

TIAToolbox: Advanced Tissue Image Analytics

AI for Understanding Cancer Whole Slide Images

Johnathan Pocock, Simon Graham, Quoc Dang Vu, Mostafa Jahanifar, Srijay Deshpande, Giorgos Hadjigeorgiou, Adam Shephard, Raja Muhammad Saad Bashir, Mohsin Bilal, Wenqi Lu, David Epstein, Fayyaz Minhas, Nasir M. Rajpoot, Shan E Ahmed Raza



Background

The latest 2018 Royal College of Pathologists workforce census showed that only 3% of NHS histopathologists departments have enough staff. Furthermore, many pathology departments moving to a fully digital pathology (DP) workflow has helped to drive rapid growth in computational pathology (CPath) which utilises advanced deep-learning (DL) algorithms to perform common histopathology tasks.

CPath pipelines have the potential to be a powerful tool for consistent and reproducible analysis of large volumes of histopathology data. Automated analysis assistance tools can address the issue of high pathologist workload via manual analysis by reducing the need for highly trained clinicians to perform tedious tasks, such as cell counting. However, there is no open-source software library providing a generic end-to-end API for reproducible pathology image analysis using best practices. This restricts the development of advanced algorithms to specialist users. Furthermore, it is currently common to re-implement steps in the pipeline. To help overcome this bottleneck, we present TIAToolbox, a Python toolbox designed to make computational pathology accessible to computational, biomedical, and clinical researchers.

Reading Whole Slide Images

TIAToolbox provides a generic API for reading various whole slide image formats including: Aperio SVS, Omnyx JP2, DICOM, NGFF and many more.

```
from tiatoolbox.wsicore.wsireader import WSIReader
# 1. Create a reader object for the whole slide image
wsi = WSIReader.open(input_img="crc-sample.svs")
# 2. Decode part of the WSI
img = wsi.read_rect(
    location=(123, 123),
    size=(348, 134),
    resolution=0.5,
    units="mpp", # Microns-per pixel
)
```

Tissue Masking

We provide a simple API to mask out regions of no diagnostic value.

```
from tiatoolbox.wsicore.wsireader import WSIReader
from tiatoolbox.tools.tissuemask import MorphologicalMasker
wsi = WSIReader.open("crc-sample.svs")
masker = MorphologicalMasker()
mask, = masker.fit_transform([
    wsi.slide_thumbnail()
])
```

VirtualWSI

Our *VirtualWSI* class allows reading mask and tissue using the same coordinate space.

```
from tiatoolbox.wsicore.wsireader import WSIReader
tissue_wsi = WSIReader.open(input_img="crc-sample.svs")
# 1. Create a mask at 32 MPP as a VirtualWSI
mask_wsi = wsi.tissue_mask(resolution=32, units="mpp")
# 2. A region of both tissue and mask may be read using
# an identical coordinate space.
tissue, mask = (
    wsi.read_rect(
        location=(123, 123),
        size=(512, 512),
        resolution=0.5,
        units="mpp",
    )
    for wsi in (tissue_wsi, mask_wsi)
)
```

Stain Normalization

We implement several published stain normalisation methods, including: Reinhard, Ruifrok, Macenko, and Vahadane.

```
from tiatoolbox.tools import stainnorm
from tiatoolbox import data
from tiatoolbox.wsicore.wsireader import WSIReader
# 1. Set up our normaliser object
target = data.stain_norm_target()
normalizer = stainnorm.VahadaneNormalizer()
normalizer.fit(target)
# 2. Read an image region
wsi = WSIReader.open(input_img="crc-sample.svs")
sample = wsi.read_rect(
    location=(123, 123),
    size=(512, 512),
)
# 3. Apply stain normalisation
normalised = normalizer.transform(sample)
```

Patch Prediction

We include pre-trained models for patch-wise prediction including a nine class model colorectal tissue (Kather 100K) and binary model for breast cancer tissue (Cameylon16 Patch Cam).

```
from tiatoolbox.wsicore.wsireader import WSIReader
from tiatoolbox.models.engine.patch_predictor import PatchPredictor
wsi = WSIReader.open("crc-sample.svs")
predictor = PatchPredictor(
    pretrained_model="resnet18-kather100k",
    batch_size=32,
)
output = predictor.predict(imgs=[wsi], mode="wsi")
```

Semantic Segmentation

Our models are also capable of handling patch, visual field and WSI-scale inference including post-processing such as merging of results across overlapping tiles.

Nucleus Segmentation & Classification

Morphology derived from accurate nucleus segmentation in addition to cell classification is a valuable feature for downstream analysis. We include models pre-trained on several datasets for both cell segmentation and classification.

Deep Feature Extraction

Easy extraction of feature vectors from pre-trained models.

```
from shapely.geometry import Polygon
from tiatoolbox.annotation.storage import SQLiteStore
# Open a store on disk
store = SQLiteStore("annotations.db")
# Query the store for annotations
results = store.query(
    geometry=Polygon.from_bounds(0, 0, 10, 10),
    where="props['class'] == 4",
)
```

Annotation Storage

Compact storage and efficient querying of millions of annotation geometries.

Full Pipelines

Additionally, we provide implementations of full DL CPath pipelines via Jupyter notebooks.

Graph Analysis (SlideGraph+)

The SlideGraph+ (Wenqi *et al.*) pipeline for graph-based WSI classification can easily be fully reproduced using modules from our toolbox.

Slide Classification (IDaRS)

We also show replication of mutation prediction via an iteratively sampled bag of patches method, IDaRS (Bilal *et al.*).

Interactive Visualisation

We also have an interactive web-based viewer for slides, annotations and model predictions.

Full source code on GitHub:
<https://github.com/TissueImageAnalytics/tiatoolbox>

Paper preprint on bioRxiv:
<https://doi.org/10.1101/2021.12.23.474029>

Formatted documentation:
<https://tiae-toolbox.readthedocs.io/en/latest/>