# Complete Connexin Analysis Workflow Guide

## Table of Contents

# Overview

This comprehensive guide walks through the entire connexin analysis workflow, from raw .tiff images to final statistical analysis. The workflow combines QuPath for image analysis with custom Python scripts for advanced connexin characterization.

---

# Prerequisites and Setup

## Software Requirements

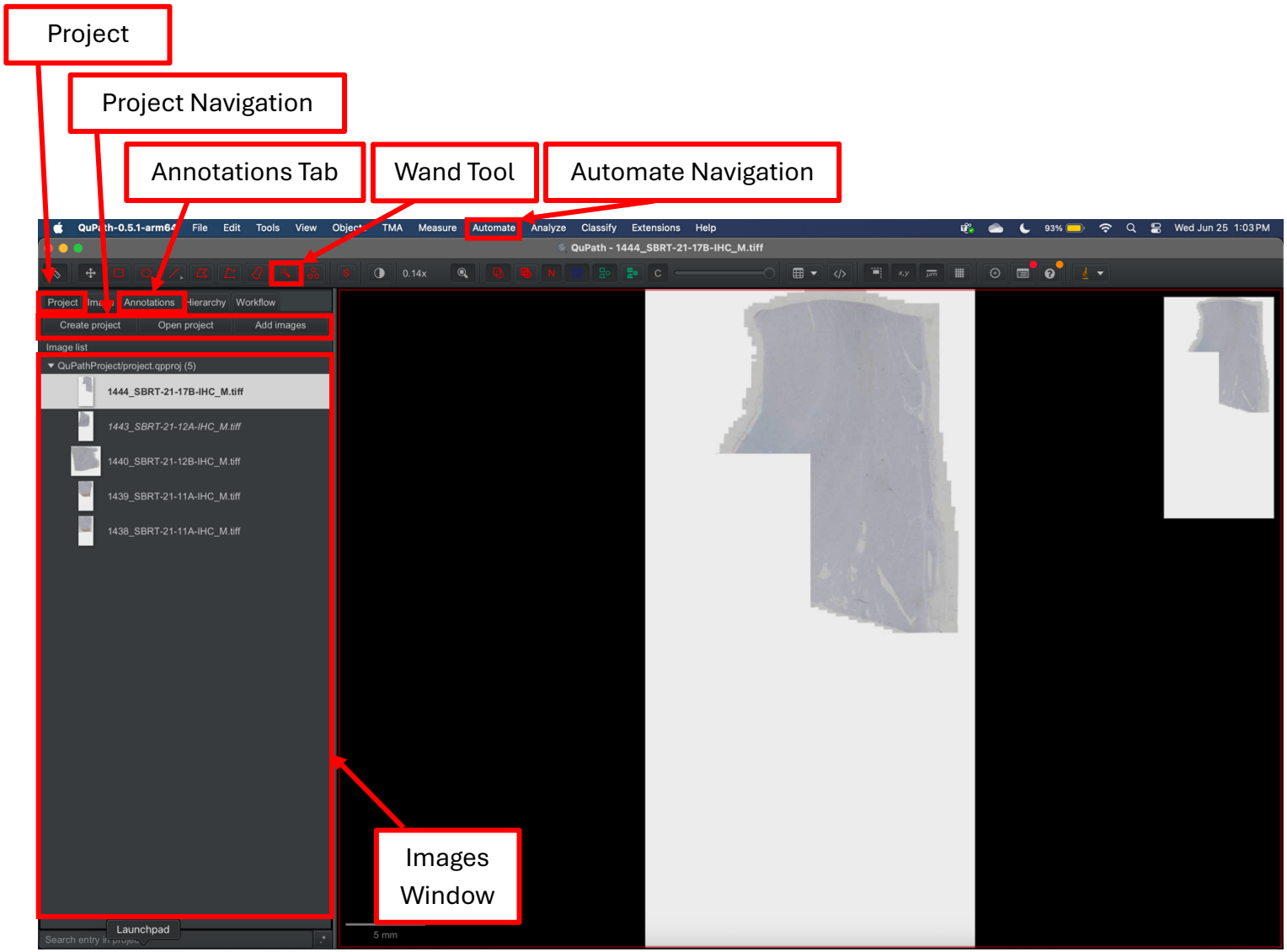
- **QuPath 0.4.0 or later** - For image analysis and object detection
- **Python 3.8+** - For advanced analysis pipeline
- **Required Python packages**:
- `pip install pandas numpy matplotlib seaborn scipy scikit-learn tkinter`

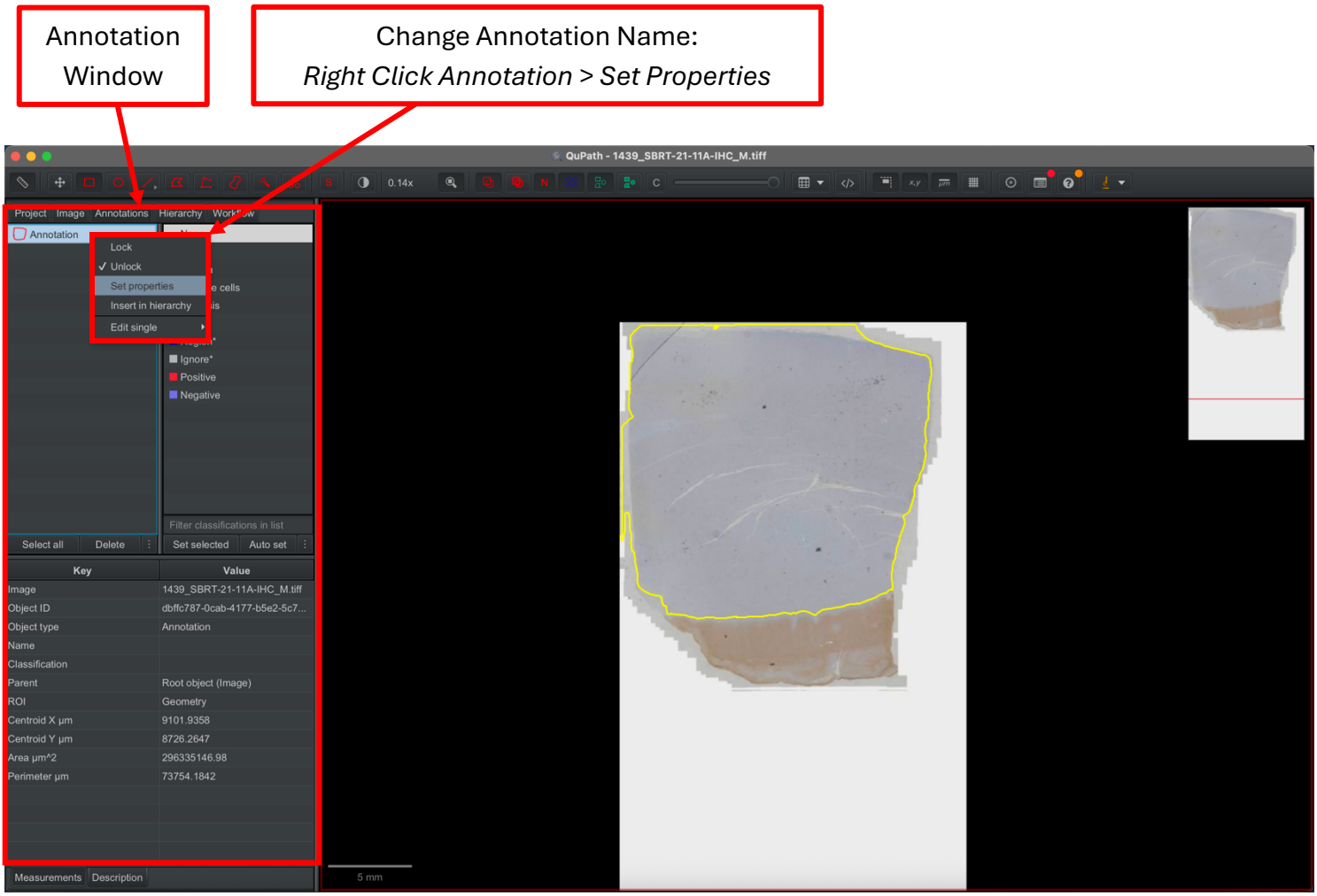
## Hardware Recommendations

- **RAM**: Minimum 16GB (32GB+ recommended for large images)
- **Storage**: SSD recommended for faster image loading
- **CPU**: Multi-core processor for parallel processing

---

# QuPath






# 1. QuPath Project Setup

## Creating a New Project

I. **Launch QuPath**
   - Open QuPath application
   - Navigate to **File → Project → Create project**

II. **Project Configuration**
   - Open Project *or* Create Project
     1. Create Project:
        - Choose project directory (directory must be empty)
        - Name your project
        - From Desktop/QuPathProjectResources, copy folders *scripts* and *classifiers* into new QuPath project directory
     2. Open Project:
        - Select the *project.qpproj* file within desired project directory

# 2. Image Import and Organization

## Importing .tiff Images

I. **Image Naming is Important**
   - Use meaningful naming conventions

     ** Naming of image files is important **

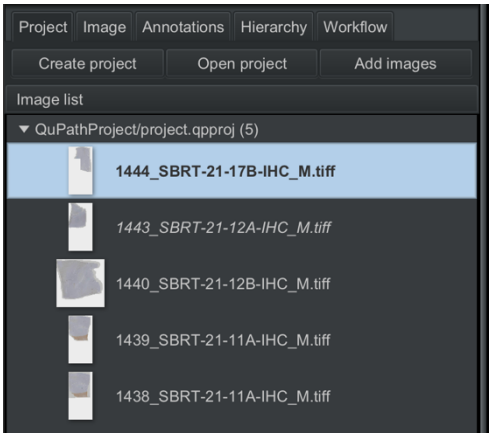
     <image_number>_<study>-<subject>-<sample>-IHC.tiff

     Ex: 1349_SBRT-11-5A.tiff
     Image number = 1349
     Study = SBRT
     Subject = 11
     Sample = 5A

     ** Using this naming convention allows analysis scripts to iterate appropriately!

II. **Add Images to Project**
   - Right-click in the **Project** tab
   - Select **Add images**
   - Navigate to your images folder
   - Select all .tiff files for analysis

III. **Image Organization**
   - Images will appear in the Project tab

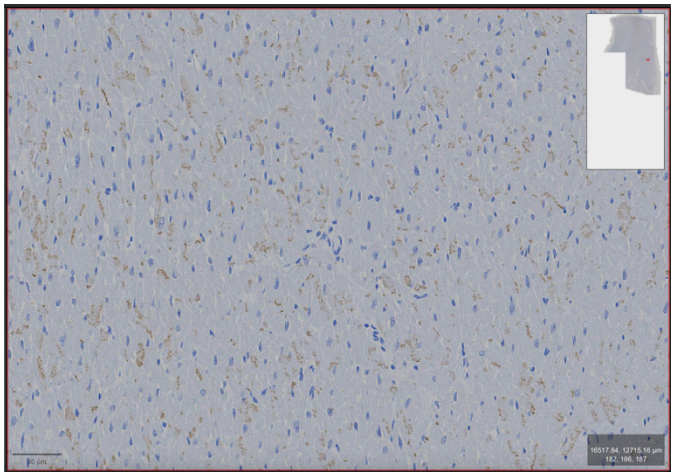
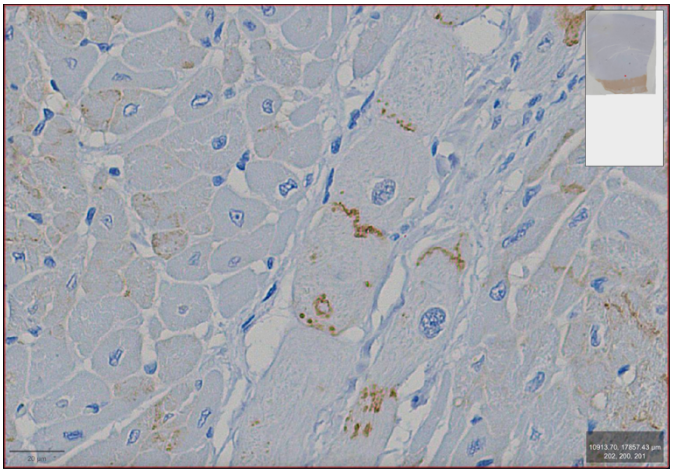
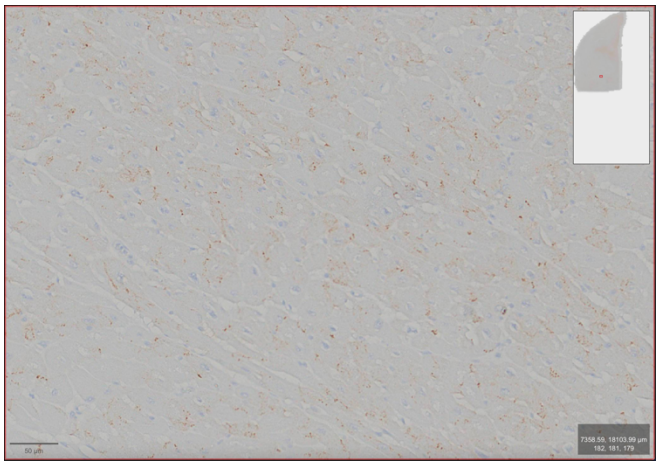
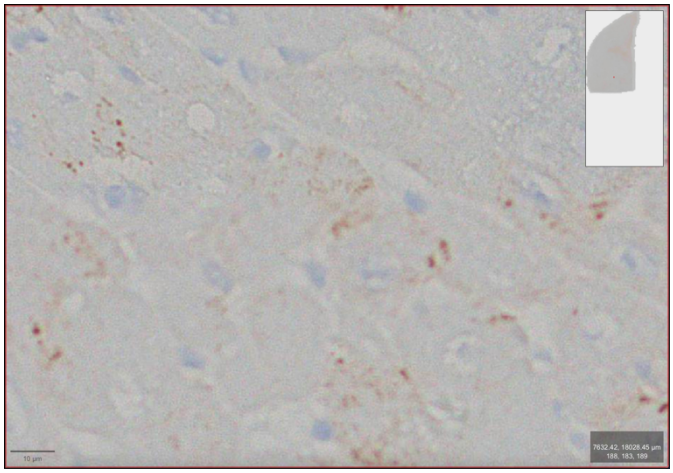



## Image Quality Assessment

ASSES EACH IMAGE FOR QUALITY to guide thresholding selection and areas to review post-detection:

- **Focus quality**: Ensure nuclei are sharp and well-defined
- **Staining intensity**: Adequate DAB staining for connexin detection

- **Tissue integrity**: Minimal artifacts or damage
- **Background**: Clean background for optimal detection

| Good Image Quality – Can start w/ high threshold (0.08 *or* 0.10) | Poor Image Quality – Start with lower threshold but balance over-detecting non-nuclei/connexins |
|---|---|
|   |   |

## 3. Tissue Annotation

### Manual Tissue Annotation with Wand Tool

I. **Open Image for Annotation**
   - Double-click an image in the Project tab
   - Image opens in the main viewer

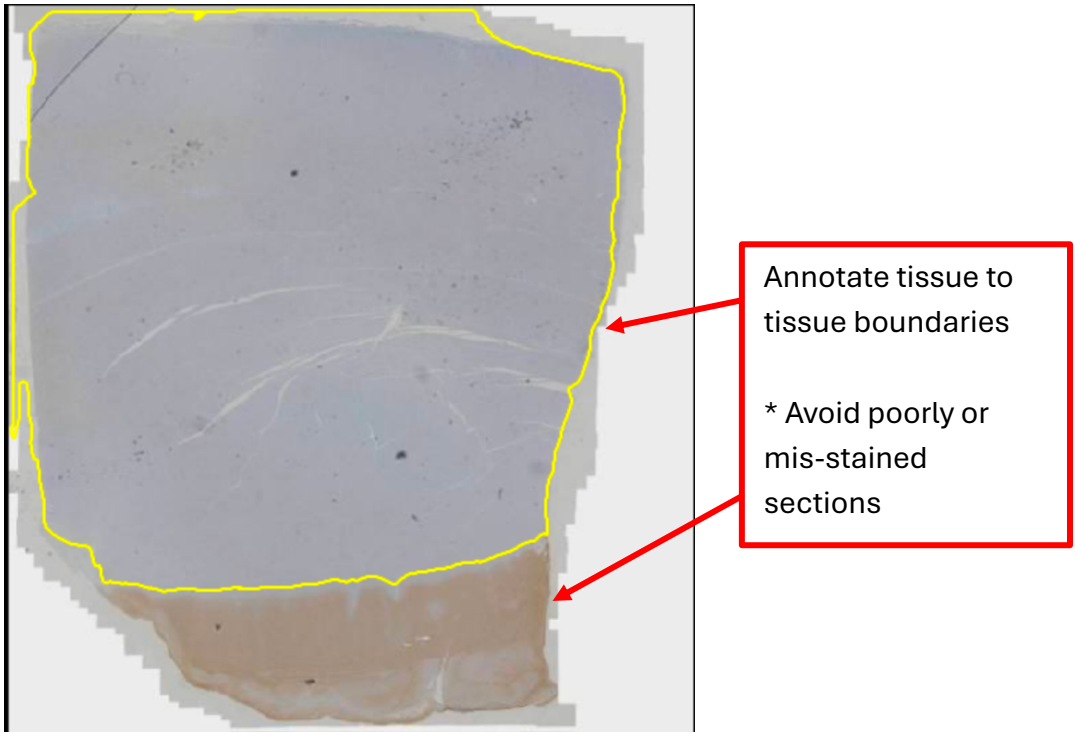
II. **Select Annotation Tool**
   - In the toolbar, click the **Wand tool** (magic wand icon)
   - Alternatively: **Tools → Wand**

**Option: Configure Wand Settings** (Not Usually Necessary)

- **Tolerance**: Start with 10-20, adjust based on tissue contrast
- **Region**: Select "Add to existing objects" if annotating multiple areas
- **Smoothing**: Enable for cleaner boundaries

III. **Annotate Tissue Areas**

- Click on tissue regions to select them
- The wand will automatically detect similar regions
- **Emphasize catching tissue boundaries. Algorithm will fill holes in the annotation automatically**
- **Tips**:
  1. Work on one tissue section at a time
  2. Avoid including large artifacts or folded tissue
  3. Include representative areas of the entire tissue

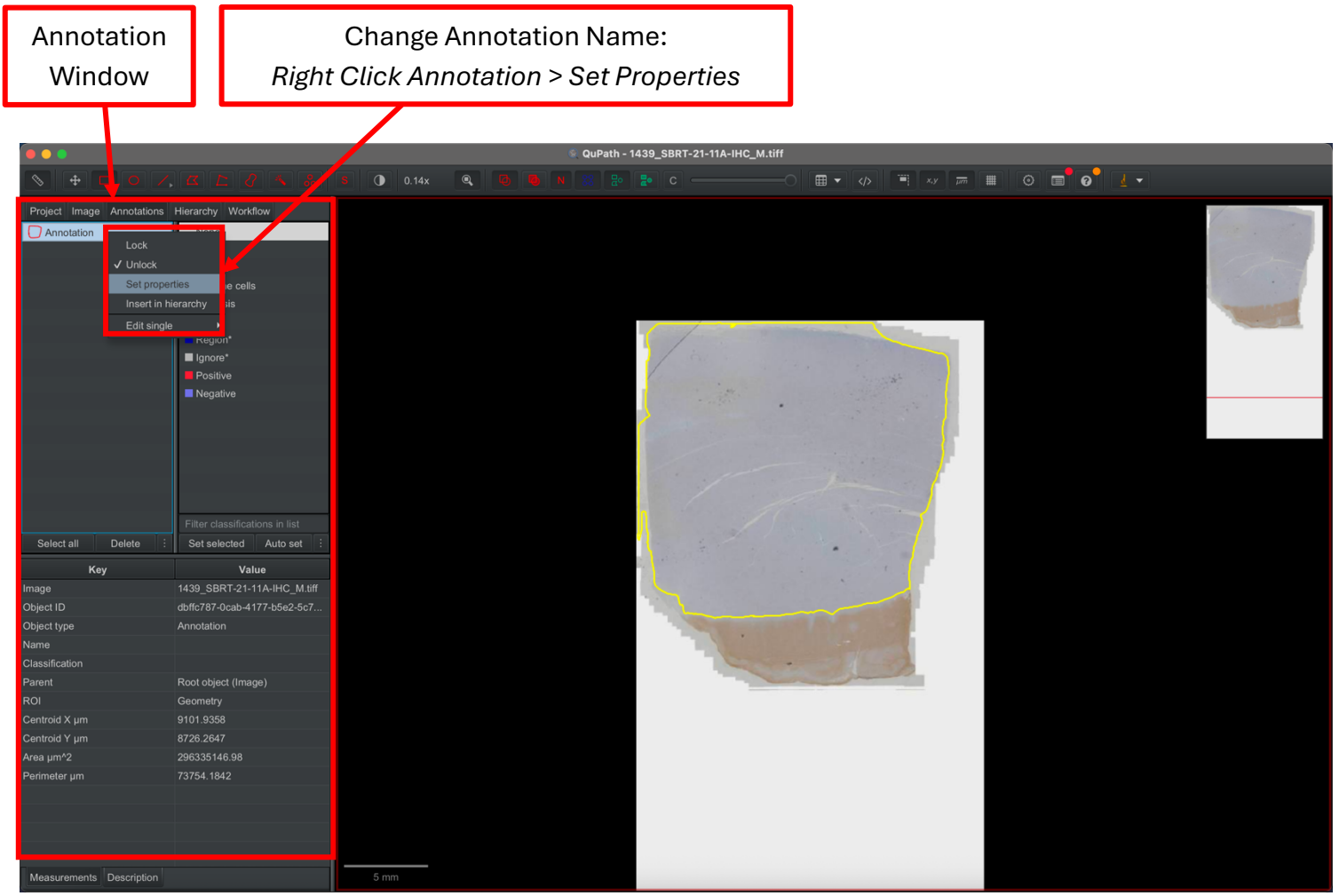



IV. **Refine Annotations** (If Needed)

- Use **Edit → Delete** to remove unwanted selections
- Use **Polygon tool** for manual refinement if needed
- Ensure annotations capture viable cardiomyocyte tissue

V. **Name the Annotation (IMPORTANT)**

- With annotation selected, go to **Annotations** tab
- Right click on Annotation > Set Properties > Change **Name** to "Tissue"
- This name is critical for the analysis pipeline




## Quality Control for Annotations

I. **Check Each Image Annotation**

- **Coverage**: Annotations should represent the tissue adequately
- **Exclusions**: Avoid blood vessels, artifacts, and damaged regions
- **Consistency**: Use similar annotation strategies across all images

# 4. Automation: Cell Detection + Classification

## Understanding Detection Parameters (Process is Automated, Review if Customizing Own Cell Detection)

The cell detection uses a watershed algorithm with these key parameters:

**Core Detection Settings**

- **Detection image**: Optical density sum (combines both stains)
- **Requested pixel size**: 0.5 µm (balances speed vs. resolution)
  - For improved detection, can run at 0.2 µm. (Will take 3x time)
- **Background radius**: 3.0 µm (local background estimation)
  - Increase if image background is non-white
- **Median filter radius**: 0.0 µm (noise reduction)
- **Sigma**: **VARIABLE** 1.0, 1.5, 2.0 µm (Gaussian smoothing for detection)

**Size and Threshold Constraints**

- **Min area**: 1.0 µm² (excludes tiny artifacts)
- **Max area**: 200.0 µm² (excludes large, merged objects)
- **Threshold**: **VARIABLE** (0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10)
- **Max background**: 2.0 (background normalization)

**Post-processing Options**

- **Watershed post-process**: true (separates touching cells)
  - Turn to *false* if connexin or cell plaques are multi-segmented
- **Exclude DAB**: false (includes DAB-positive regions)
- **Cell expansion**: 0.0 µm (no cytoplasm expansion)
- **Include nuclei**: true (creates nuclear objects)
- **Smooth boundaries**: true (cleaner object boundaries)

## Understanding the IHC Classifier

The classifier distinguishes between:

- **CardiomyocyteNuclei**: Nuclear objects representing cardiomyocyte cells
- **Connexin**: DAB-positive objects representing connexin gap junctions

### Classification Criteria

The classifier uses these features:

1. **Morphological features**: Size, shape, compactness
2. **Intensity features**: DAB and hematoxylin staining intensity
3. **Spatial features**: Context within tissue

## Understanding the Lateralization Classifier

*** ONLY to be applied to images utilizing Cell Detection w/ Post Processing**

The classifier distinguishes between:

- **CardiomyocyteNuclei**: Nuclear objects representing cardiomyocyte cells
- **LateralConnexin**: DAB-positive objects representing lateralized connexins
- **IntercalatedConnexin**: DAB-positive objects representing connexins at intercalated discs

### Classification Criteria

The classifier uses these features:

4. **Morphological features**: Size, shape, compactness
5. **Intensity features**: DAB and hematoxylin staining intensity
6. **Spatial features**: Context within tissue

## Choosing the Right Detection Script

****Bottom Line: If over-segmenting is occurring (one connexin or nuclei is identified as multiple different objects), run equivalent threshold w/ post processing to attempt to combine nearby detections. Only run lateralization classifier on post-processing ran images as it is only trained on these classification patterns.****

You have 11 detection scripts with different threshold/smoothing/post-processing values. Explanation of variations:

- Threshold: Pixel intensity threshold, adjustable to account for variations in quality of image and contrast
- Smoothing: Gaussian smoothing function applied to weighting of neighboring pixels in image to determine decay of intensity of pixels and create boundary
- Post-Processing: Allows detection-by-detection growth and shrinking to connect neighboring detected objects

**Standard Scripts:**

| Threshold | Best For | Image Characteristics |
|---|---|---|
| **0.10** | Highest-quality staining | Very strong nuclear contrast, minimal background |
| **0.09** | High-quality staining | Strong nuclear contrast, minimal background |
| **0.08** | Moderate-quality staining | Strong nuclear contrast, moderate background |
| **0.07** | Standard staining | Good nuclear definition, moderate background |
| **0.06** | Moderate staining | Adequate contrast, some background noise |
| **0.05** | Poor staining | Weak staining, higher background |
| **0.04** | Very poor staining | Very weak staining, significant background |

| Post-Processing Scripts (combines smaller plaques/nuclei segments together): | | |
|---|---|---|
| **Threshold w/ Post Processing** | **Best For** | **Image Characteristics** |
| **0.10 w/PostProc.** | Highest-quality staining | Very strong nuclear contrast, minimal background |
| **0.08 w/PostProc.** | Moderate-quality staining | Strong nuclear contrast, moderate background |
| **0.06 w/PostProc.** | Moderate staining | Adequate contrast, some background noise |
| **0.04 w/PostProc.** | Very poor staining | Very weak staining, significant background |

Example images w/ Applied Detection + Corrections

| Image Appearance | Suggested Detection Algorithm | Cell Detection Results |
|---|---|---|
| | | |

## Running Cell Detection

I. **Choose Detection Script**
   - **Automate → Project Scripts → Directly Select Script to run**
   - Scripts will populate

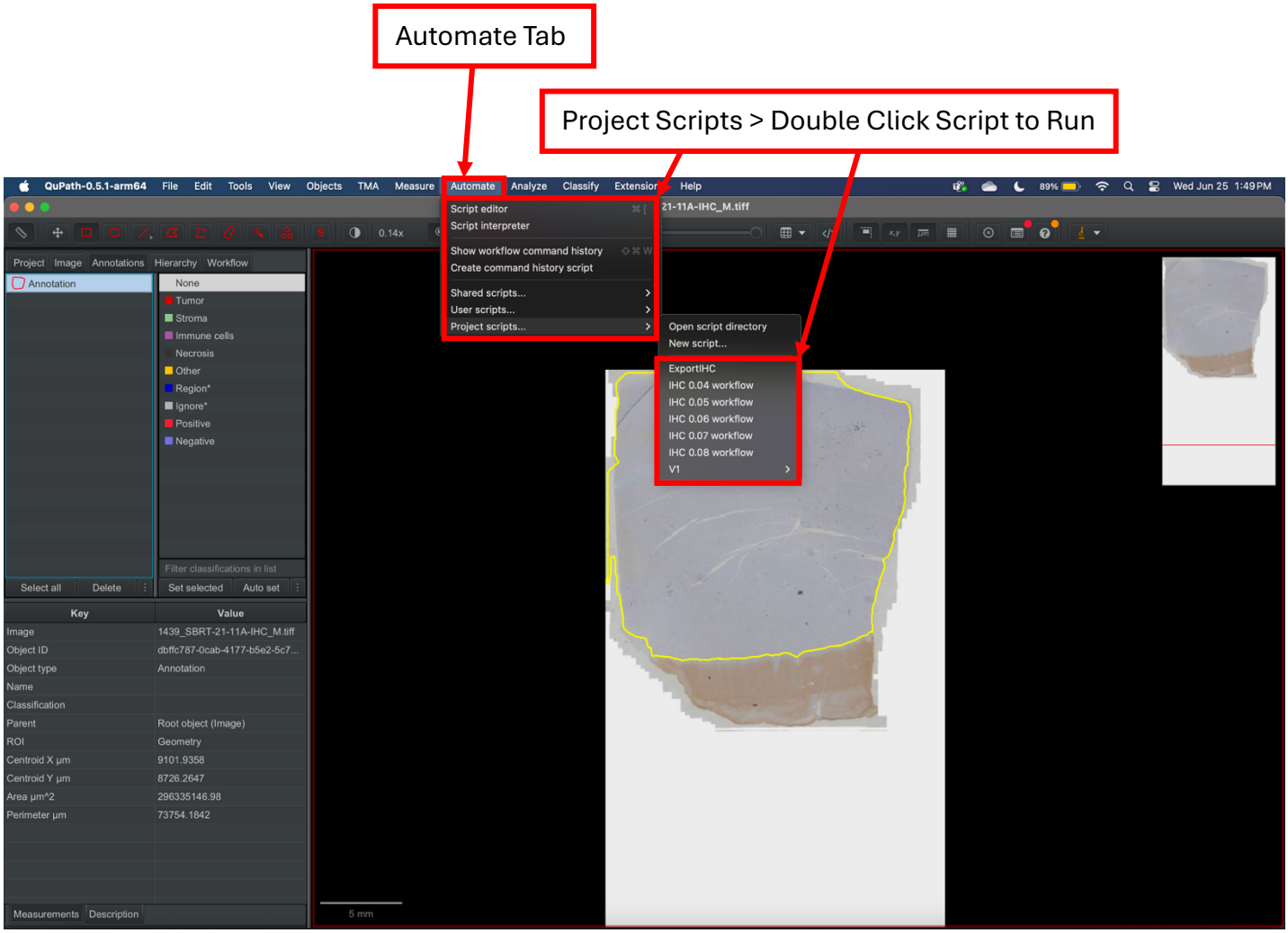
     IHC 0.04 w/ Post Processing workflow
     IHC 0.04 workflow
     IHC 0.05 workflow
     IHC 0.06 w/ Post Processing workflow
     IHC 0.06 workflow
     IHC 0.07 workflow
     IHC 0.08 w/ Post Processing workflow
     IHC 0.08 workflow
     IHC 0.09 workflow
     IHC 0.10 w/ Post Processing workflow
     IHC 0.10 workflow
     Lateralization Classifier*
     Export_IHC.groovy *(This one is for later)*

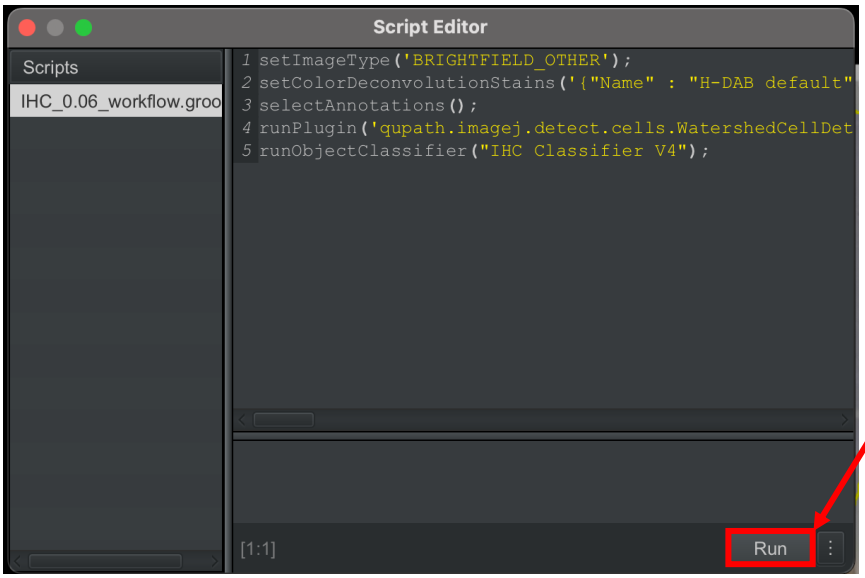
     *** Only use Lateralization Classifier if ran a Post Processing workflow!**

     **If no scripts populate:**
     - All scripts are found in Desktop/QuPathProjectResources
     - Drag and Drop the files into the QuPath Window

   - Based on your image quality assessment, select appropriate script

* Suggested to **start at 0.08** Threshold and adjust based on performance
* Suggested to also **try 0.08 w/ Post Processing** to compare performance




New window will pop up. Select Run

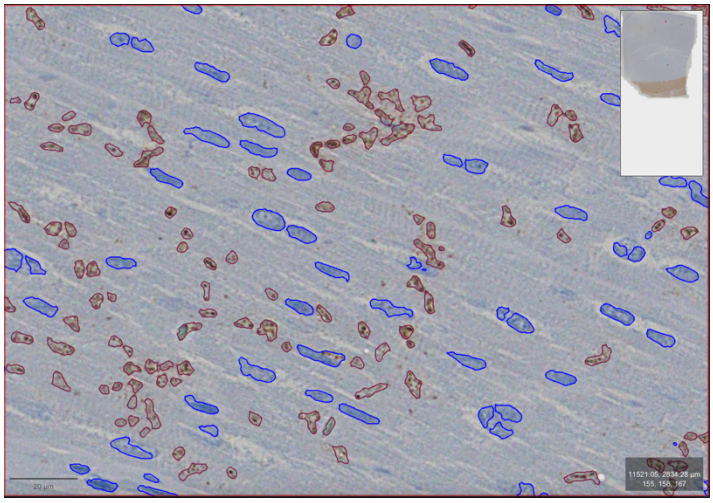
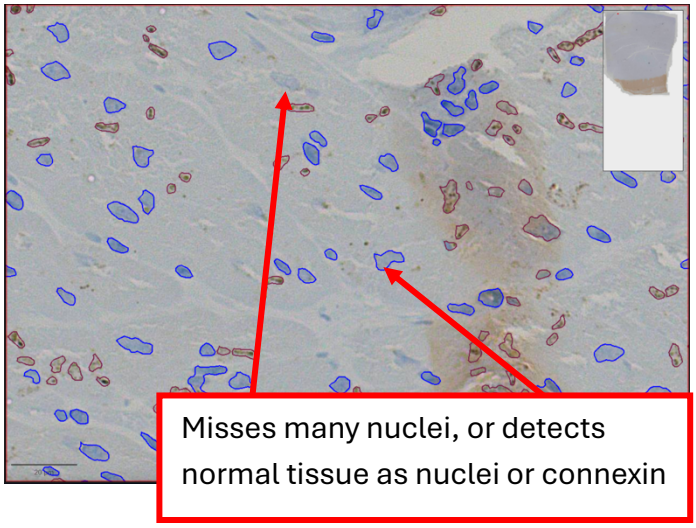
II. **Run Detection Script**
    - Click **Run** button in the bottom right corner of the script editor window

III. **Record** (Possibly on IHC_Master_File) **what script and/or settings were used for optimal detection on each image for references and reproducibility**

## Detection Quality Assessment

After running detection, evaluate:

1. **Detection Coverage**
    - Are most nuclei detected?
    - Are there significant gaps in detection?
    - Are there cell boundaries or non-nuclei being counted as nuclei?
2. **Over-detection**
    - Are background artifacts being detected?
    - Are there too many small objects?
3. **Under-detection**
    - Are faint but real nuclei missed?
    - Are large nuclei split inappropriately?

| Appropriate Cell Detection | Poor Cell Detection |
|---|---|
|  |   |

| Detection Challenge | Suggested Correction | Cell Detection Results |
|---|---|---|
| | | |

### Save Image: File > Save

### Adjusting Detection Parameters

If detection quality is poor:

1. **For over-detection**: Increase threshold or min area
2. **For under-detection**: Decrease threshold
3. **For background noise**: Increase background radius
4. **For merged cells**: Enable watershed post-processing

### Manual Classification Review

After automatic classification:

1. **Review Results**
    - Check classification accuracy
    - Look for obvious misclassifications

      ** If obvious misclassification, take note of the slide and we will train a new classifier. Currently using IHC Classifier V4. **

2. **Manual Corrections** (if needed)
    - Select misclassified objects
    - Right-click → **Set class**
    - Choose correct classification

Classification Quality Metrics

Good classification should show:

- **CardiomyocyteNuclei**: Round/oval, moderate size, hematoxylin-positive
- **Connexin**: Small, irregular, DAB-positive
- **Minimal misclassification**: <5% error rate

## 5. Data Export: Complete After Analyzed Every Image In Project

### Using the Export Script

I. **Open Script Editor**
    - **Automate → Project Scripts**
II. **Select Export_IHC.groovy**

      Export_IHC.groovy

III. **Choose Export Directory**
    - The script will prompt for output directory
    - Choose your desired folder

## Understanding the Export Process

The export script will:

1. **Process All Images**: Iterates through entire project
2. **Extract Data**: For each image:
    - Annotation data (tissue boundaries and measurements)
    - Detection data (all objects with classifications and measurements)
3. **Generate Files**: Creates paired CSV files:
    - `[ImageName]-Annotation.csv`
    - `[ImageName]-Detection.csv`

## Understanding Export Data

### Annotation CSV Columns

- **Image**: Source image name
- **Object ID**: Unique identifier
- **Object type**: Annotation
- **Name**: "Tissue"
- **Classification**: Tissue type
- **ROI**: 'Polygon'
- **Centroid X/Y µm**: Center coordinates
- **Area µm²**: Total tissue area
- **Perimeter µm**: Tissue perimeter

### Detection CSV Columns

- **Image**: Source image name
- **Object ID**: Unique identifier
- **Object type**: Detection
- **Name**: Object name
- **Classification**: CardiomyocyteNuclei or Connexin
- **Parent**: Parent annotation
- **ROI**: 'Polygon'
- **Centroid X/Y µm**: Object center
- **Nucleus measurements**: Area, perimeter, circularity, etc.

# Python

## Pipeline Overview

The Python analysis consists of three main components:

1. **connexin_top_level.py**: Main controller
2. **advanced_connexin_analysis_batched.py**: Individual sample analysis
3. **connexin_data_analysis.py**: Results compilation

## Setting Up Analysis (If necessary, should be complete)

1. **Prepare Directory Structure: All .py Files and Settings should be in the same folder**

```
Analysis_Scripts/
├── connexin_top_level.py
├── advanced_connexin_analysis_batched.py
├── connexin_data_analysis.py
├── IHC_settings.json
└── ExportIHC.groovy
```

8. **Configure Settings**
    - Edit `IHC_settings.json` with your specific parameters
    - Update file paths to match your system

## 1. Configure IHC_Master_File.csv

I. **Create or Copy over your IHC_Master_File.csv to the Desktop (IMPORTANT)**
    - File should be a CSV exported from Excel or other program with the following columns:

| slide_name | Fibrosis Percentage | Notes |
|---|---|---|
| Ex: 11-5A | Ex: 34.44 | Ex: Infarct |

slide_name: Identification of slide being analyzed
*** Important that the column name matches <slide_name> exactly**
Fibrosis Percentage: *Not required*
Notes: Comments on slide gross pathology location *or* notes on tissue state (healthy, infarct, etc.)
* Can include any additional columns desired

Example IHC_Master_File.csv:

| | A | B | C | D |
|---|---|---|---|---|
| 1 | slide_name | Fibrosis Percentage | Notes | Infarction |
| 2 | 09-4B | 73.08 | Center | yes |
| 3 | 09-4C | 3.64 | Healthy | no |
| 4 | 09-5A | 50.49 | center | yes |
| 5 | 09-5B | 68.64 | center | yes |
| 6 | 09-5C | 4.83 | Healthy | no |
| 7 | 09-6A | 59.99 | Center | yes |
| 8 | 09-6B | 66.55 | Center | yes |
| 9 | 09-6C | 2.62 | Healthy | no |
| 10 | 09-7A | 59.74 | Center | yes |
| 11 | 09-7B | 65.6 | Center | yes |

- Save file to Desktop

## 2. Configure Settings

II. **Open IHC_Settings.json File on Desktop** (Not usually necessary)
- Edit specific parameters
- Save file

### Understanding Analysis Settings

**Key Parameter Explanations**

**Memory Management Parameters:**

- `total_objects_threshold`: When to start subsampling large datasets
- `processing_chunk_size`: Size of data chunks for memory efficiency
- `max_nuclei_subsample` / `max_connexins_subsample`: Limits for very large datasets

**Analysis Parameters:**

- `max_association_distance_um`: Maximum distance to associate connexins with nuclei (typically 25-50 μm)
- `cell_radius_multiplier`: Factor to estimate cell size from nucleus (3-4x typical)
- `grid_size_um`: Size of spatial analysis grid (50-200 μm depending on resolution needed)

**Output Control:**

- `expanded_analysis`: Enable comprehensive lateralization and remodeling analysis
- `create_visualizations`: Generate detailed plots (memory intensive)
- `suppress_console_output`: Reduce verbose output for batch processing

## 3. Running the Analysis

I. **Click IHC Analysis Desktop Icon**
   - Pipeline runs automatically

### Understanding Analysis Workflow (Python Handles These Steps)

**Step 1: File Discovery and Validation**

- Finds all `*Detection.csv` files
- Validates corresponding `*Annotation.csv` files exist
- Reports file counts and any missing files

**Step 2: Individual Sample Analysis**

For each sample, the pipeline:

1. **Data Loading**
   - Loads Detection and Annotation CSV files
   - Validates required columns and data format
   - Separates cardiomyocyte nuclei from connexin objects
2. **Basic Metrics Calculation**
   - Total counts (nuclei, connexins)
   - Area measurements (individual and total)
   - Density calculations (connexins per cell, per tissue area)
3. **Association Analysis** (if expanded analysis enabled)
   - Associates each connexin with nearest nucleus within threshold distance
   - Calculates spatial relationships
   - Determines connexin distribution patterns
4. **Advanced Analysis** (if expanded analysis enabled)
   - **Lateralization Analysis**: 3 different methods to assess connexin distribution
   - **Spatial Heterogeneity**: Grid-based analysis of spatial variation
   - **Remodeling Scores**: Composite scores indicating pathological changes

**Step 3: Results Compilation**

- Combines all individual analysis results
- Merges with master experiment file (if available)

- Performs statistical analysis across samples
- Generates group comparisons and summaries

## Understanding Analysis Outputs

### Per-Sample Outputs

1. **Analysis Report** (`[SampleName]_analysis_report.json`)
   - Comprehensive metrics in structured JSON format
   - Includes all calculated measurements and statistics
2. **Processed Data** (`[SampleName]_processed_nuclei.csv`, `[SampleName]_processed_connexins.csv`)
   - Enhanced CSV files with calculated metrics
   - Additional columns for lateralization, association, etc.
3. **Visualizations** (`[SampleName]_advanced_connexin_analysis.png`) - if enabled
   - Multi-panel plots showing distributions and relationships
   - Spatial maps and statistical summaries

### Compiled Outputs

1. **Compiled Results** (`compiled_connexin_analysis_results.csv`)
   - All sample metrics in single spreadsheet
   - Ready for statistical analysis in R/SPSS/etc.
2. **Statistical Summary** (`statistical_analysis.json`)
   - Descriptive statistics across all samples
   - Group comparisons if metadata available
3. **Pipeline Report** (`pipeline_summary_report.json`)
   - Execution summary with success/failure rates
   - Processing times and any errors encountered

# Results Interpretation – IHC_Master_File_Results.csv

## Basic Metrics

### Count Metrics

- **Total Nuclei**: Number of cardiomyocyte nuclei detected
- **Total Connexins**: Number of connexin objects detected
- **Connexins per Cell**: Average connexin count per cardiomyocyte

*Normal ranges vary by species and tissue region*

### Area Metrics

- **Mean Plaque Size**: Average area of individual connexin plaques
- **Total Connexin Area**: Sum of all connexin plaque areas
- **Connexin Area per Cell**: Average connexin area per cardiomyocyte

**Density Metrics**

- **Connexin Density**: Connexins per unit tissue area
- **Nuclei Density**: Cardiomyocytes per unit tissue area
- **Connexin/Tissue Ratio**: Proportion of tissue area occupied by connexins

## Advanced Metrics (Expanded Analysis)

**Lateralization Index**

**What it measures**: Distribution of connexins around cell perimeter

- **<15%**: Normal, connexins distributed around entire perimeter
- **15-30%**: Mild lateralization, some redistribution
- **>30%**: Severe lateralization, connexins concentrated on lateral surfaces

**Clinical significance**: Lateralization often indicates early remodeling

**Remodeling Score**

**What it measures**: Composite score combining multiple pathological indicators

- **<0.5**: Mild remodeling
- **0.5-1.0**: Moderate remodeling
- **>1.0**: Severe remodeling

**Components include**: Lateralization, plaque size changes, density alterations, cell shape changes

**Spatial Heterogeneity Index**

**What it measures**: Coefficient of variation across tissue regions

- **<40%**: Uniform distribution
- **40-80%**: Moderate heterogeneity
- **>80%**: High heterogeneity (patchy distribution)

**Clinical significance**: High heterogeneity may indicate regional remodeling

### Statistical Considerations

**Sample Size**

- **Minimum**: 5-10 images per group for basic metrics
- **Recommended**: 15-20 images per group for robust statistics
- **Power analysis**: Consider effect size for study design

**Multiple Comparisons**

- Apply appropriate corrections (Bonferroni, FDR) when testing multiple metrics
- Focus on pre-specified primary endpoints

**Biological vs. Statistical Significance**

- Consider biological relevance of observed differences
- Small p-values don't always indicate meaningful changes

# Troubleshooting

## Common QuPath Issues

**Poor Cell Detection**

**Symptoms**: Missing nuclei, over-detection of artifacts **Solutions**:

- Adjust threshold (try different detection scripts)
- Check image quality and staining
- Modify min/max area constraints
- Improve tissue annotation to exclude artifacts

**Classification Errors**

**Symptoms**: Connexins classified as nuclei or vice versa **Solutions**:

- Retrain classifier with more examples
- Manual correction of obvious errors
- Check staining quality and color deconvolution

**Memory Issues**

**Symptoms**: QuPath crashes or becomes unresponsive **Solutions**:

- Increase QuPath memory allocation
- Process smaller image regions
- Close unnecessary applications

## Common Python Analysis Issues

### File Not Found Errors

**Symptoms**: "Detection.csv files not found" **Solutions**:

- Check export completed successfully
- Verify file naming convention
- Ensure all Detection files have corresponding Annotation files

### Memory Errors

**Symptoms**: "MemoryError" or system becomes unresponsive **Solutions**:

- Reduce `max_nuclei_subsample` and `max_connexins_subsample`
- Increase `processing_chunk_size`
- Set `expanded_analysis: false` for basic analysis only
- Process fewer files at once

### Settings File Issues

**Symptoms**: "Settings file not found" or analysis uses wrong parameters **Solutions**:

- Ensure `IHC_settings.json` is in same directory as scripts
- Check JSON syntax (use online validator)
- Verify file paths use correct format for your OS

### Association Rate Issues

**Symptoms**: Very low association rates (<30%) **Solutions**:

- Increase `max_association_distance_um`
- Check if tissue annotation is named "Tissue"
- Verify connexin and nuclei are properly classified

## Data Quality Issues

### Inconsistent Results

**Symptoms**: High variability between similar samples **Solutions**:

- Standardize staining protocols
- Use consistent detection thresholds
- Ensure similar tissue regions are analyzed
- Check for batch effects in staining

**Unrealistic Values**

**Symptoms**: Extremely high/low connexin counts or areas **Solutions**:

- Review detection parameters
- Check for classification errors
- Validate against manual counting
- Consider tissue-specific normal ranges

## Performance Optimization

**Slow Analysis**

**Solutions**:

- Enable parallel processing
- Use SSD storage for data files
- Increase system RAM
- Process smaller batches
- Set `suppress_console_output: true`

**Large File Sizes**

**Solutions**:

- Reduce image resolution if appropriate
- Use subsampling for very large datasets
- Compress results files after analysis
- Archive raw data separately

# Best Practices

## QuPath Workflow

1. **Consistent annotation**: Use similar tissue selection criteria
2. **Quality control**: Review detection and classification results
3. **Documentation**: Record which detection threshold used for each image
4. **Batch processing**: Process similar images with same parameters

## Analysis Pipeline

1. **Settings management**: Use version-controlled settings files
2. **Data backup**: Keep copies of original CSV exports

3. **Quality metrics**: Monitor association rates and processing statistics
4. **Result validation**: Spot-check results against manual assessment

### Statistical Analysis

1. **Pre-registration**: Define analysis plan before data collection
2. **Multiple comparisons**: Apply appropriate corrections
3. **Effect sizes**: Report effect sizes along with p-values
4. **Visualization**: Create clear, interpretable plots

## Methods Description

Describe how slides were stained first.

The slides were scanned using the MorphoLens 6 slide scanner (Morphle Labs Inc., New York, NY, USA). The images were and analyzed in QuPath v0.5.1 [reference below]. Cell nuclei and connexin detection was performed using a watershed cell detection algorithm with various Gaussian blur filters, radii included for background intensity estimation, and pixel intensity thresholds depending on slide staining quality. Post processing utilizing either a secondary watershed transform or one cycle of dilution and erosion was implemented in cases of over segmentation. A random forest model was trained using 3-4 randomly identified tiles in 8 images where a combined 2,400 nuclei and connexins were manually classified by consensus among research staff to classify each watershed-detected object. The QuPath detections were then exported and analyzed in Python.

Qupath reference: Bankhead, P. et al. QuPath: Open source software for digital pathology image analysis. *Scientific Reports* (2017).
https://doi.org/10.1038/s41598-017-17204-5