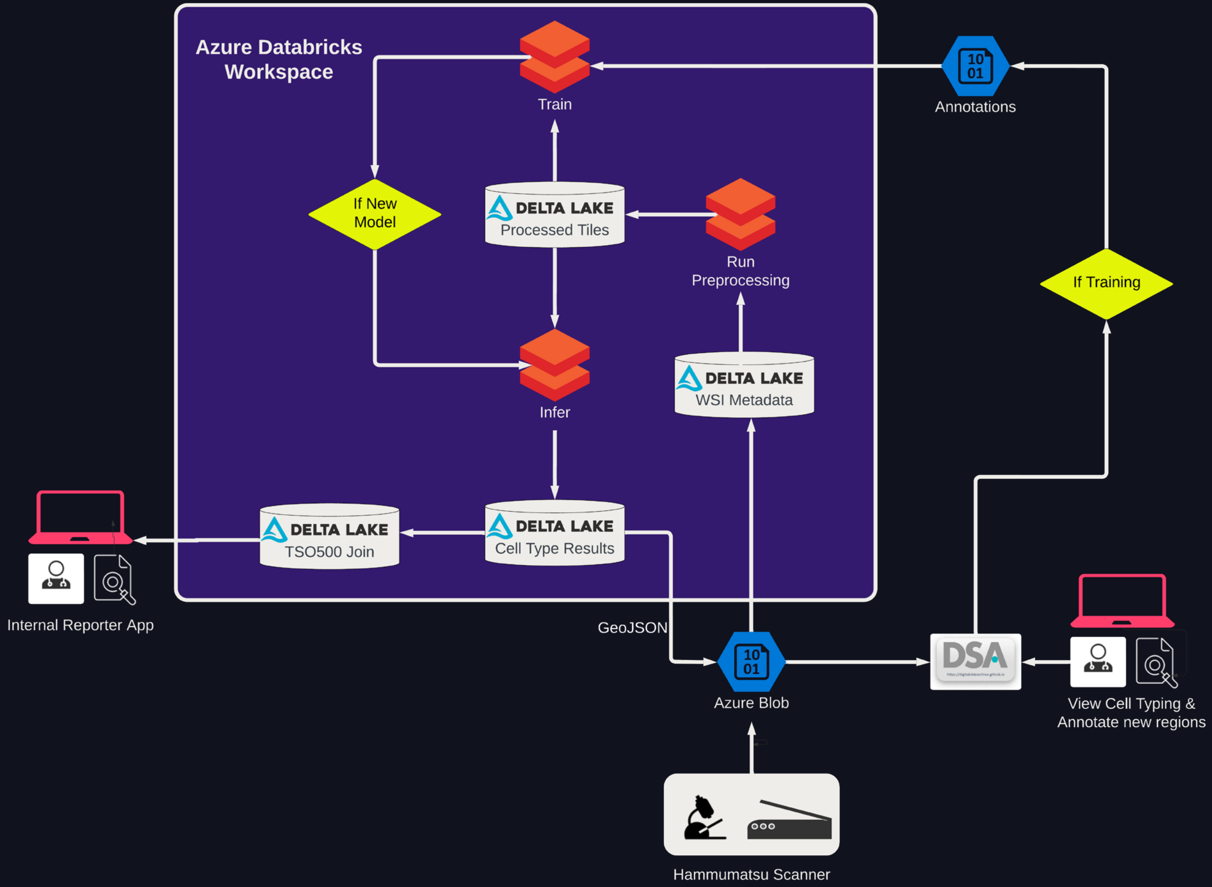


Production Distributed Spark Model



©2024 Databricks Inc. — All rights reserved

Planned Production Workflow

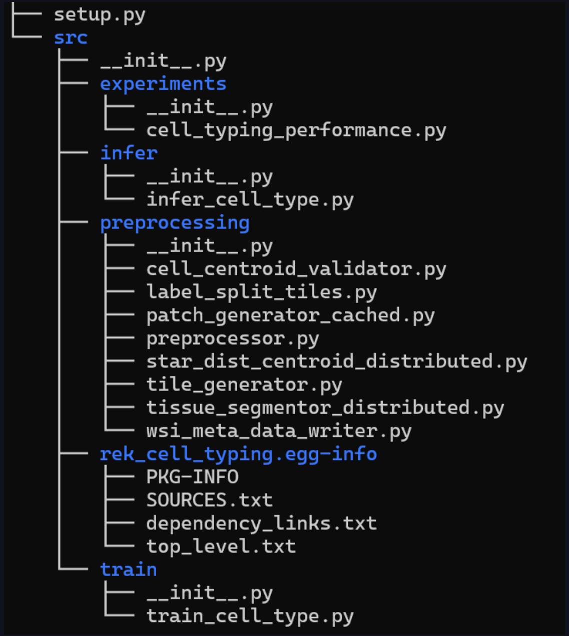


©2024 Databricks Inc. — All rights reserved

Distributed Cell Typing Source Code

Use repos and arbitrary files to create typical Python modules

1. Metadata table consisting of information from scanned whole slide images (WSI, .ndpi) created from reading blob storage
2. The WSI's are split into tile coordinates & each preprocessing class acts on independent tiles as rows in a delta table
3. WSI's are cached on each executors VM HD as needed for efficient I/O of OpenSlide image objects



```
├── setup.py
└── src
    ├── __init__.py
    ├── experiments
    │   ├── __init__.py
    │   └── cell_typing_performance.py
    ├── infer
    │   ├── __init__.py
    │   └── infer_cell_type.py
    ├── preprocessing
    │   ├── __init__.py
    │   ├── cell_centroid_validator.py
    │   ├── label_split_tiles.py
    │   ├── patch_generator_cached.py
    │   ├── preprocessor.py
    │   ├── star_dist_centroid_distributed.py
    │   ├── tile_generator.py
    │   ├── tissue_segmentor_distributed.py
    │   └── wsi_meta_data_writer.py
    ├── rek_cell_typing.egg-info
    │   ├── PKG-INFO
    │   ├── SOURCES.txt
    │   ├── dependency_links.txt
    │   └── top_level.txt
    └── train
        ├── __init__.py
        └── train_cell_type.py
```

Inference Code Walkthrough

Delta table split and sent to executors as independent Pandas Dataframes, allowing reuse of python classes with I/O modifications

Delta Table based orchestration

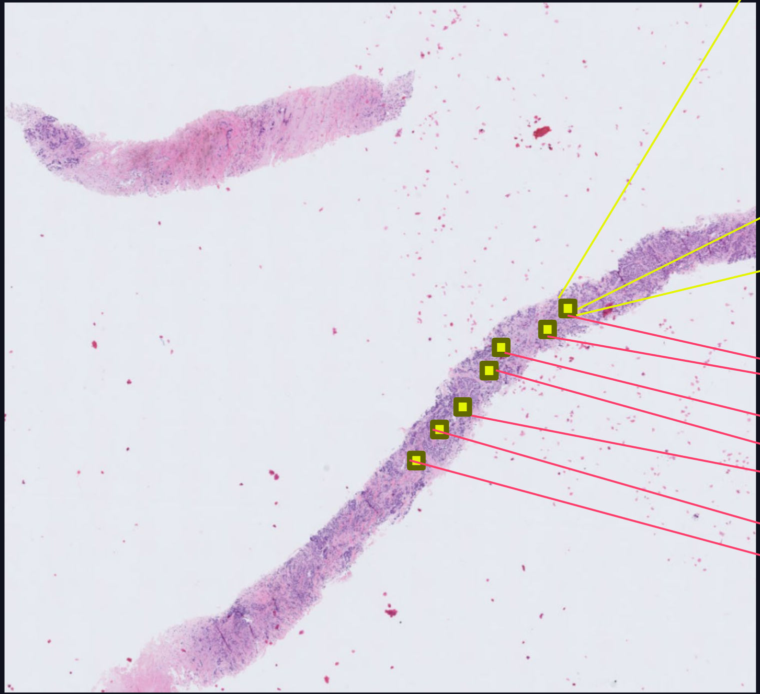
```
preprocessor = pre.Preprocessor(preprocess_config)
preprocessor.prepare_meta_wsi_df()

mask_extract_df =
    preprocessor.prepare_mask_extract_df(cache_flag)
tiled_df = preprocessor.prepare_tiled_df(mask_extract_df,
    cache_flag)
centroid_df = preprocessor.get_centroids(tiled_df, cache_flag)
centroid_patch_df =
    preprocessor.get_centroid_patch_df(centroid_df,
    cache_flag)
```

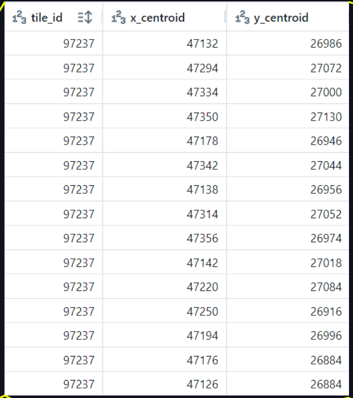
Distribute by applying with MapInPandas()

```
from src.infer import infer_cell_type as inf
cell_type_processor = inf.InferCellType(model_dir=model_dir,
    labels_list=preprocessor.labels_list)

    pred_df = (
   centroid_patch_df
   .repartition(32)  # Adjust based on your cluster setup and
    data size
   .mapInPandas(
      cell_type_processor.make_predictions,
      schema=cell_type_processor.schema
   )
)
```

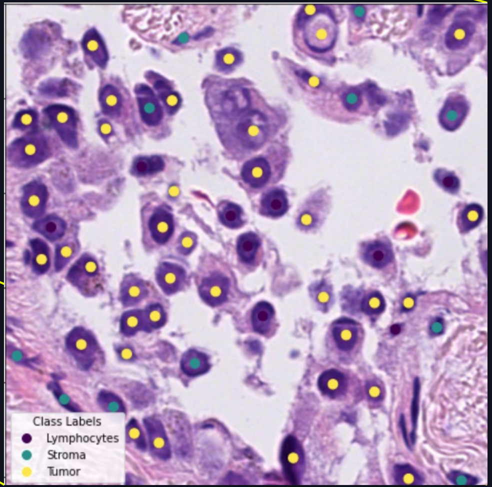
©2024 Databricks Inc. — All rights reserved

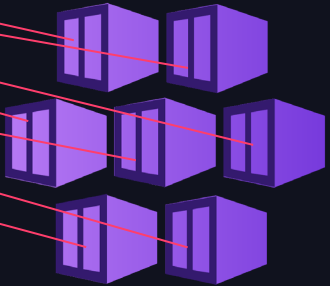
MapInPandas() Maps Python/PyTorch Across Spark Executors





tile_id	x_centroid	y_centroid
97237	47132	26986
97237	47294	27072
97237	47334	27000
97237	47350	27130
97237	47178	26946
97237	47342	27044
97237	47138	26956
97237	47314	27052
97237	47356	26974
97237	47142	27018
97237	47220	27084
97237	47250	26916
97237	47194	26996
97237	47176	26884
97237	47126	26884



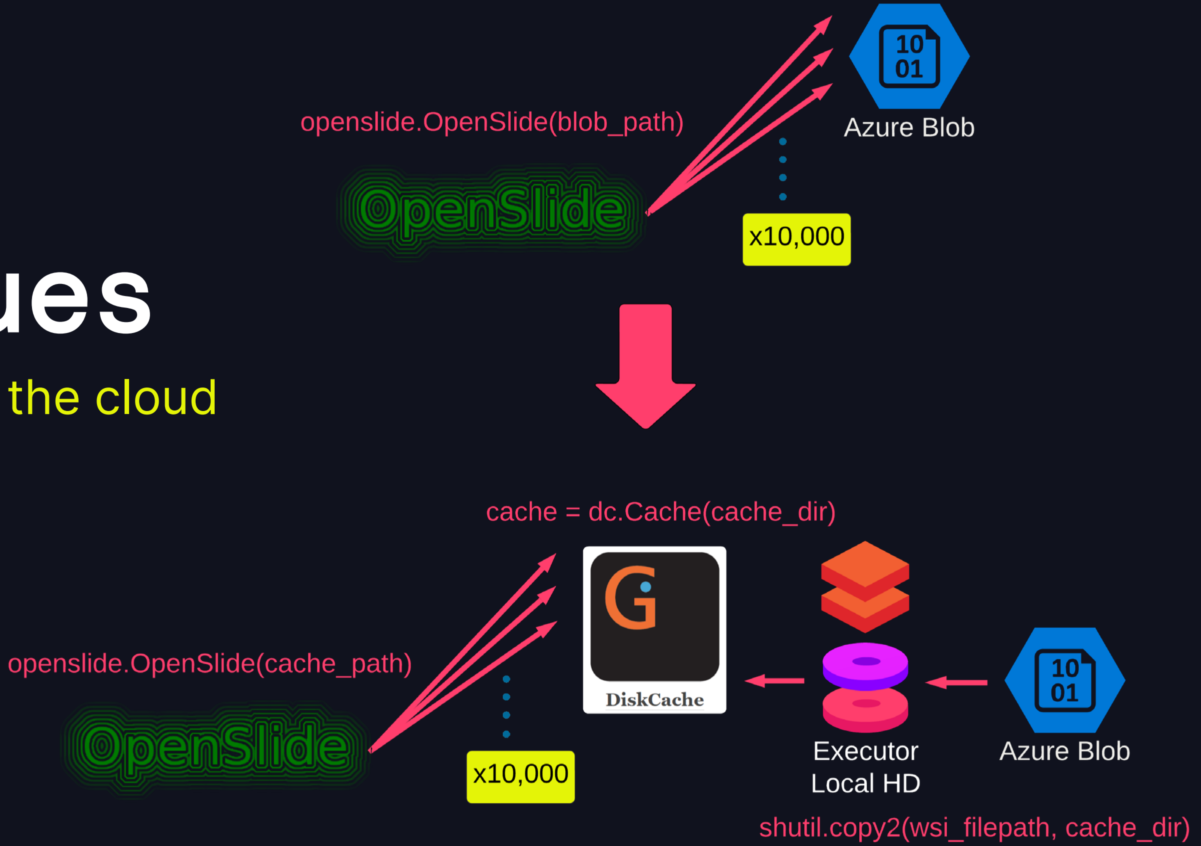


Tiles Spread Across Executors

©2024 Databricks Inc. — All rights reserved

I/O Issues

Using OpenSlide in the cloud

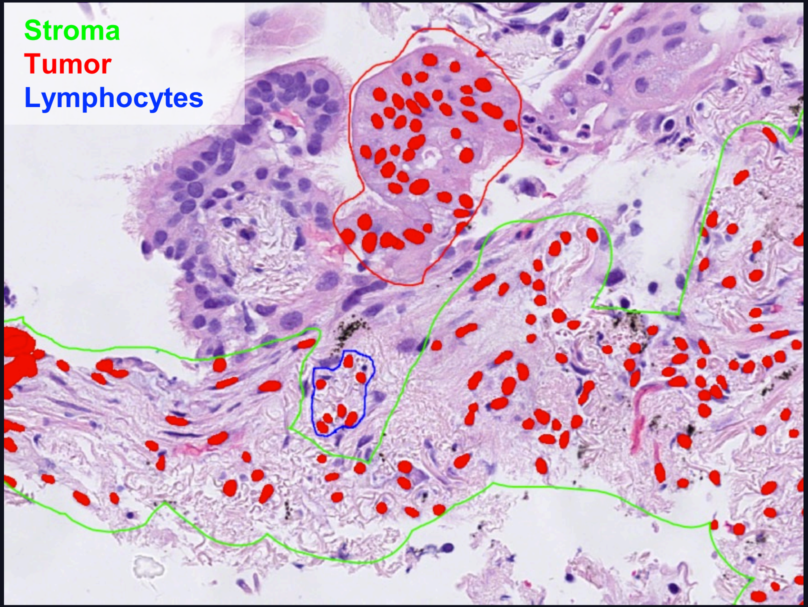

openslide.OpenSlide(blob_path)
OpenSlide
Azure Blob
x10,000
cache = dc.Cache(cache_dir)
openslide.OpenSlide(cache_path)
OpenSlide
x10,000
DiskCache
Executor
Local HD
Azure Blob
shutil.copy2(wsi_filepath, cache_dir)

©2024 Databricks Inc. — All rights reserved

Label Cell Types with Spatial Joins

Vectorized Shapely 2.0 operations distributed with MapInPandas()

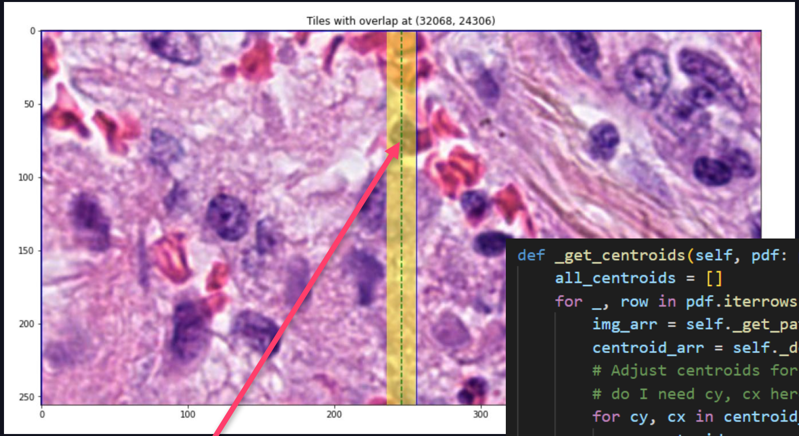
- Regions labeled by Molecular Genomics Lab pathologist available as GeoJSON files
- Cell Centroids detected with StarDist Keras Model
- For each region geometry, find all cell centroids in the annotated region with a vectorized spatial join



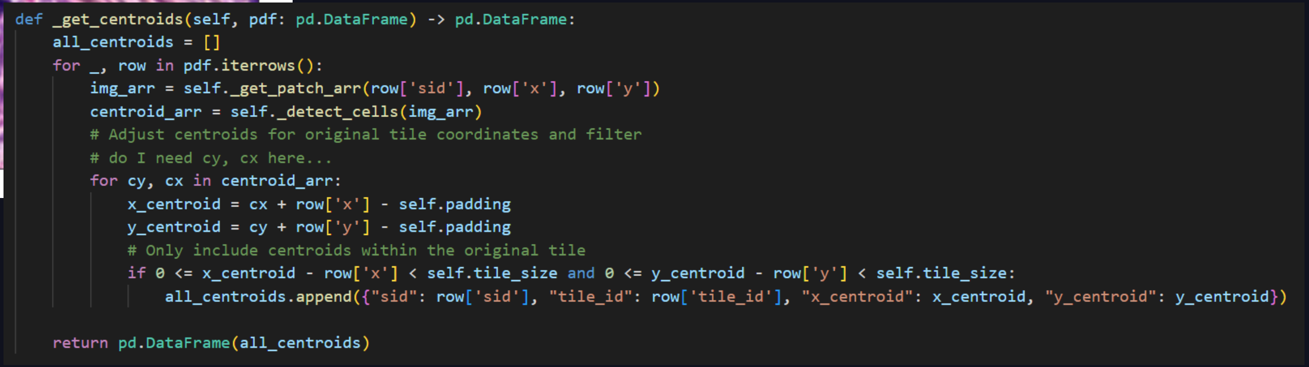
©2024 Databricks Inc. — All rights reserved

Distributing Stardist Cell Centroid Model

Add padding to all tiles and use tile id to predict unique edge case cells



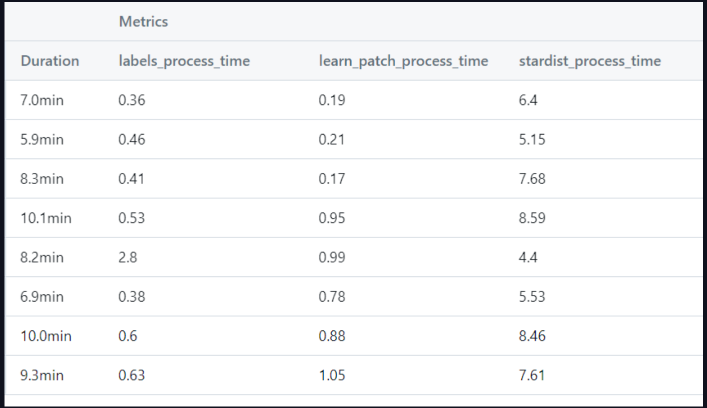
How do we predict cells on the edge?



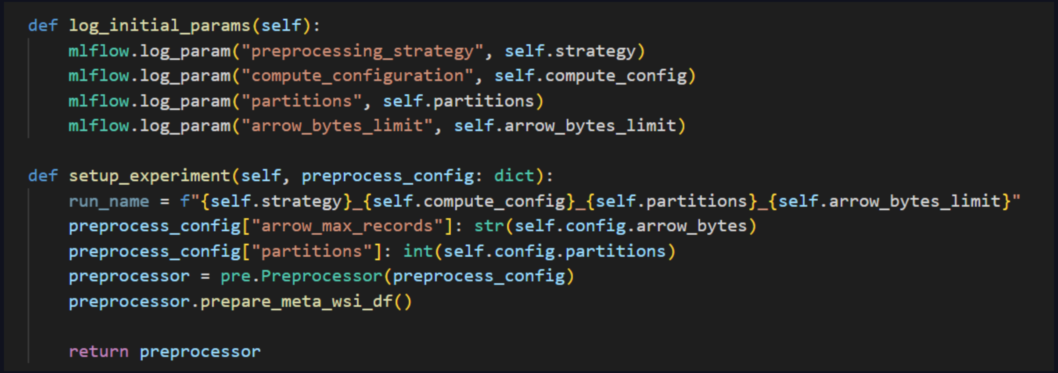
```python
def _get_centroids(self, pdf: pd.DataFrame) -> pd.DataFrame:
    all_centroids = []
    for _, row in pdf.iterrows():
        img_arr = self._get_patch_arr(row['sid'], row['x'], row['y'])
        centroid_arr = self._detect_cells(img_arr)
        # Adjust centroids for original tile coordinates and filter
        # do I need cy, cx here...
        for cy, cx in centroid_arr:
            x_centroid = cx + row['x'] - self.padding
            y_centroid = cy + row['y'] - self.padding
            # Only include centroids within the original tile
            if 0 <= x_centroid - row['x'] < self.tile_size and 0 <= y_centroid - row['y'] < self.tile_size:
                all_centroids.append({"sid": row['sid'], "tile_id": row['tile_id'], "x_centroid": x_centroid, "y_centroid": y_centroid})

    return pd.DataFrame(all_centroids)
```

©2024 Databricks Inc. — All rights reserved

Mlflow for Performance Experiments



	Metrics		
Duration	labels_process_time	learn_patch_process_time	stardist_process_time
7.0min	0.36	0.19	6.4
5.9min	0.46	0.21	5.15
8.3min	0.41	0.17	7.68
10.1min	0.53	0.95	8.59
8.2min	2.8	0.99	4.4
6.9min	0.38	0.78	5.53
10.0min	0.6	0.88	8.46
9.3min	0.63	1.05	7.61



```python
def log_initial_params(self):
    mlflow.log_param("preprocessing_strategy", self.strategy)
    mlflow.log_param("compute_configuration", self.compute_config)
    mlflow.log_param("partitions", self.partitions)
    mlflow.log_param("arrow_bytes_limit", self.arrow_bytes_limit)

def setup_experiment(self, preprocess_config: dict):
    run_name = f"{self.strategy}_{self.compute_config}_{self.partitions}_{self.arrow_bytes_limit}"
    preprocess_config["arrow_max_records"]: str(self.config.arrow_bytes)
    preprocess_config["partitions"]: int(self.config.partitions)
    preprocessor = pre.Preprocessor(preprocess_config)
    preprocessor.prepare_meta_wsi_df()

    return preprocessor
```

©2024 Databricks Inc. — All rights reserved

Distributed Production Results

The distributed pipeline is faster, scalable, and more cost effective

Process Step	Distributed Avg 1-WSI /5.25 DBU Compute	Research Avg 1-WSI /5 DBU GPU Compute
Stardist Cell Centroid Detection	5 min	16 min
Cell Typing Inference	4 min	164 min

- Infinite scale: 6 executors ~9 min/slide -> 30 executors ~1.8 min/slide
- Delta Lake meta-data orchestration allows for quick analysis
- DiskCache handles the concurrent OpenSlide I/O well
- Simple integration with our existing clinical Databricks workflows

©2024 Databricks Inc. — All rights reserved

Acknowledgements

Special thanks to:

Earle A. Chiles Research Institute:

Angela Crabtree – Researcher & Initial Cell Typing Developer

Brian Piening, PhD – Molecular Pathology Core Director (ML Group)

Providence Molecular Genomics Lab:

Jacob Able, MD & Christine Moung-Wen, MD - Pathologist Annotators

Carlo Bifulco, MD Director Molecular Genomics Lab

Providence Healthcare Intelligence

Lindsay Mico, AVP Enterprise Data Science

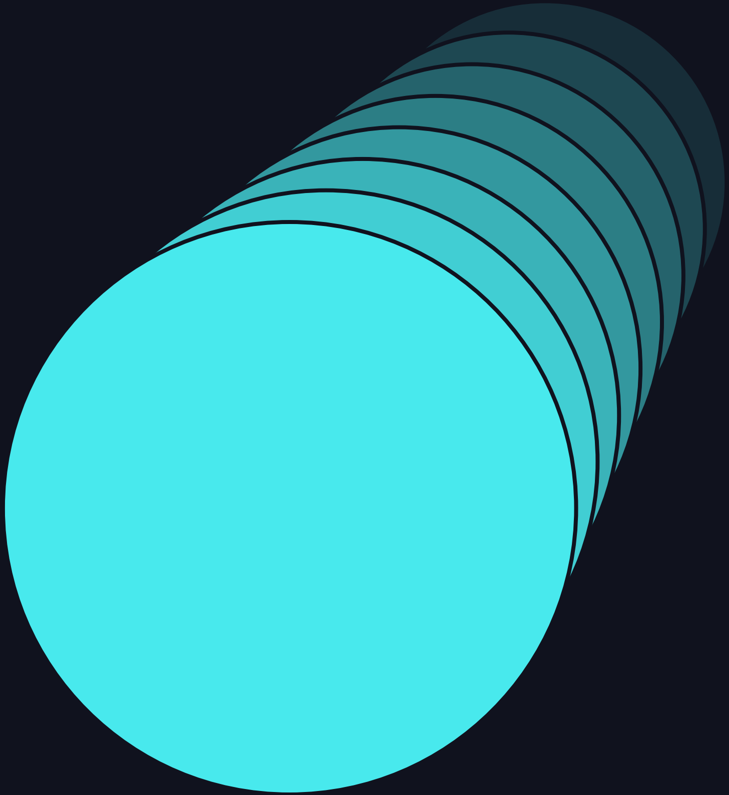
Databricks Partners

Providence Health Innovation Research

©2024 Databricks Inc. — All rights reserved

DATA+AI SUMMIT BY databricks

Questions?



©2024 Databricks Inc. — All rights reserved

DATA+AI SUMMIT

©2024 Databricks Inc. — All rights reserved

Appendix

©2024 Databricks Inc. — All rights reserved

Deep Learning Applications Using H&E Images Improve Clinical Sequencing Workflows

Jacob Abel[1*], Angela Crabtree[2*], Robert Kramer[1], Christine Moung-Wen[1], Lingyu Wang[1], Eric Shull[1], Brian Piening[1,2], Carlo Bifulco[1,2]
[1] Providence Health & Services Molecular Genomics Laboratory, Portland, OR, USA
[2] Earle A. Chiles Research Institute, Providence Cancer Institute, Portland, OR, USA
*These authors contributed equally to this work.

INTRODUCTION

The term "tumor purity" or "tumor percentage" (TP) describes the fraction of cancer cells in a sample as compared to non-tumor cells. Pre-analytical assessment of TP and subsequent sample acceptance or rejection is a critical component of quality control and is utilized for downstream processes such as the correction of gene copy number estimates. TP can be assessed through histologic estimation or aggregating the variant allele frequency (VAF) of somatic mutations, but these approaches are subjective. Here, we report on a machine-learning (ML) model for the quantitation of tumor, stroma, and lymphocytic cells from whole slide images (WSI) and how this could fit into a clinical workflow.

MATERIALS & METHODS 1

Our in-house database was queried for all non-cytologic cases of primary lung adenocarcinoma since 2022 which had accompanying Hematoxylin and Eosin (H&E) whole slide images (WSI) and TruSight Oncology 500 NGS data, resulting in a dataset of 280 cases. Of these, 38 cases were randomly selected for use as training (22), validation (7), and test (9) samples. Overall workflow is depicted in Figure 1. Two pathologists non-exhaustively annotated tissue regions containing high densities of the target cell types. The annotations were performed in QuPath and exported for labeling training data during preprocessing (Figure 2). Training data was prepared by segmenting cell nuclei using StarDist, then creating 96x96 pixel image tiles centered on each nucleus within annotated tissue regions (Figure 3). A VGG16 model was initialized with pre-trained weights and further trained on over 180,000 tiles, achieving 80% accuracy on a test set of 38,310 tiles (Figure 4). The model was used to classify a random subset of cells from each slide in a set of 276 unlabeled slides and inferences were aggregated at the slide level. Cell counts and cell type proportions in biopsy slides were assessed for correlations with NGS findings.

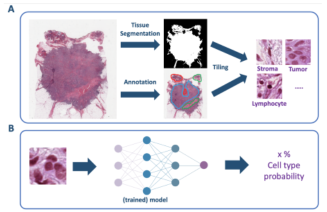


Figure 1: Workflow of (A) image preprocessing and (B) model utilization steps.

MATERIALS & METHODS 2

Annotation and Segmentation

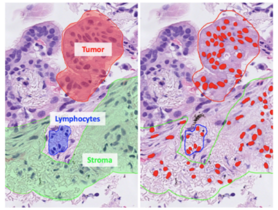


Figure 2: During the training process, regions of 38 lung adenocarcinoma slides were annotated as "tumor," "stroma," or "lymphocytes" (Left). Cell nuclei were then segmented from these regions using the StarDist segmentation model (Right).

Cell Tiles

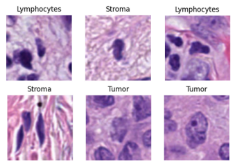


Figure 3: 96 x 96 pixel tiles were produced from the segmented nuclei. Each tile center is a cell centroid.

Model Performance

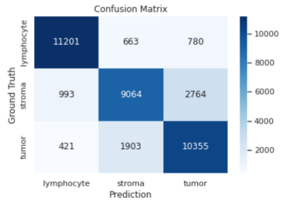


Figure 4: A confusion matrix showing performance of model vs ground truth (i.e. classified by pathologist) for the three cell categories assessed in this study.

RESULTS

The predicted number of total cells in core biopsies positively correlated with extracted DNA concentrations ($r^2 = 0.84$, $p < 0.001$), while stromal cell density was negatively correlated ($r^2 = -0.52$, $p = 0.001$). TP estimates were congruous with the pathologist-estimated tumor content (r2=0.50, p=0.002) and with the average clinically significant variant allele frequency (r2=0.59, p<0.001). Inspection of outliers revealed a sample where TP was particularly underestimated by the pathologist at sign-out and this case was flagged for review. It was determined that subclonality of detected variants skewed assessment of TP in this case.

Example Fields of Model Predicted Nuclei

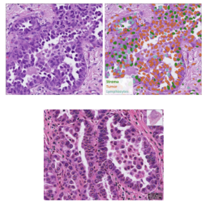


Figure 5: (**Top left**) Example field of lung adenocarcinoma. (**Top right)** Same field with segmented nuclei highlighted in color (stroma = green, tumor = red, and lymphocytes = teal). (**Bottom**) Example field of lung adenocarcinoma case where TP was underestimated by pathologist reviewer. The deceptive micropapillary morphology in conjunction with use of a subclonal *STAG2* variant highlight some of the challenges in TP estimation.

Comparison with Manual Quantitation of Tumor Cells

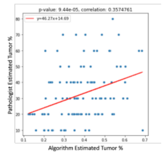


Figure 6: Plot of TP as estimated by pathologist vs. TP as estimated by algorithm. Linear regression performed in R.

Comparison with Mean Clinically Significant Variant Allele Frequency

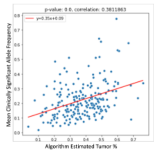


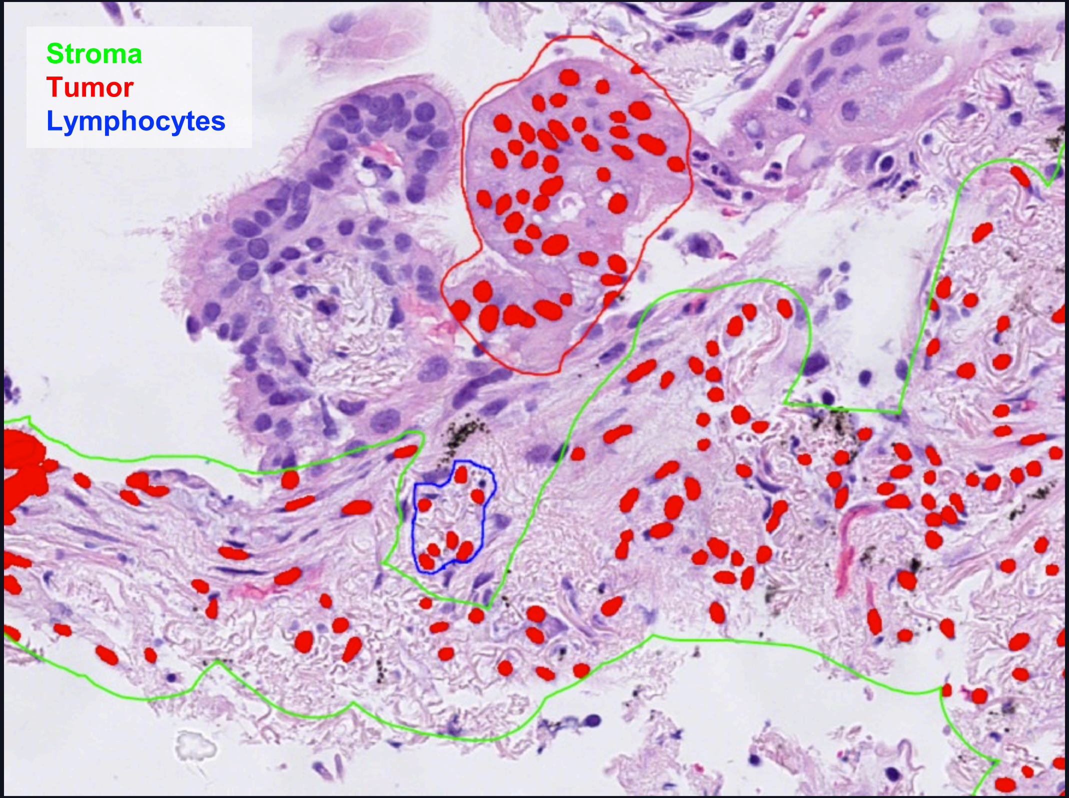
Figure 7: Plot of TP estimated by mean VAF of clin. significant mutations vs. algorithm estimated TP. Linear regression performed in R.

CONCLUSIONS

Automated TP estimation represents one example of how the integration of digital pathology and AI/ML tools can improve pathology workflows. The rapid and accurate quantitation of tumor cellular components is useful on its own as a quality metric and has a wide variety of potential applications both for routine clinical processes as well as enabling large-scale research analytics.

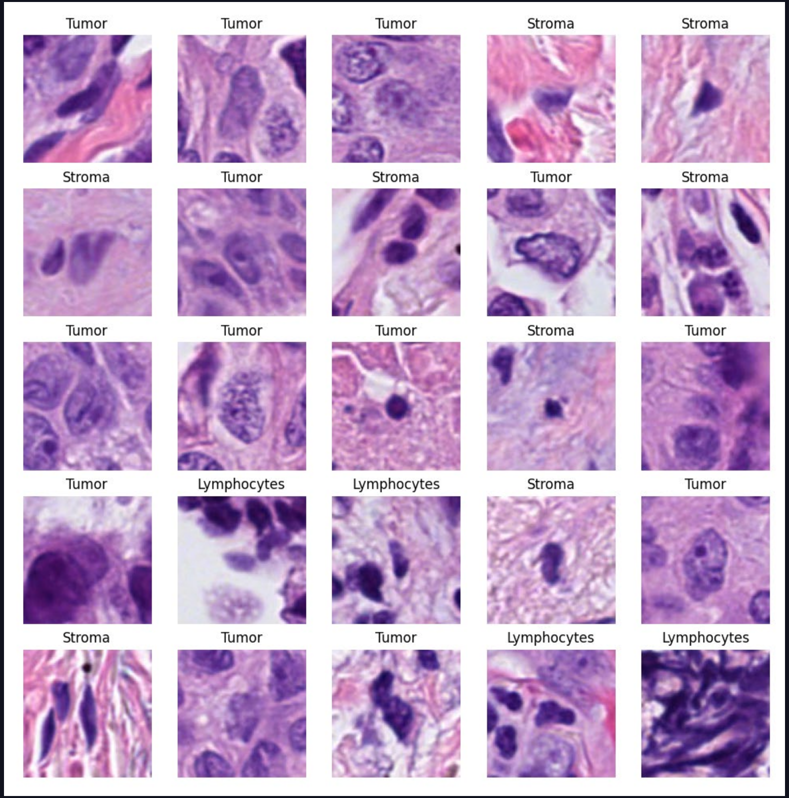
Created with BioRender Poster Builder

Cell types annotated by pathologists



©2024 Databricks Inc. — All rights reserved

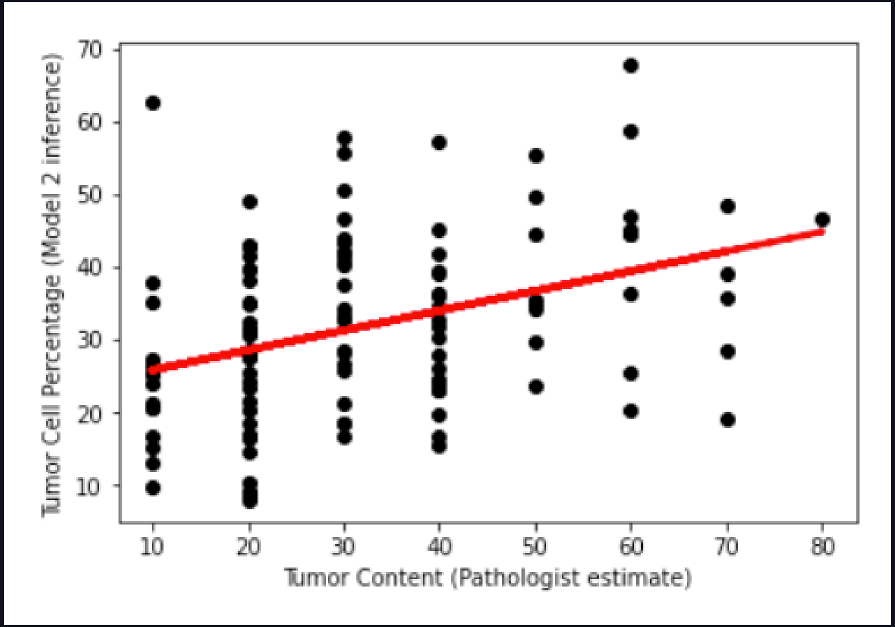
Tiles Generated from Cell Centroids



©2024 Databricks Inc. — All rights reserved

Tumor % predicted by model

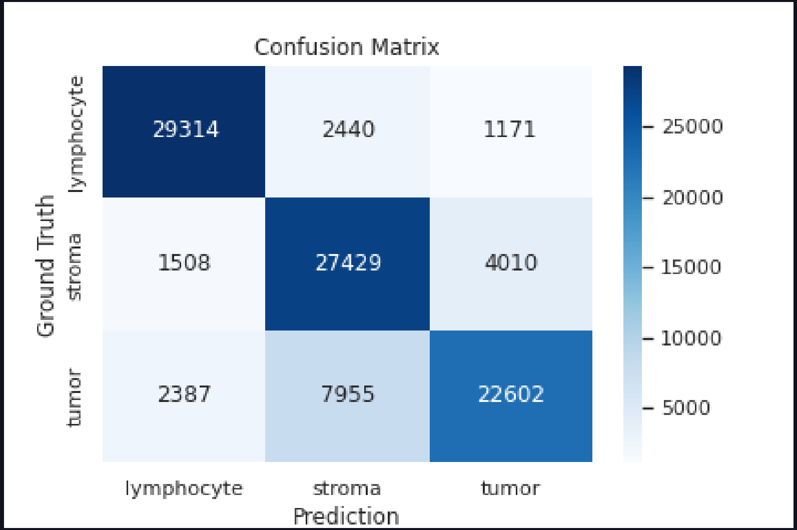
$r^2 = 0.50$, $p = 0.002$



©2024 Databricks Inc. — All rights reserved

3 class cell classifier trained

Confusion Matrix

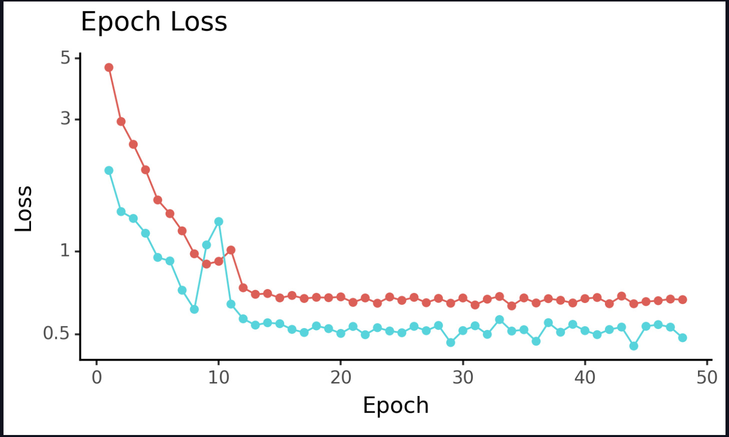


Confusion Matrix

Ground Truth \ Prediction	lymphocyte	stroma	tumor
lymphocyte	29314	2440	1171
stroma	1508	27429	4010
tumor	2387	7955	22602

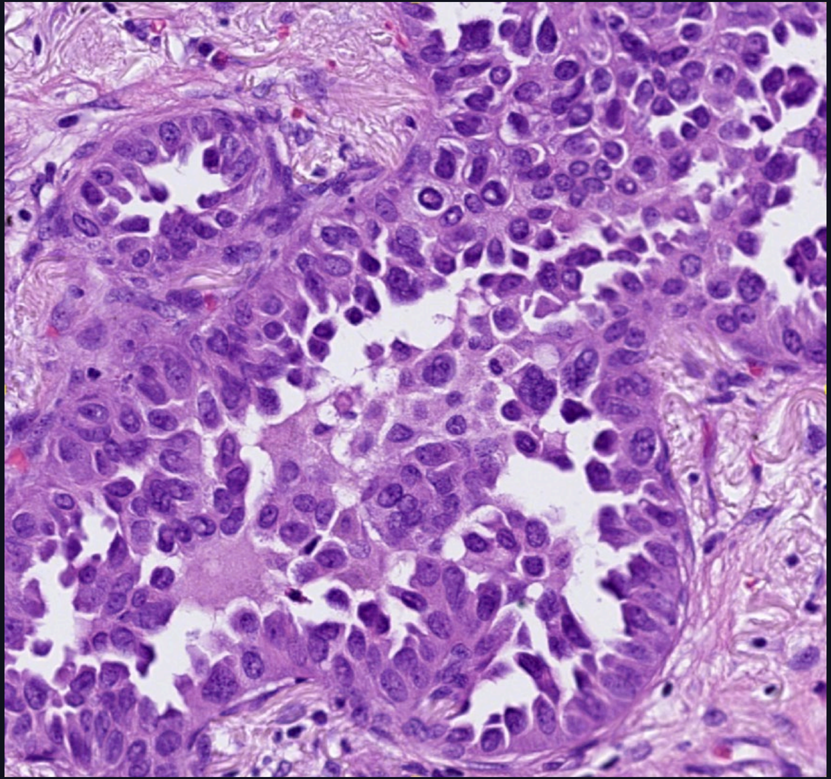
25000
20000
15000
10000
5000

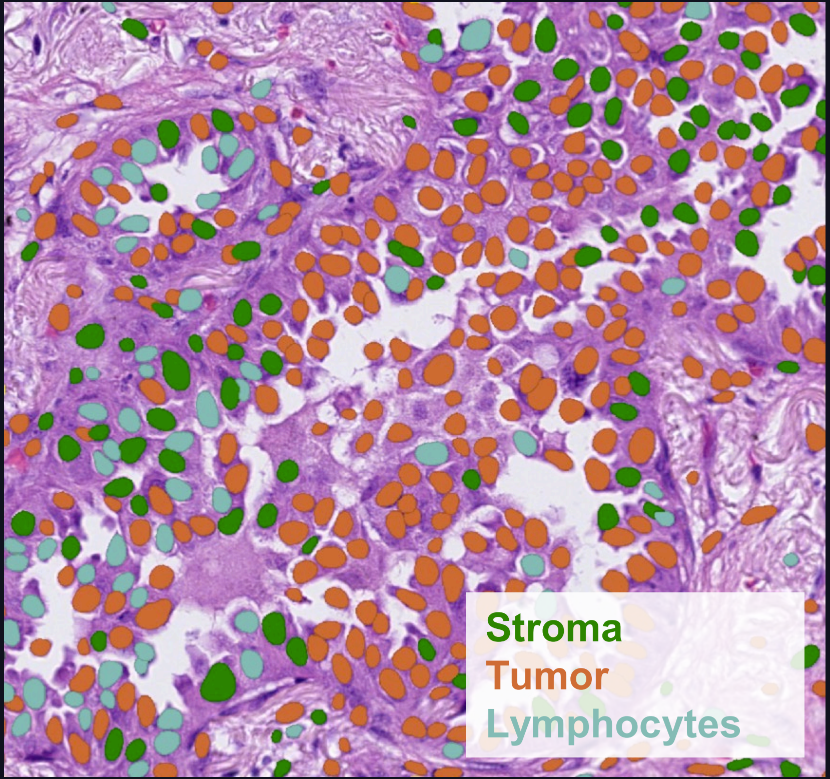
Model training loss



©2024 Databricks Inc. — All rights reserved

Predictions overlaid on cells





©2024 Databricks Inc. — All rights reserved