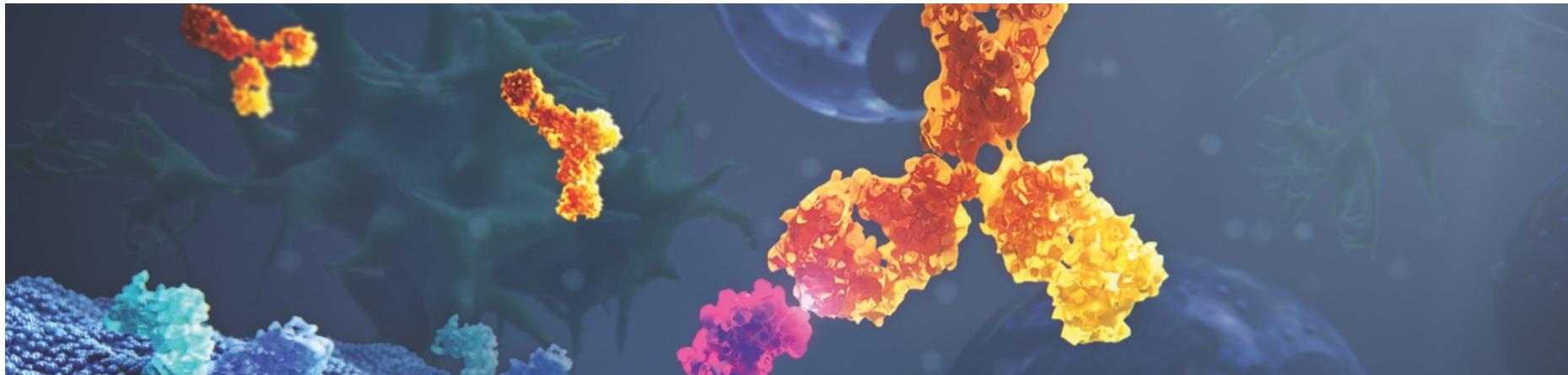


Hackathon Task and Evaluation Metrics

Dr. Lars Tornberg

Associate Principal Scientist, Data Science and Modelling, Pharmaceutical Sciences, AstraZeneca R&D

Nov 2020

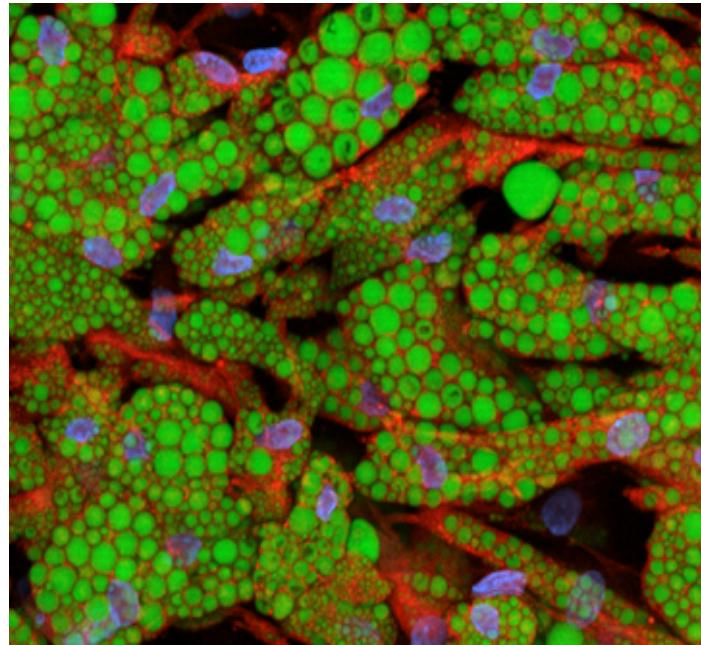
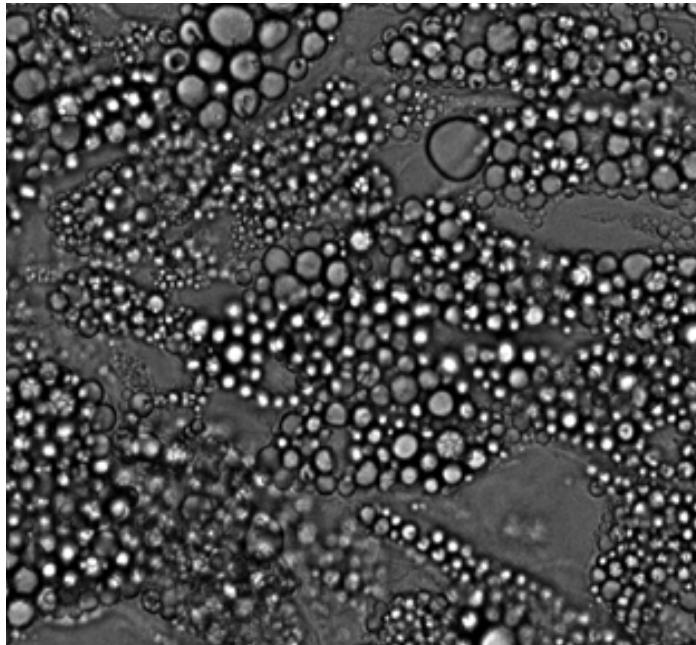


Outline

- Recap of the task
- Evaluation metrics
 - Overview
 - Quantitative evaluation
 - Project report
- Contact details



The Task

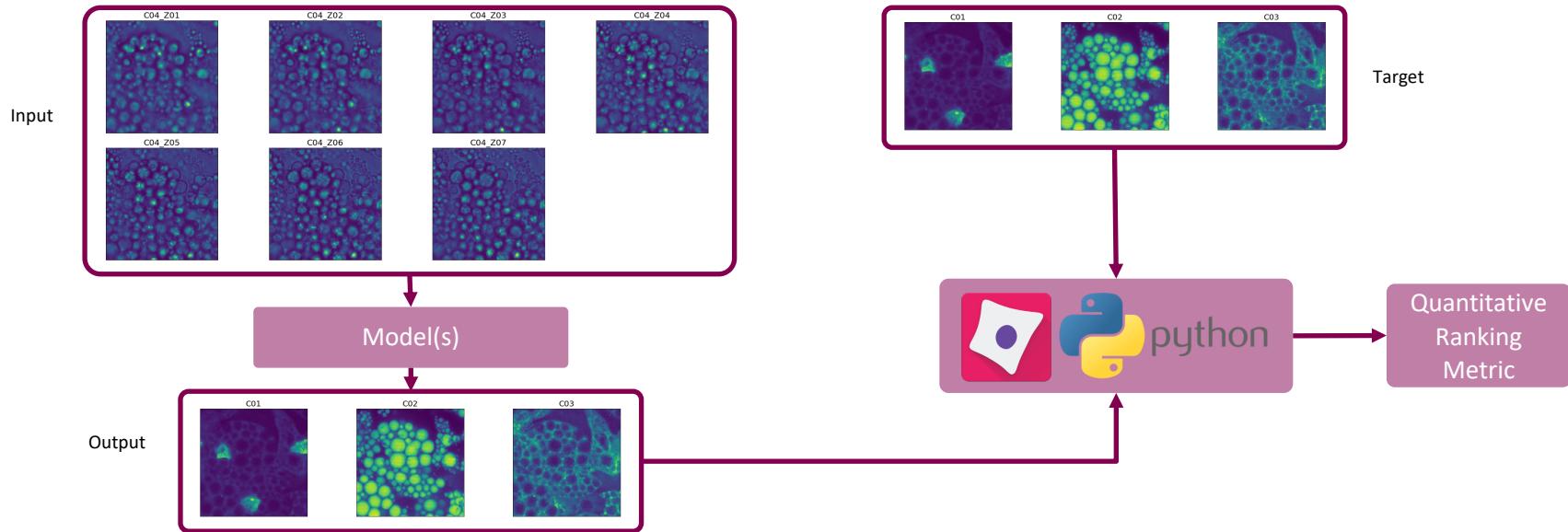


Evaluation Metrics Overview

Metric Description	Weight	Metric
Combined metric based on pixelwise and image feature mean absolute error – the fidelity is calculated on all channels and magnifications.	0.50	Float in {0,1} Number is normalized to the maximum error
Coverage of the project report with respect to code.	0.15	Three levels in {0,1} <ul style="list-style-type: none">• > 2 follow up questions• 1-2 follow up questions• No follow up.
Model complexity measured as number of tuned hyper-parameters including pre-processing, model architecture and training.	0.05	Float in {0,1} Number is normalized to the maximum contribution
Inference execution time on test set. This is run on A100.	0.05	Float in {0,1} Number is normalized to the maximum contribution
Jury presentation	0.25	Ten levels in {0,1} How was the problem solved and what was the result? This is an opportunity to show things that are not assessed in the metrics above.



Quantitative Evaluation Workflow

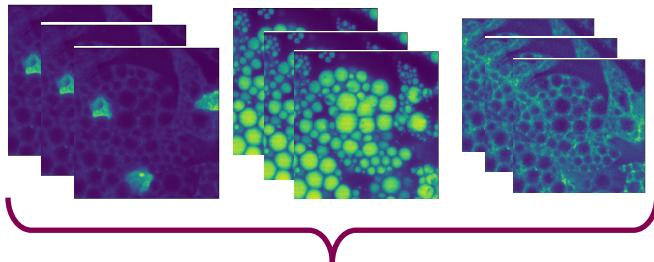


	20x	40x	60x
Wells	9	9	9
No. of image sites per well	6	8	12
Site orientation			

	20x	40x	60x
Training data	480	640	960
Hold out data	60	80	120



CellProfiler Feature Extraction



The directory must contain image triplets with the following naming convention:

- AssayPlate_Greiner_#655090_D02_T0001F007L01A01Z01C01.tif
 - AssayPlate_Greiner_#655090_D02_T0001F007L01A01Z01C02.tif
 - AssayPlate_Greiner_#655090_D02_T0001F007L01A01Z01C03.tif
- representing the three target channels for each well and field of view.



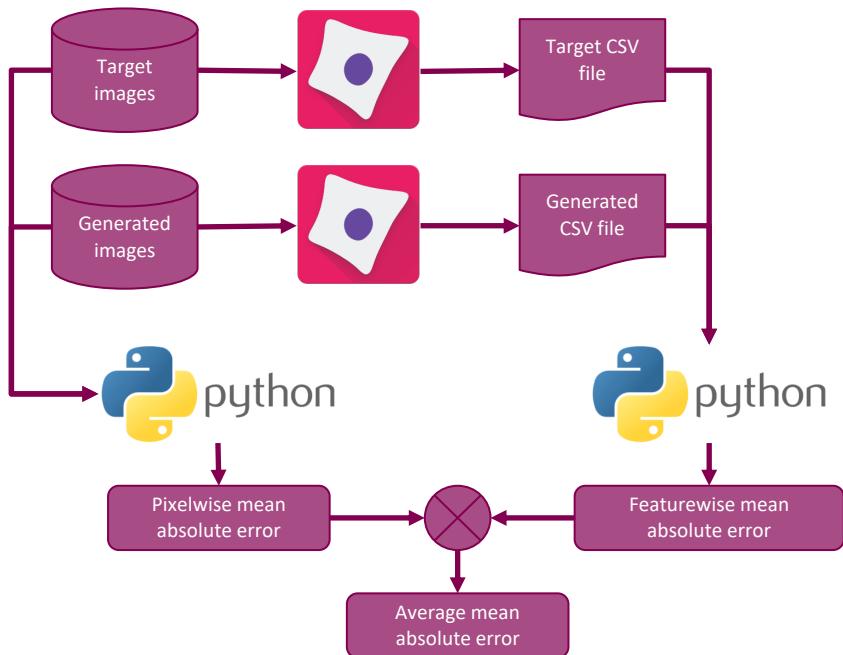
	Count_Lipids_no_edge	Count_cells_no_edge	Mean_Lipids_no_edge_AreaShape_Area
0	6347.0	25.0	406.597448
1	6882.0	41.0	379.807178
2	6765.0	30.0	372.060015

...

96 features per image triplet

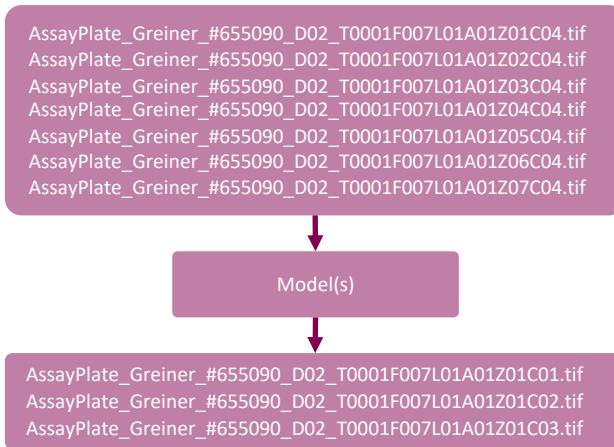


Quantitative Evaluation



$$\text{MAE}_{\text{tot}} = \frac{1}{n_{\text{tot}}} (n_{20x} \text{MAE}_{20x} + n_{40x} \text{MAE}_{40x} + n_{60x} \text{MAE}_{60x})$$

- **IMPORTANT!** To be able to compare target and generated images using the provided pipe-line, the images in the respective sets need to adhere to the same naming convention.
- *We recommend that the generated images are named identically to the corresponding targets to avoid errors in the evaluation process.*



Quantitative Evaluation Pipeline

main 1 branch 0 tags

Go to file Add file Code

Tornberg, Lars updated readme e40e571 2 days ago 34 commits

cellprofiler_pipelines updated readme 2 days ago

documentation updated readme 2 days ago

evaluation_code updated readme 2 days ago

.gitignore added gitignore 2 days ago

README.md updated readme 2 days ago

README.md

Adipocyte Cell Challenge

In this repository we provide extensive documentation to facilitate the understanding of the problem, its context and how the hackathon contributions will be evaluated. We furthermore provide the necessary code to run the evaluation suite used to generate the quantitative evaluation metric used to assess the hackathon contributions. The content and structure of the repo is given by the following

```
.
├── README.md
├── cellprofiler_pipelines
│   ├── Adipocyte_pipeline_20x_ver2.cpproj : Cell profiler pipe line to generate image features for 20x magnification
│   ├── Adipocyte_pipeline_40x_ver1.cpproj : Cell profiler pipe line to generate image features for 40x magnification
│   └── Adipocyte_pipeline_60x_ver1.cpproj : Cell profiler pipe line to generate image features for 60x magnification
├── README.md
└── cp_pipeline.png
```

documentation

```
.
├── README.md
└── dataset\ description.pdf : Description of data set from the call
    └── evaluation_criteria_hackathon.pdf : Description of all evaluation metrics and how they are calculated
        └── problem_formulation_with_logos.pdf : Problem formulation from the call
    └── report_template.docx : Template detailing the expected content of the project report
    └── report_template.pdf : Template detailing the expected content of the project report
```

evaluation_code

```
.
├── README.md
└── evaluation_pipe_line.PNG
```

hackathon_evaluation_metrics.ipynb : code to run the evaluation pipe line for the quantitative evaluation

For more detailed information about the content of each folder, please see the corresponding readme file.

Hackaton evaluation metrics

In this notebook we provide the functions that are used to calculate the quantitative evaluation metric for each magnification in the data set. The quantitative evaluation metric used to assess the hackathon contributions is divided into two parts: The images are compared using the mean absolute error (MAE) between generated and ground truth images using 1) features extracted by the CellProfiler and 2) pixel values. These are combined by averaging.

The evaluation suite in this notebook will be run on each magnification separately and the final metric is given by the weighted average

$$\text{MAE}_{\text{tot}} = \frac{1}{n_{\text{tot}}} (n_{20x} \text{MAE}_{20x} + n_{40x} \text{MAE}_{40x} + n_{60x} \text{MAE}_{60x})$$

where n_{tot} and $n_{20x} / n_{40x} / n_{60x}$ is the number of images in the total and 20x / 40x / 60x data set respectively.

Function Definitions

Below we give the functions that underlie the calculation of each metric.

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.metrics import mean_absolute_error
import sys
import os
import glob
import re

def get_featurewise_mean_absolute_error(targ_file, pred_file):
    """The relative mean absolute error between two data sets.

    Parameters
    -----
    targ_file : str
        Path to csv file containing the CellProfiler results for ground truth images
    pred_file : str
        Path to csv file containing the CellProfiler results for generated images

    Returns
    -----
    mae_per_feature : array
        Mean absolute error (mae) for each feature in the dataset. Each feature-mae is normalized
        with the corresponding feature median to account for different feature scales.

    mae : float64
        Averaged mae_per_feature

    feature_names : object
        names of features in data set
    """

    # read the results into dataframes
    df_targ = pd.read_csv(targ_file)
    df_pred = pd.read_csv(pred_file)
```



Project Report

- The purpose of the project report is to enable the end-user to understand the solution and to run the code for evaluation.
- The report will be input to a paper describing the hackathon contributions, data set and evaluation approach.
- The quality and coverage of the project report are themselves part of the evaluation criteria for the hackathon contribution.

Adipocyte Hackathon Report Template

TO BE REMOVED BEFORE SUBMITTING This is a template for the project report to be handed in at the hackathon dead-line. The purpose of this document is to enable the end-user to understand the solution and to run the code for evaluation. The report is also a document which allows the end-user to re-produce the result of the solution from scratch.

Furthermore, the quality and coverage of the project report are themselves part of the evaluation criteria for the hackathon contribution. This template thus serves the purpose of aligning the reports from the different hackathon participants, allowing a fair report comparison.

High level description of submitted solution

This section describes the high-level description of the submitted solution. This should include all main steps in the data processing, modelling and analysis. Pointers to code for the different steps. We encourage the use of flow charts and other visual aids when communicating this.

Data Processing

This section describes the steps taken to process the data from raw input to analytical data set. This could include e.g. data loading, data cleaning, filtering augmentation etc. What methods (if any) are used to pre-process the images. The size and form of the resulting analytical data set used for modelling should be documented.

Data Processing Parameters

This section includes descriptions and values of the parameters in the methods used in the processing step to generate the analytical data set.

Model Architecture

This section details the model architecture. Examples of things to include: High level building blocks such as residual learning, dense-connections, multi-path learning etc. We encourage the use figures and other visual aids to help communicate this. Pointers to code for the different steps/parts.

Model Parameters

This section includes description and value of the parameters used to define the model architecture e.g. number of layers, number of high-level building blocks, convolution filter parameters, choice of non-linearities etc.

Model Size

Here is information about the model size including number of trainable parameters and size in memory.

Model Output

Here goes the description of the model output.

Model Training

Training Scheme Overview

This section describes the main parts of the training scheme. Has the training been divided into several phases? Is the learning rate adapted over epochs? What method is used to tune hyper-parameters? All necessary steps to re-produce the training results should be documented here. Pointers to code for the different steps.

Pre-Training

This section describes any potential pre-training that was used to generate the solution. A detailed reference to the additional data set(s) is required as well as a documentation of all processing and training steps.

Loss Function

This section describes the loss function used to train the model(s).

Training Parameters

List all parameters that were tuned for the model training, and their values used for the submitted solution. This includes e.g. choice of optimizer, optimizer parameters, drop-out rates, learning rate scheduler etc.

Consistency in training results

This section describes the consistency of the training results. Examples of question to answer include; Is it easy to reproduce the results of the submitted solution? Is the output consistent over training runs?

Training time

This section includes the time it takes to train the model end-to-end. Please specify the hardware that was used for training.

Analysis of Model Performance/Output

This section describes the post-analysis the performance and model outputs that has been necessary to achieve the result on the submitted proposal. How has the output from the evaluation pipe line been connected to the model training scheme? How has the loss function been adapted to target the evaluation pipe line?

Model Execution End-to-End

This section describes how to set up and run the model from raw input to model output, both on training and inference. A step-by-step guide is recommended to avoid missing any steps by mistake.



Contact Details

- Open Zoom sessions Mon-Fri, 12.00 CET
 - <https://astrazeneca.zoom.us/j/92963367998?pwd=MDIha21TeEpJMTFWNXYweCtkaFpWdz09>
- Slack – Lars Tornberg
- Email: lars.tornberg@astrazeneca.com

