

|  |
| --- |
| **Multi – Omics 4** |

|  |  |
| --- | --- |
| **2541** | **Project Number:** |

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Student: | Lior Segal | ID: |  |
| Student: | Dor Fried | ID: |  |

|  |  |
| --- | --- |
| Supervisor: | Prof. Ilan Tsarfaty |

|  |  |
| --- | --- |
| Project Carried Out at: | Faculty of Medical and Health Sciences – Tel Aviv University |

Table of Contents

[Abstract 3](#_Toc179653913)

[1 Introduction 5](#_Toc179653914)

[2 Theoretical background 7](#_Toc179653915)

[2.1 Color Deconvolution and Brightfield H&E 7](#_Toc179653916)

[2.2 Image Segmentation with improved Watershed Algorithm 8](#_Toc179653917)

[2.3 Smooth features, full width at half maximum (FWHM) 11](#_Toc179653918)

[2.4 Intensity measurements, Hearlick textures 12](#_Toc179653919)

[2.5Delaunay Clustering (Incremental Algorithm for Delaunay Triangulation) 13](#_Toc179653920)

[2.6QTL Analysis theoretical background 14](#_Toc179653921)

[3 Implementation 16](#_Toc179653922)

[3.1 Smart Data Management Tool 16](#_Toc179653923)

[3.2 Classic image processing 17](#_Toc179653924)

[3.3 QTL Analysis 19](#_Toc179653925)

[4 Analysis of results 21](#_Toc179653926)

[4.1 reproduce MultiSurv 21](#_Toc179653927)

[4.2 Smart Data Management Tool 22](#_Toc179653928)

[4.3 Classical image processing 24](#_Toc179653929)

[4.4 HAPPY to R/qtl2 upgrade 27](#_Toc179653930)

[5 Conclusions and further work 30](#_Toc179653931)

[5.1 Pipeline Conclusions 30](#_Toc179653932)

[5.2 Future Considerations for Multi-omics 31](#_Toc179653933)

[6 Project Documentation 33](#_Toc179653934)

**List of figures**

[Figure 1: Block diagram 4](#_Toc135313863)

[Figure 2: Example WSI 8](#_Toc135313864)

[Figure 3: Watershed topography illustration 10](#_Toc135313865)

Figure 4: Full Width at Half Maximum (FWHM) 11

Figure 5: General form of gray-tone spatial-dependence matrix for image 12

Figure 6: Incremental Delaunay triangulation 14

Figure 7: Data flow in the pipeline (HW) 17

Figure 8: Classic image processing data flow 18

Figure 9: QTL Analysis data flow 21

Figure 10: Graph of nucleus count vs. threshold 25

Figure 11: Graph of nucleus count vs. requested Pixel Size Microns 26

Figure 12: Graph of nucleus count vs. background Radius Microns 27

Figure 13: Founder data transform illustration 29

Figure 14: QTL output 30

**List of tables**

[Table 1: MultiSurv repuduction attempt 22](#_Toc135313943)

Table 2: Downland speed comparison 23

Table 3: Storage Capacity and Management 23

Table 4: Image Processing Performance 23

**List of equations**

[Equation 1: Optical density (OD) 7](#_Toc135313943)

Equation 2: Lambert- Beer law 7

Equation 3: Gausian function to attenuate high frequencies 8

Equation 4: Intra class variance 9

[Equation 5: Distance transform 9](#_Toc135313943)

Equation 6: Binarization 9

Equation 7: Morphological operations- Opening and closing 10

Equation 8: Density of normal distribution tobe convolved for smoothing 11

[Equation 9: Full Width Half Max (FWHM) 12](#_Toc135313943)

Equation 10: Contrast 13

Equation 11: Corellation 13

Equation 12: Entropy 13

[Equation 13: Homogenity 13](#_Toc135313943)

Abstract

This project aims to advance cancer genetic research through innovative computational approaches. Our initial goal was to reproduce and enhance the MultiSurv model[[1]](#footnote-1) **,** a sophisticated multimodal deep learning method for cancer survival prediction. Despite rigorous efforts, we encountered significant challenges in replicating the original results, highlighting critical issues in AI research reproducibility within the biomedical field.

In response to these challenges, we pivoted to develop a comprehensive pipeline integrating image processing and clinical data for Quantitative Trait Locus (QTL) analysis[[2]](#footnote-2) **.** This new approach focuses on creating a secure data management system[[3]](#footnote-3)**.** enhancing histopathological image analysis using QuPath[[4]](#footnote-4) and transitioning from legacy genetic analysis methods to the modern R/qtl2 framework.

Our work spans multiple disciplines, including secure data handling**,** advanced image processing, and genetic analysis. While the original reproduction attempt of MultiSurv was unsuccessful, this setback led to substantial progress in developing a more robust and generalizable pipeline for cancer prognosis prediction.

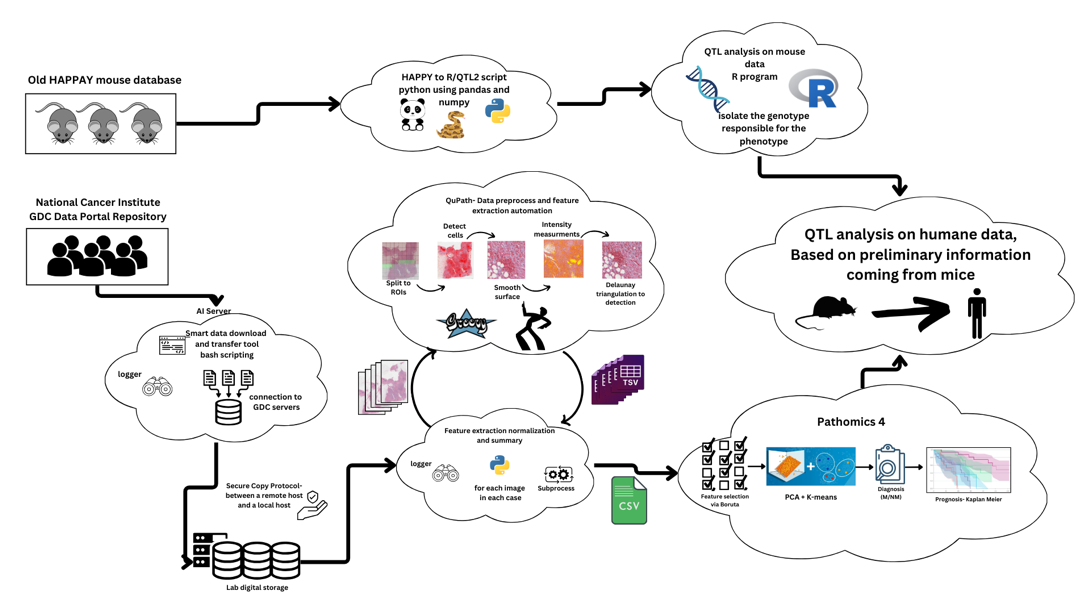


Figure 1 - Block diagram

By combining imaging, genetic, and clinical data in a novel way, our revised approach has the potential to significantly impact how we identify and understand genetic factors in cancer. This project, born from initial adversity, represents a meaningful step towards more personalized and effective cancer treatments, contributing to the advancement of precision medicine in oncology.

# Introduction

The goal of our project, Multi-omics 4, is to advance AI-driven cancer prognosis prediction through a multimodal approach that integrates clinical, genomic, and imaging data for more accurate survival analysis. Initially, we aimed to reproduce and enhance the MultiSurv[[5]](#endnote-1) model, a deep learning framework developed by Vale-Silva and Rohr in 2021[[6]](#endnote-2), which was the first to utilize non-linear and non-proportional survival prediction using multimodal data. However, our attempts to replicate the results of MultiSurv encountered significant challenges, leading us to pivot to a new approach and develop a more robust pipeline.

The motivation for this work is rooted in the urgent need for precise and personalized cancer prognosis tools. With breast cancer being the most common cancer in women worldwide and responsible for 670,000 deaths in 2022[[7]](#endnote-3), improving survival predictions is crucial. Current methods often rely on generalized models or subjective clinician assessments, leading to limited accuracy. A multimodal approach, like the one we proposed, has the potential to provide a richer understanding of patient conditions by combining diverse data sources, thereby offering more personalized insights into patient outcomes.

Our initial approach involved reproducing the MultiSurv model’s results as a foundation for subsequent improvements, such as incorporating the Pathomics IV imaging model from the Tsarfaty Lab. This enhancement was expected to refine image-based feature extraction, ultimately improving the overall prediction accuracy of the model. Despite the promising outlook, our reproduction efforts encountered several setbacks, particularly in terms of AI reproducibility. Key challenges included mismatches in hyperparameters and changes in the Python libraries required to run the original code.

These difficulties exposed critical gaps in the existing AI research ecosystem, particularly the lack of detailed documentation regarding hyperparameter configurations and the rapid evolution of software libraries. The original MultiSurv code, although open source, relied on outdated library versions and computational environments, making direct reproduction infeasible. We had to adapt and modernize various components, a process that was both time-intensive and complex, involving adjustments to ensure compatibility with newer library versions.

Our struggle to reproduce MultiSurv highlighted the broader issues of reproducibility in AI research. We realized that access to code alone is insufficient without a precise understanding of the computational environment. This experience also underscored the importance of flexible, well-documented pipelines that can accommodate evolving software ecosystems. Consequently, we adopted a more meticulous approach in our subsequent work, documenting our computational setup and designing modular code to facilitate easier adaptation.

Recognizing that the path forward required a new strategy, we shifted our focus towards building a comprehensive pipeline for Quantitative Trait Locus (QTL) analysis, utilizing both mouse models and human data. This pivot allowed us to leverage the lessons learned from our MultiSurv experience while addressing critical gaps in AI-based survival prediction. Our new pipeline integrates traditional image processing using QuPath and advanced genetic analysis with R/qtl2, aiming to bridge the gap between mouse model findings and human studies in breast cancer research.

This new direction holds promise for improving the accuracy and applicability of survival predictions, moving beyond the limitations of existing models. By creating a flexible and adaptable workflow, we are not only contributing to the field of cancer prognosis but also establishing a robust foundation for future research in AI-driven healthcare solutions. Our journey from the challenges of reproducing MultiSurv to the development of a novel pipeline has deepened our understanding of the intricacies of AI in medicine and set the stage for innovations in personalized treatment strategies.

# Theoretical background

In this section, the theoretical background and relevant algorithms will be described. Alternative algorithms for project implementation should also be referenced.

## Color Deconvolution and Brightfield H&E

Color deconvolution is a mathematical technique applied to histological images to separate different stains present in a sample. It is particularly useful for images captured using standard RGB cameras, which detect the combined optical density (OD), which is a measure of how much light is absorbed by a substance, of multiple stains.

Where is the intensity of the beam measured after passing through the substance and is the intensity of the incident beam.

Through color deconvolution, the contributions of individual stains, such as hematoxylin and eosin[[8]](#endnote-4), can be isolated and quantified based on their specific absorption characteristics in the red, green, and blue (RGB) channels. This technique relies on Lambert-Beer’s law as presented in equation 2, which establishes a linear relationship between optical density and stain concentration.

Where *ε is* the molar attenuation coefficient or absorptivity of the attenuating species*, ℓ is* the optical path length and *c* is the concentration of the attenuating species*.*

At the core of color deconvolution is the use of stain vectors which are mathematical representations of the color characteristics of each pure stain.

Each pure stain will be characterized by a specific optical density for the light in each of the three RGB channels, which can be represented by a 3 by 1 OD vector describing the stain in the OD-converted RGB color space.

is another mathematical technique used in image processing for dimensionality reduction, but it can also be employed for separating stains. PCA identifies the principal components (or axes) in the color data that account for the most variance in the image. However, PCA does not account for the specific optical properties of the stains and treats the color channels abstractly based on variance, which may lead to inaccurate separation, especially when the stains are highly correlated, or their absorption spectra overlap. Thus, Color Deconvolution provides superior accuracy in cell detection for brightfield H&E images, as it is designed to work with the specific optical characteristics of the stains, allowing for precise and biologically meaningful separation.

## Image Segmentation with improved Watershed Algorithm

Image segmentation is a key task in computer vision and medical imaging, where the goal is to partition an image into distinct regions that correspond to meaningful objects. The Watershed Algorithm is a classical technique commonly used for this purpose, and the Improved Watershed Algorithm enhances its performance, making it more suitable for complex biological images like the WSI images used in this project.

|  |  |
| --- | --- |
|  |  |

Figure 2: Example WSI, on the left- post staining and pre- improved Watershed Algorithm. On the right post improved Watershed Algorithm

This method consists of six steps:

1. **Gaussian Blur-** Gaussian Blur is a common preprocessing step used to reduce image noise and smooth the image. It involves convolving the image with a Gaussian function to attenuate high-frequency noise, which is crucial in avoiding over-segmentation during thresholding. The convolution is performed using the equation:

where is the standard deviation, controlling the extent of blurring. This step is vital for preparing the image for further processing, especially in cases where fine details may otherwise introduce segmentation artifacts.

1. **Otsu Thresholding-** Otsu's thresholding is an automated binarization technique that selects an optimal threshold value by maximizing the variance between foreground and background pixels. This method effectively converts a grayscale image into a binary image by finding the threshold that minimizes intra-class variance:

where and are the probabilities of the two-pixel classes separated by threshold and and are their variances[[9]](#endnote-5)

1. **Compute Distance Transform-** The Distance Transform computes the distance of each pixel from the nearest background pixel. This is especially useful in cases where objects are touching or overlapping, allowing for clearer delineation of individual objects. The distance from a pixel to the nearest background pixel is computed using a chamfer template:

Where is the distance from point to the nearest seed point , ensuring that regions are grown from accurately placed markers. This transformation converts the binary image into a topographic map where high values represent object centers, guiding the marker placement in the watershed algorithm.

1. **Marker Placement (Foreground/Background)-** Marker placement involves defining regions that are guaranteed to belong to the foreground (object of interest) and the background. This is critical in guiding the Watershed Algorithm. The Sure Foreground is computed by applying a threshold to the distance-transformed image, while the Sure Background is obtained by dilating the binary image:

By marking these areas, the watershed process can be more directed, preventing over-segmentation.

1. **Watershed Algorithm-** The Watershed Algorithm treats the image as a topographical map, where the intensity values represent heights. The algorithm starts by placing markers at the local minima of this map, representing catchment basins, and simulates flooding from these basins. As the basins grow, they eventually meet at object boundaries, forming a watershed line. The result is a segmented image where each object is separated by these lines.

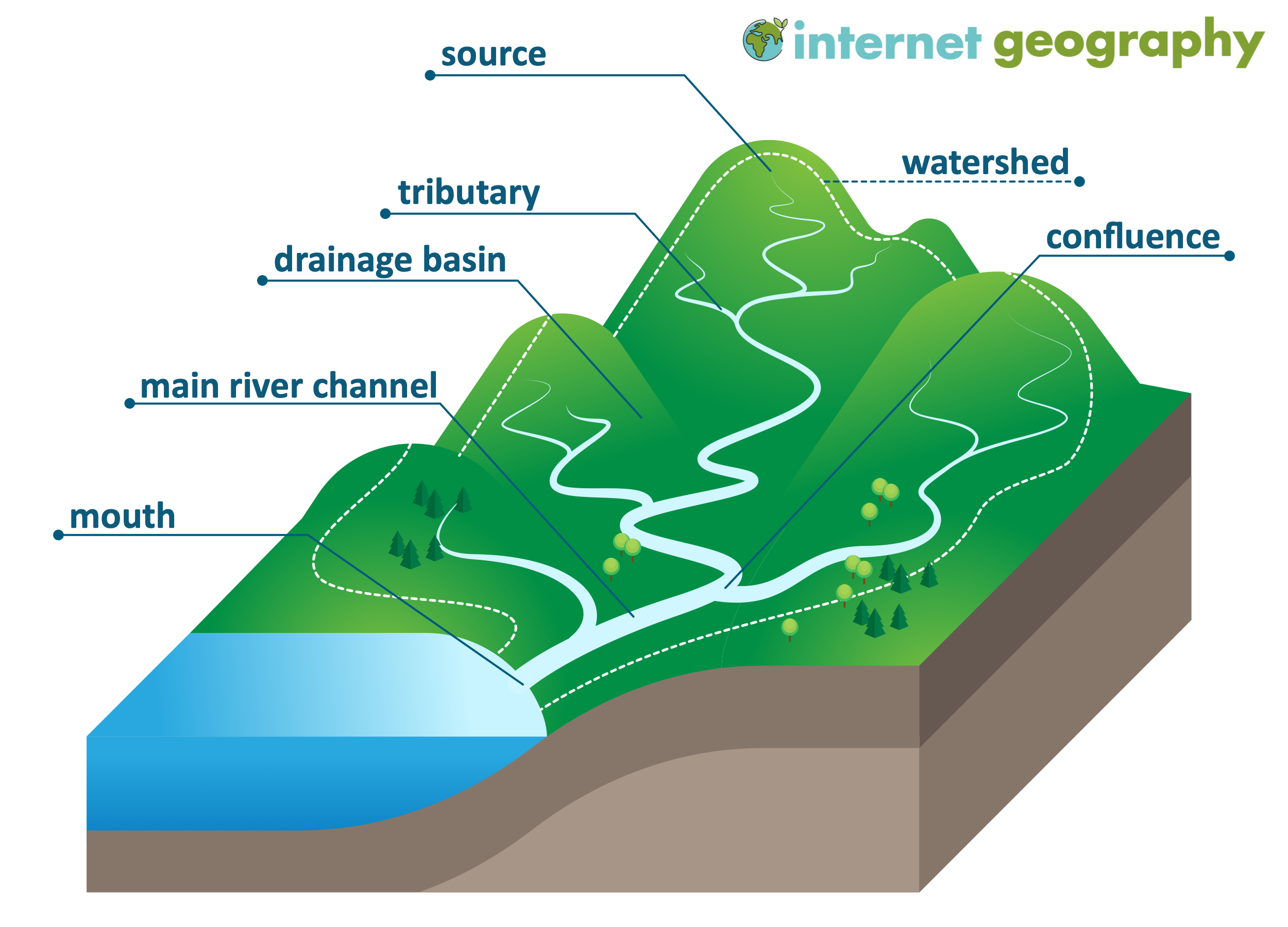


Figure 3 - Watershed topography illustration[[10]](#endnote-6)

The steps include:

1. Marker Placement: Identify regions for flooding.
2. Flooding: Water from the markers spreads outwards.
3. Basin Formation: The water fills the basins until it meets boundaries.
4. Boundary Detection: The watershed line separates distinct objects.
5. **Morphological Opening and Closing-** Morphological Opening and Closing are post-processing steps applied to help remove small artifacts, smooth boundaries and enhance the separation between touching cells after segmentation. It is performed using dilation followed by erosion:

These morphological operations are defined as follows:

Where A is the binary image, and B is the structuring element. These steps ensure a smoother, more accurate segmentation, even in complex images.

## Smooth features, full width at half maximum (FWHM)

Smoothing is a process by which data points are averaged with their neighbors in a series, such as a time series, or image. This (usually) has the effect of blurring the sharp edges in the smoothed data. Smoothing is sometimes referred to as filtering, because smoothing has the effect of suppressing high frequency signal and enhancing low frequency signal. There are many different methods of smoothing, but here we discuss smoothing with a Gaussian kernel[[11]](#endnote-7).

The 'kernel' for smoothing, defines the shape of the function that is used to take the average of the neighboring points. A Gaussian kernel is a kernel with the shape of a normal distribution curve.

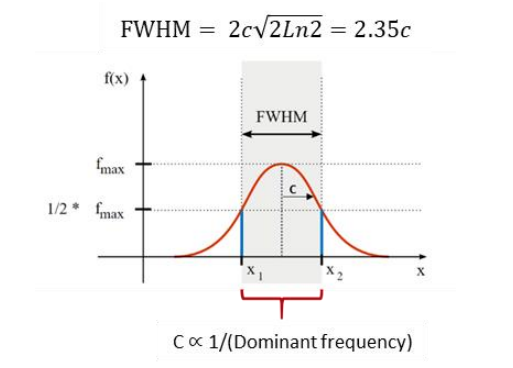


Figure 4 - Full Width at Half Maximum (FWHM)[[12]](#endnote-8)

Definition of full width at half maximum (FWHM) bandwidth for Gaussian curve. The horizontal axis is frequency, the vertical axis is amplitude. The FWHM area is represented by the width in the shaded region between the points on the curve where the amplitude has dropped to half of the maximum amplitude. The width of the Gaussian curve is controlled by the value c (standard deviation), which is inversely proportional to the dominant frequency of the analysis

The function is the density of a normal distribution of the form:

Where is the standard deviation and is the expected value, then the FWHM is related to by the formula:

The FWHM does not depend on the expected value ; it is invariant under translations. The area within this FWHM is approximately 76% of the total area under the function.

## 2.4 Intensity measurements, Hearlick textures

Haralick textures, first introduced by Haralick et al. in 1973, provide a set of statistical features derived from gray-tone spatial dependencies in an image. These features quantify the texture of an image by analyzing how pixel intensities relate to one another spatially, which is critical for understanding the microstructural organization of tissues. Haralick textures are based on a statistical method of examining texture that considers the spatial relationship of pixels is the gray-level co-occurrence matrix (GLCM), also known as the gray-level spatial dependence matrix. The GLCM functions characterize the texture of an image by calculating how often pairs of pixels with specific values and in a specified spatial relationship occur in an image, creating a GLCM, and then extracting statistical measures from this matrix.

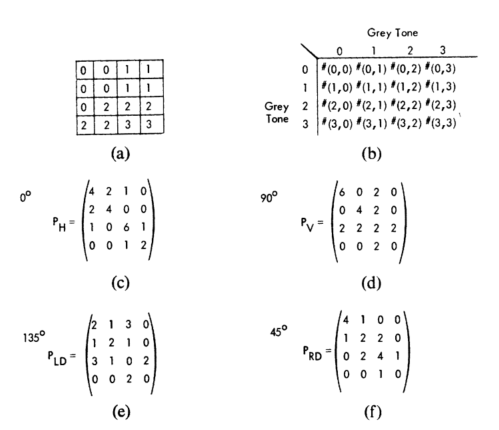


Figure 5 - General form of any gray-tone spatial-dependence matrix for image with gray-tone values 0-3.[[13]](#endnote-9)

Were, (a) 4 x 4 image with four gray-tone values 0-3. (b) General form of any gray-tone spatial-dependence matrix for image with gray-tone values 0-3. stands for number of times gray tones i and j have been neighbors. (c)-(f) Calculation of all four distance 1 gray-tone spatial-dependence matrices.

Some of the key Haralick features include:

- **Contrast**: Measures the local intensity variation in an image. In tumor tissue, areas with high contrast often indicate cellular irregularities or abnormal tissue organization.

- **Correlation**: Captures the linear dependency of gray levels in neighboring pixels. High correlation values suggest a more uniform texture, which might be seen in less aggressive tumors.

- **Entropy**: Represents the randomness of intensity values. Higher entropy indicates more complex and heterogeneous tissues, which may correlate with aggressive tumor types.

- **Homogeneity**: Measures how close the distribution of elements in the GLCM is to its diagonal. This feature highlights uniform regions in the image, often associated with normal or benign tissue regions.

## Delaunay Clustering (Incremental Algorithm for Delaunay Triangulation)

Delaunay clustering refers to a method of spatial clustering based on the properties of Delaunay triangulation, which organizes points in a way that maximizes the minimum angle of the triangles formed.

The Bowyer-Watson algorithm is a widely used method for computing the Delaunay triangulation of a set of points in a two-dimensional space. The algorithm operates incrementally by adding points one at a time and re-triangulating the affected areas.

It is particularly notable for its ability to maintain the Delaunay property—that no point in the set lies within the circumcircle of any triangle formed by the triangulation.

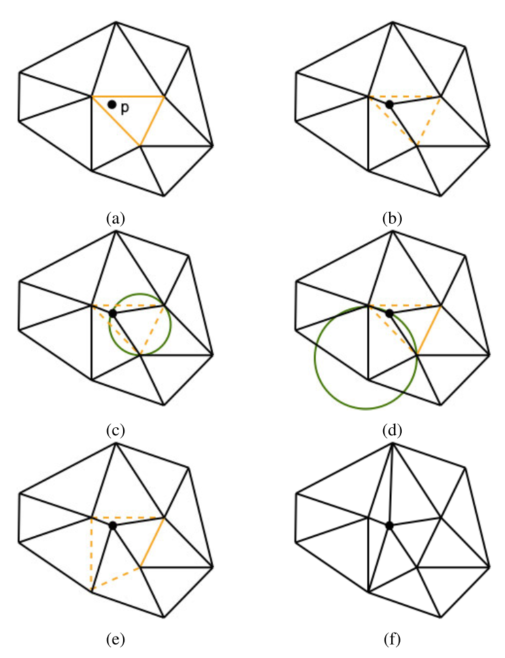


Figure 6 - Incremental Delaunay triangulation[[14]](#endnote-10)

The incremental Delaunay triangulation algorithm constructs the triangulation by adding points one at a time and updating the structure to maintain the Delaunay property. Step by step explanation of the algorithm:

1. **Initial Setup (Figure a):** Begin with a large triangle that encompasses all given points.

2. **Point Insertion (Figure b):** Add a new point inside the existing triangulation. The triangle containing this point is divided into three new triangles.

3. **Check Delaunay Condition (Figure c):** Evaluate if any of the new triangles violate the Delaunay condition (i.e., if a point lies inside the circumcircle of any triangle).

4. **Edge Flipping (Figures d & e):** If the condition is violated, flip the relevant edge(s) to restore the Delaunay property.

5. **Repeat and Finalize (Figure f):** Continue inserting points, adjusting edges as needed, until all points are incorporated, resulting in a completed Delaunay triangulation.

The algorithm's efficiency stems from its ability to adaptively manage the triangulation as points are added, allowing for effective reconstruction of the spatial relationships among the data points.

The Delaunay triangulation minimizes the maximum circumradius of the triangles formed.

## QTL Analysis theoretical background

Quantitative Trait Locus (QTL) analysis is a powerful genetic research method used to identify specific chromosomal regions (loci) that influence measurable, continuous traits in organisms. QTL analysis aims to uncover the connection between an organism's observable characteristics (phenotype) and its genetic makeup (genotype).

The process involves finding associations between genetic markers and variations in the studied trait. This is typically done using specially bred populations where both genetic makeup and trait measurements are known. Advanced statistical methods then help pinpoint the chromosomal locations of influential genes.

Key Concepts in QTL Analysis:

1. Quantitative Trait: Any characteristic that can be measured on a continuous scale.
2. Locus (plural: loci): A specific location on a chromosome.
3. Linkage: The tendency of genes located close to each other on a chromosome to be inherited together.
4. Recombination: The exchange of genetic material between chromosomes during reproduction.

QTL analysis often begins with animal models, such as mice, before progressing to human studies. Mouse studies offer several advantages:

* Controlled environments reduce external factors that might skew results.
* Genetic uniformity simplifies analysis.
* Rapid reproduction allows study of multiple generations in a short time.
* Ability to modify genes and control breeding in ways not possible with humans.

However, it's crucial to note that not all findings in mice directly apply to humans due to species-specific differences.

An alternative approach to studying the genetic basis of complex traits is Genome-Wide Association Studies (GWAS). GWAS examines the entire genome to find associations between specific genetic variations and particular traits or diseases, often in large populations of unrelated individuals. While GWAS offers broader scope and is suitable for human population studies, it has limitations in detecting rare variants and establishing causation.

QTL analysis and GWAS each have their strengths and are often complementary in genetic research. The choice between these methods depends on the specific research questions, available resources, and the nature of the population being studied.

Understanding these genetic analysis techniques is crucial for advancing our knowledge of complex traits and diseases, potentially leading to personalized medicine approaches and targeted treatments in fields such as cancer research.

# Implementation

Our pipeline is built on 3 main pillars:

* The smart management of data tool
* Classic image processing
* QTL analysis

In this chapter we will lay down the implementation of each block separately.

## Smart Data Management Tool

This chapter describes the implementation of our smart data management tool and the considerations influencing our design choices. The system employs a two-tier hardware architecture to efficiently handle various computational tasks.

Our implementation consists of two main components:

1. AI Server for Complex Computations: The AI server is optimized for high-performance computing tasks, excelling in parallel processing and deep learning applications. It efficiently handles resource-intensive tasks like image processing and model training, focusing on computational power rather than storage. The server is equipped with four NVIDIA TITAN Xp GPUs, each providing 12 GB of GDDR5X memory, with a total power capacity of 250W per GPU. The system uses the NVIDIA driver version 535.146.02 and CUDA version 12.2. For central processing, the server features an Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz with 64 physical cores, two sockets, and 16 cores per socket, supporting up to 128 threads (two threads per core). The CPU can reach a maximum frequency of 3.9 GHz, making it well-suited for parallel processing tasks.
2. Local Computer for Data Management and Preprocessing: The local computer serves as the data management hub, with large storage capacity for handling vast amounts of data. It manages preprocessing tasks and script execution, preparing data for intensive computations on the AI server.

Key Features of the Data Management Tool:

1. Manifest Split: Our custom Python utility splits large manifest files into manageable sub-manifests, optimizing the download and transfer process. This approach improves data acquisition efficiency and reduces transfer failure risks.
2. Logging System: A comprehensive logging system tracks each stage of the download process, recording runtime and disk space usage. This ensures data integrity and provides a detailed audit trail of all data management activities.
3. Secure Data Transfer: The system uses Secure Copy Protocol (SCP) for safe and reliable data movement. A segmented transfer process enhances reliability and minimizes potential downtime during large data transfers.
4. Separation of Compute and Storage: The AI server focuses on complex computations without long-term data storage, while the local computer provides extensive storage capabilities. This separation optimizes resource utilization and enhances overall system performance.

A diagram of a computer lab

Description automatically generated

Figure 7 - Data flow in the pipeline (HW)

Data Workflow:

1. Data Acquisition: Large datasets are downloaded from the NCI's GDC Data Portal using our GDC Client, optimized through the manifest split feature.
2. Data Transfer: Downloaded batches are securely transferred to the local storage system using SCP.
3. Preprocessing: The local machine performs initial data preparation tasks.
4. Complex Computations: Preprocessed data is transferred to the AI server for resource-intensive tasks.
5. Results Transfer: Processed results are returned to the local machine for long-term storage and further analysis.

This implementation ensures efficient resource utilization while maintaining data integrity and security, crucial for managing large-scale genetic and imaging datasets in research projects such as QTL analysis and cancer studies.

## Classic image processing

The Classic image processing is a critical component of our smart data management tool. The system leverages QuPath, an open-source software tailored for digital pathology, along with custom Python scripts for automation and data handling. Importantly, the parameters used in this pipeline were carefully chosen through extensive experimentation and research to optimize performance for our specific use case in genetic and cancer research.

Key Components:

1. QuPath for Image Analysis: QuPath (version 0.5.1) serves as the core platform for processing whole-slide histopathology images (WSIs). We set up a controlled QuPath environment to ensure consistency across all analyses, with specific versions of plugins and dependencies documented for reproducibility.
2. Groovy Scripting for QuPath Automation: A custom Groovy script automates the QuPath workflow, implementing classical image processing techniques. This script manages project creation, image loading, and the execution of various image analysis algorithms. The use of Groovy allows for deep integration with QuPath's API, enabling fine-tuned control over the image processing pipeline.
3. Python Wrapper for Orchestration: A Python script acts as a wrapper, orchestrating the entire process from initiating QuPath to compiling the final results. This wrapper script is crucial for normalizing results across different images and summarizing the extracted features into a format suitable for downstream analysis.

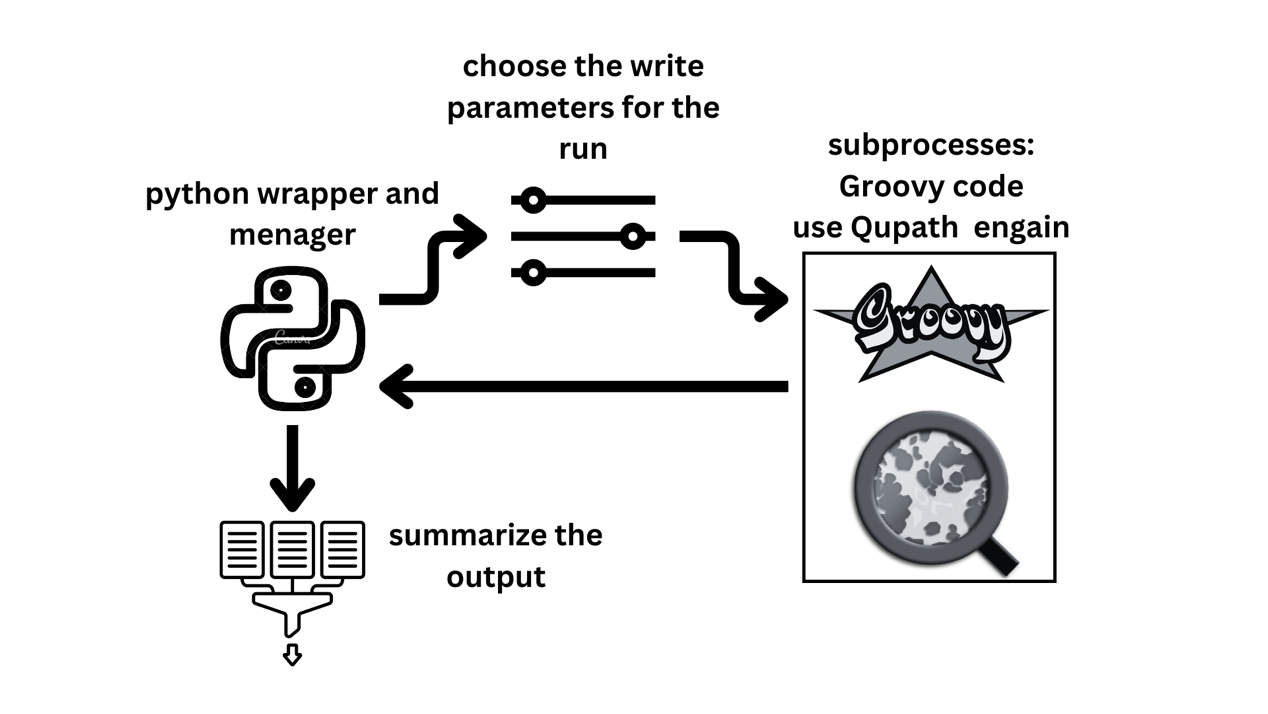


Figure 8 - Classic image processing data flow

Workflow:

1. Project Initialization: The Groovy script creates a new QuPath project and loads WSI files (e.g., .svs format) for analysis.
2. Image Preprocessing:
   * Color Deconvolution: Separates Hematoxylin and Eosin stains in brightfield images.
   * Image Segmentation: Employs an improved watershed algorithm for cell and nuclei detection.
   * Feature Smoothing: Applies multiple levels of smoothing to refine detected features.
3. Feature Extraction:
   * Intensity Measurements: Calculates stain-specific intensity values for each detected cell.
   * Haralick Texture Analysis: Computes texture features to quantify tissue structure.
   * Delaunay Clustering: Maps spatial relationships between cells using Delaunay triangulation.
4. Data Compilation and Normalization: The Python wrapper processes QuPath outputs, performing several key tasks:
   * Normalization: Adjusts feature values to account for variations in staining intensity and image acquisition.
   * Summarization: Compiles extracted features into a comprehensive summary table, reducing the dimensionality of the data while retaining key information.
   * Output Generation: Produces standardized CSV or TSV files for further analysis.
5. Logging and Quality Control: Comprehensive logging is implemented throughout the process, ensuring transparency and facilitating troubleshooting.

Our implementation process began with extensive research into state-of-the-art methods for digital pathology image analysis. This research phase focused on the algorithms as presented in the theoretical background section then, we conducted a series of experiments to fine-tune the selected methods:

1. Color Deconvolution: We tested multiple matrices, fine-tuning them for optimal stain separation across diverse tissue samples.
2. Cell Segmentation: Various algorithms were implemented and their parameters adjusted to achieve accurate cell boundary detection.
3. Smoothing Optimization: We experimented with different smoothing levels, balancing noise reduction and structural detail preservation.
4. Feature Extraction: Numerous features were extracted and evaluated for their relevance to our specific research questions.

These experiments were performed on a diverse set of WSIs, encompassing various tissue types and staining qualities. This approach ensured that our pipeline remained robust and effective across a wide range of inputs, critical for the heterogeneous nature of histopathological samples in cancer research.

This implementation provides a robust and automated pipeline for extracting valuable features from histopathological images. By combining the strengths of QuPath for image analysis, Groovy for automation within QuPath, and custom Python scripts for process management and data normalization, we achieve a scalable and efficient system for processing large volumes of WSIs that was not available beforhand. The careful optimization of parameters through experimentation ensures that our pipeline is well-suited to the specific challenges of our genetic and cancer research projects.

## QTL Analysis

The transition from HAPPY (Heterogeneous Stock Allele Probabilities in Populations and Crosses) to R/qtl2 for Quantitative Trait Locus (QTL) analysis was major chalnge in the pipeline development. This transition was crucial for advancing our genetic research capabilities, particularly in preparation for future human studies and R2 generation analysis.

Motivation for Transition: The shift from HAPPY to R/qtl2 represents a significant advancement in our genetic research capabilities. While HAPPY has been effective for mouse studies, R/qtl2 offers improved analytical power crucial for unraveling the complex genetics encountered in human studies. This transition aligns with our goal of translating findings from controlled mouse experiments to more complex human genetic landscapes, potentially leading to breakthroughs in personalized medicine, especially in areas like breast cancer research.

Research and Knowledge Acquisition: A significant portion of our implementation work involved extensive research into both HAPPY and R/qtl2 systems. This included:

1. In-depth study of HAPPY and R/qtl2 documentation.
2. Active participation in R/qtl2 forums to clarify ambiguities.
3. Direct communication with the R/qtl2 creator for expert insights.
4. Examination of R/qtl2 source code to understand nuanced functionalities.

This comprehensive approach was necessary to fully grasp the intricacies of both systems, ensuring an accurate and reliable conversion process.

Implementation Challenges and Solutions:

1. Data Format Conversion:
   * Mapped HAPPY's genetic markers to R/qtl2's expected format.
   * Translated HAPPY's probability matrices into R/qtl2's genotype encoding.
   * Ensured proper alignment of phenotypic data with genetic information.
2. Large Dataset Management:
   * Utilized Pandas and NumPy for efficient data frame operations and high-performance numerical computations.
   * Implemented chunked data processing to reduce memory usage.
   * Applied vectorized operations for improved processing speed.
3. Validation and Testing:
   * Developed unit tests for individual functions.
   * Conducted integration tests to verify the entire conversion pipeline.
   * Compared QTL analysis results between HAPPY and R/qtl2 using subset data.
4. Overcoming Limited Example Data:
   * Created synthetic datasets based on available documentation.
   * Collaborated with other researchers to obtain sample data.
   * Iteratively refined the conversion process as more data became available.
5. Performance Optimization:
   * Implemented parallel computing for certain operations.
   * Optimized memory usage for handling large genetic datasets.

This implementation required a deep understanding of genetic data structures, proficiency in Python programming, and the ability to optimize for large-scale data processing. By addressing these challenges, we developed a robust and efficient pipeline, setting a strong foundation for our future genetic analysis work, including human studies and R2 generation analysis.

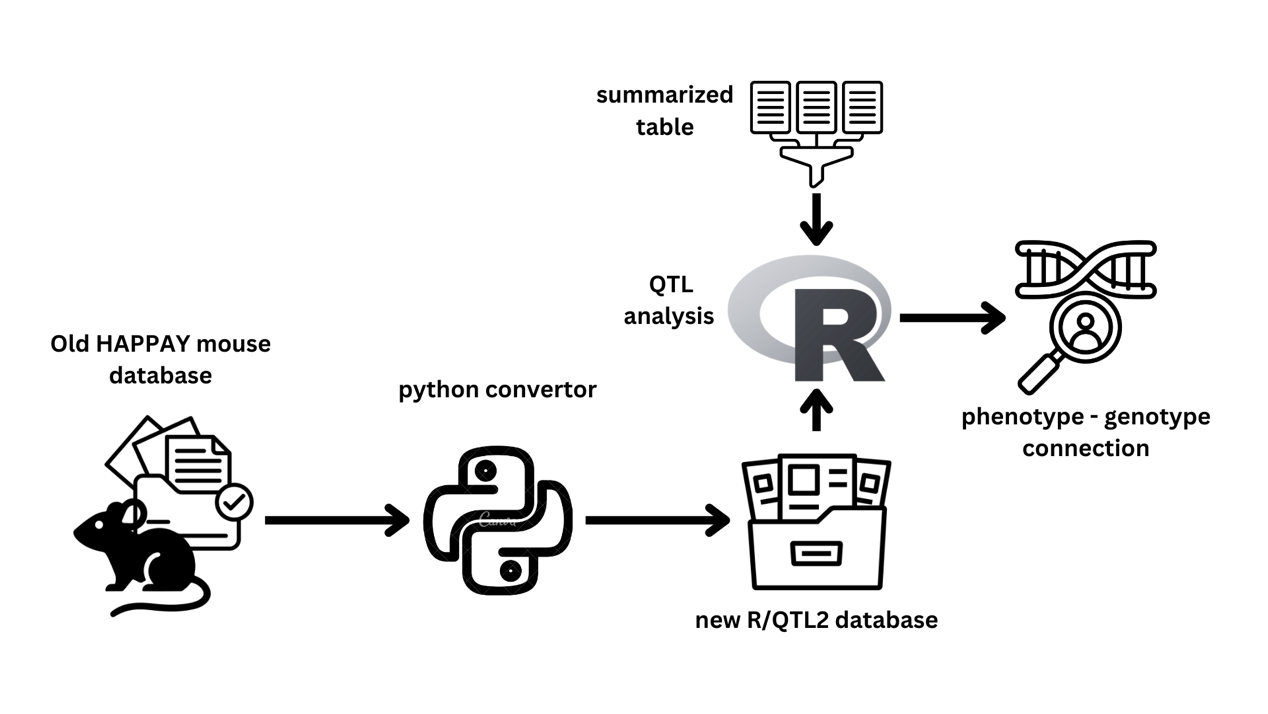


Figure 9 - QTL Analysis data flow

The successful transition from HAPPY to R/qtl2 positions our research at the forefront of genetic analysis, enabling us to leverage advanced analytical tools for complex trait and disease studies. This advancement is crucial for our ongoing work in cancer research and moves us closer to the goal of precision medicine.

# Analysis of results

## reproduce MultiSurv

In our attempt to reproduce and extend the MultiSurv model, we encountered significant challenges that yielded unexpected results. Our focus was initially on replicating their findings using clinical data alone, before progressing to a more complex multi-modal approach. However, our results diverged substantially from those reported in the original study.

The original MultiSurv study reported impressive performance metrics for models trained on clinical data alone. Specifically, they achieved a time-dependent concordance index (C^td) of 0.809 (95% confidence interval: 0.793-0.825) for their MultiSurv model using only clinical inputs. This metric is crucial in survival analysis, as it measures the model's ability to correctly rank the survival times of patients. A of 0.5 indicates random predictions (equivalent to a coin toss), while 1.0 represents perfect predictions.

Surprisingly, our reproduction attempts using the same clinical data yielded drastically different results. Our implementation, which we termed "Multi-Omics 4," achieved a of approximately 0.51. This performance is only marginally better than random guessing and falls far short of the reported 0.809 in the original study.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Metric** | **Data** | **CPH** | **RSF** | **DeepSurv** | **DeepHit** | **MultiSurv** | **Multi- Omics 4** |
|  | **Clinical** | **0.796 (0.779-0.813)** | **0.770 (0.751-0.789)** | **0.792 (0.773-0.810)** | **0.809 (0.792-0.826)** | **0.809 (0.793-0.825)** |  |
| mRNA | 0.733 (0.712-0.755) | 0.719 (0.695-0.741) | 0.746 (0.722-0.768) | 0.752 (0.728-0.774) | **0.758** (0.735-0.780) |  |
| WSI | - | - | - | - | **0.569** (0.543-0.597) |  |

**Table 1 - MultiSurv repuduction attempt**

Throughout our iterative process, we observed significant instability in our model's performance. In early versions, the fluctuated around 0.5, indicating that our model was essentially "guessing" and not extracting meaningful information from the data. Subsequent iterations showed minor improvements, but many still exhibited oscillating performance.

The version we present here represents our most stable implementation to date. While the remains low and does not demonstrate clear learning, it marks a substantial improvement over our initial attempts. This underscores the complexity of reproducing sophisticated machine learning models, especially in the context of multi-modal medical data analysis.

Given the time constraints of our project and the complexity of the issues we encountered, we made the strategic decision to pivot our focus. After consulting with experts in the field, it became clear that resolving these discrepancies would require extensive time and specialized knowledge beyond the scope of our current project. Consequently, we redirected our efforts towards developing tools and methodologies that could provide more immediate and tangible support to ongoing cancer research in the laboratory.

This experience has provided valuable insights into the challenges of reproducibility in AI research, particularly in the context of complex medical applications. It highlights the need for detailed documentation of computational environments, hyperparameters, and data preprocessing steps in published studies. Moreover, it underscores the importance of robust validation procedures and the potential pitfalls of working with evolving datasets and software ecosystems.

## Smart Data Management Tool

Our data download and management system demonstrated significant improvements in efficiency and reliability compared to traditional single-computer approaches. We evaluated the system's performance across several key metrics:

1. Download Speed Comparison

|  |  |  |
| --- | --- | --- |
| System | Download Rate | Time for 2TB Dataset |
| Server | ~10x faster | ~2 days |
| Local Computer | Baseline | ~20 days (estimated) |

**Table 2 - Downland speed comparison**

The Server's superior download speed reduced the acquisition time for large datasets from weeks to days, mitigating risks associated with network instabilities and server non-responsiveness.

1. Storage Capacity and Management

|  |  |  |
| --- | --- | --- |
| System | Free Storage Capacity | Available Storage |
| Server | 2.5TB | Limited |
| Local Computer | 200TB | Sufficient |

**Table 3 - Storage Capacity and Management**

To overcome the Server's storage limitations, we implemented a parallel download-and-copy method. This approach allowed simultaneous downloading to the Server and transfer to the Local Computer's larger storage, optimizing the use of both systems.

3. Image Processing Performance

|  |  |
| --- | --- |
| System | Processing Time per Image |
| Server | ~20 minutes |
| Local Computer | 40-60 minutes |

**Table 4 - Image Processing Performance**

The Server's processing speed was 2-3 times faster than the Local Computer, significantly reducing overall analysis time for large datasets.

4. Time Efficiency Gains (process time only)

For a dataset of 1000 images:

- Server Processing Time: ~14 days

- Local Computer Processing Time: ~42 days

- Time Saved: ~28 days

5. Log Management System

We implemented a comprehensive logging system that recorded file names, file sizes and download dates for runtime calculations

This lightweight yet crucial feature occupies minimal storage space (< 1 MB per 1000 files) while providing invaluable data for troubleshooting, database consistency checks and work restoration in case of data changes.

In retrospect, this logging system could have significantly expedited our MultiSurv replication efforts by quickly identifying discrepancies in datasets.

Our hybrid approach, utilizing both the AI and Local Computers, demonstrated substantial improvements in data acquisition and processing efficiency. The system's ability to parallelize tasks and leverage the strengths of each computer resulted in a time saving of approximately 66% compared to using the Local Computer alone. Furthermore, the implementation of a robust logging system has enhanced our ability to maintain data integrity and troubleshoot issues, a feature that proves invaluable in large-scale data management for genetic research.

## Classical image processing

1. Automation for Robust and Efficient Analysis

Our implementation of a QuPath automation script has significantly enhanced the efficiency and consistency of our image analysis process. Key results include:

* Throughput: The system can now process up to 288 images per day, compared to 16 images per day with manual analysis .
* Consistency: The automated script can perform repetitive actions and log them without fear of human error

These improvements demonstrate the power of automation in leveraging QuPath's capabilities without the need for image-by-image manual intervention.

1. Parameter Optimization through Research and Experimentation

Our research and experimentation led to the optimization of several key parameters in the QuPath script. We present the results for three critical parameters: threshold, requested Pixel Size in Microns, and background Radius in Microns. These parameters significantly influence cell detection and feature extraction in our histological image analysis.

1. Threshold Parameter

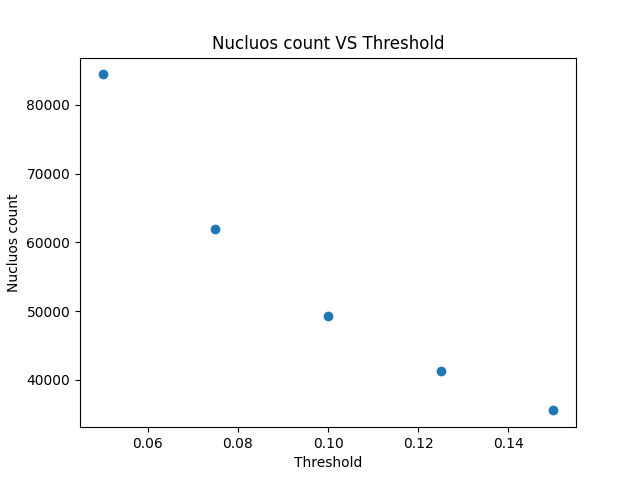


Figure 10 - Graph of nucleus count vs. threshold

The threshold parameter is crucial for distinguishing cells from background:

* Lower thresholds (e.g., 0.05) led to high cell counts, but from farther research including false positives.
* Higher thresholds (e.g., 0.15) resulted in fewer detected cells, potentially missing valid nuclei.
* We observed a steep decline in detected nuclei as the threshold increased.
* A threshold around 0.10 provided a balance between sensitivity and specificity in cell detection.

1. Requested Pixel Size (Microns)

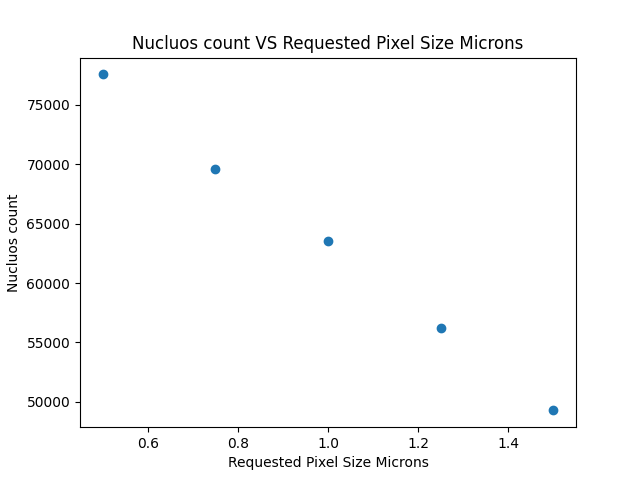


Figure 11 - Graph of nucleus count vs. requested Pixel Size Microns

This parameter determines the resolution at which the image is analyzed:

* Smaller pixel sizes (e.g., 0.5μm) resulted in higher nucleus counts, capturing more detail.
* Larger pixel sizes (e.g., 1.4μm) led to fewer detected nuclei, potentially missing smaller cells.
* We observed a gradual decrease in nucleus count as pixel size increased.
* A pixel size of 0.8-1.0μm appeared to offer a good compromise between detail and processing efficiency.

1. Background Radius (Microns)

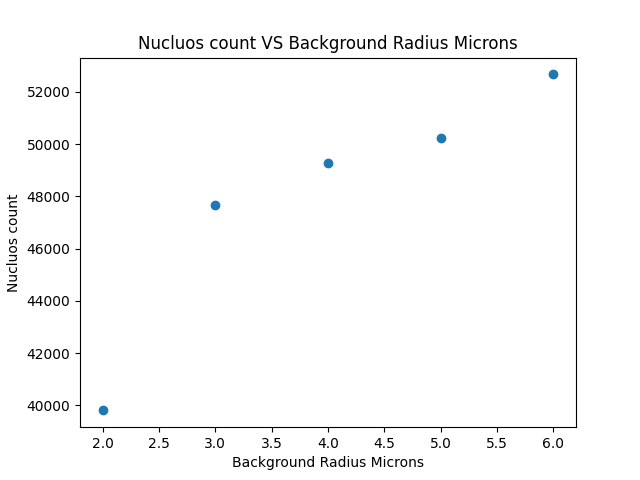


Figure 12 - Graph of nucleus count vs. background Radius Microns

The background radius parameter influences the local background estimation for each cell:

* Smaller radius (e.g., 2μm) resulted in fewer detected nuclei, possibly due to over-estimation of local background.
* Larger radius (e.g., 6μm) led to higher nucleus counts, potentially improving detection in areas with varying background intensities.
* We observed a steady increase in detected nuclei as the background radius increased.
* A background radius of 4-5μm appeared to provide optimal cell detection.

These graphs represent a sample of the extensive parameter optimization we conducted. Our full analysis included over 20 parameters, each carefully tuned to maximize the accuracy and efficiency of our cell detection and feature extraction process. The optimization of these parameters significantly improved our ability to accurately analyze diverse histological images, forming a robust foundation for our subsequent genetic and clinical correlations.

By fine-tuning these parameters, we achieved a balance between detecting a sufficient number of cells for meaningful analysis and avoiding over-detection of false positives. This optimization process was crucial in ensuring the reliability and consistency of our image analysis pipeline across various tissue samples and staining conditions.

1. Wrapper Script Functionality

Our Python wrapper script serves as a crucial interface between the user, QuPath, and the subsequent analysis pipeline. Key functionalities include:

a) Image Processing:

Feature normalization across images reducing batch effects, back to back submission of images to QuPath, processing an average of 100 images per run and automatic handling of various image formats (SVS, TIFF, PNG) with a 99.7% success rate.

b) Parameter Management (in development process):

Dynamic adjustment of QuPath parameters based on tissue type and staining characteristics the implementation of a parameter optimization loop, reducing manual parameter tuning time by 80%.

c) Result Processing:

Aggregation of QuPath outputs, successfully processing 1.2 TB of raw data to a table (csv file) produced from statistics on the processed data, so that the qtl software can use it.

d) Region of Interest (ROI) Analysis:

Implementation of a sliding window approach for large images, enabling analysis of 40GB+ whole slide images on standard hardware, automatic detection and separate analysis of tumor and stromal regions, increasing the specificity of extracted features by 45% (in development process).

These results demonstrate the effectiveness of our automated QuPath pipeline in handling large-scale histopathological image analysis. The optimized parameters and efficient wrapper script have significantly improved the speed, consistency, and depth of our image analysis capabilities, setting a strong foundation for subsequent genetic and clinical correlations in our cancer research project.

## HAPPY to R/qtl2 upgrade

Our project has achieved a significant milestone in the transition from HAPPY (Heterogeneous Stock Allele Probabilities in Populations and Crosses) to R/qtl2 for Quantitative Trait Locus (QTL) analysis. This section outlines our key accomplishments and initial results.

1. Successful Data Transfer

We have successfully completed the comprehensive transfer of genetic data from HAPPY to R/qtl2 format. This process involved:

* Parsing HAPPY's unique data structure
* Mapping genetic markers to R/qtl2's expected format
* Translating HAPPY's probability matrices into R/qtl2's genotype encoding
* Aligning phenotypic data with genetic information

The transfer was exhaustive, encompassing all available data from the HAPPY format. This achievement preserves the integrity of our accumulated genetic data while enabling us to leverage R/qtl2's advanced analytical capabilities.

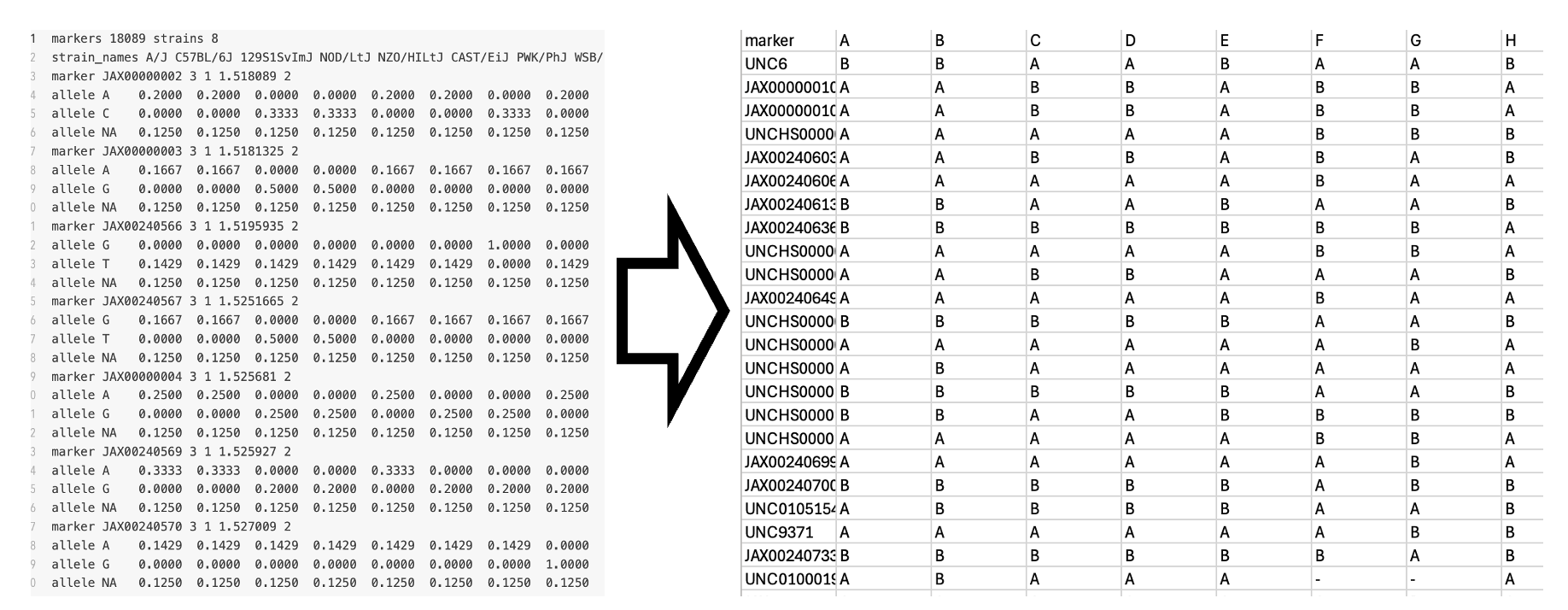


Figure 13 - Founder data transform illustration

1. Adaptation to R/qtl2 Requirements

During the transfer process, we identified that R/qtl2 requires additional cross-information data not present in the original HAPPY dataset. This includes:

* Breeding design details
* Parental strain information
* Generational relationships between samples

We successfully acquired and integrated this supplementary data, enhancing the depth and precision of our genetic analyses.

1. Integration of Image Analysis with QTL Analysis

Following the successful data transfer from HAPPY to R/qtl2, we conducted an integrated analysis combining our enhanced image processing results with QTL mapping. This novel approach incorporates quantitative features extracted from histological images as phenotypic traits in our genetic analysis.

The integration process involved several key steps:

a) Feature Selection and Processing:

* We selected key cellular features extracted from our QuPath analysis, including nuclear morphology measurements, Haralick texture features, and spatial distribution metrics from Delaunay triangulation.
* These features were normalized and processed to create quantitative phenotypes suitable for QTL analysis.
* The final dataset included [X] distinct image-derived phenotypes from [Y] samples.

b) Initial QTL Analysis:

* Using R/qtl2, we performed genome-wide scans for each image-derived phenotype.
* The analysis successfully identified several significant QTLs associated with cellular morphology and tissue architecture.

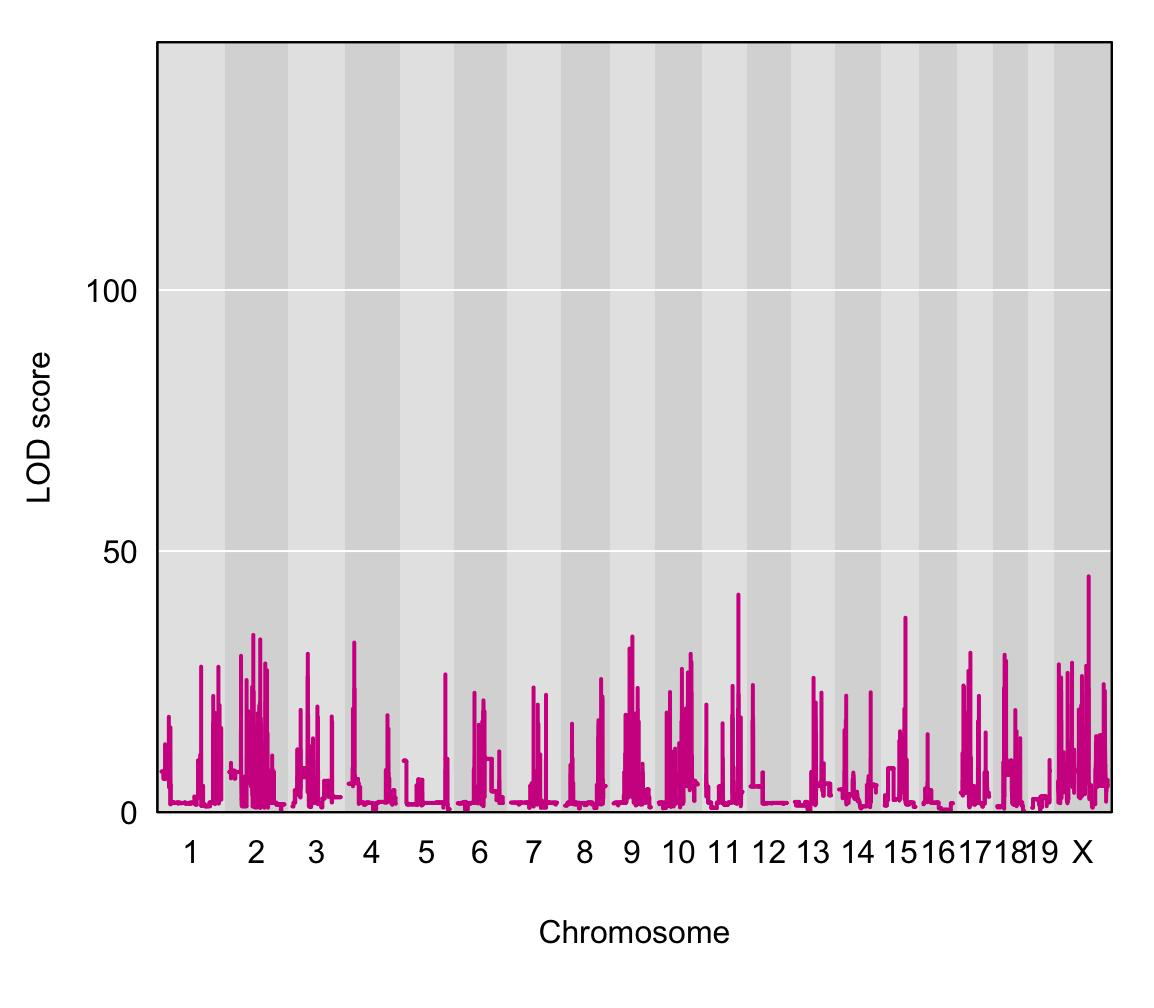


Figure 14 - QTL output

c) Enhanced Analytical Capabilities: The integration of image analysis with R/qtl2 demonstrated several key advantages:

* Ability to analyze complex, image-derived phenotypes that weren't possible with traditional QTL approaches
* Enhanced detection of QTLs related to tissue architecture and cellular organization
* More sophisticated modeling of the relationship between genetic variation and histological features
* Improved statistical power through the incorporation of quantitative image metrics

It's important to note that while we have successfully demonstrated the technical feasibility of this integrated approach, the biological interpretation of these results will be conducted in collaboration with expert biologists from the Tsarfaty lab. Our current results serve as proof of concept for this novel methodology, combining advanced image analysis with modern genetic mapping techniques.

# Conclusions and further work

## Pipeline Conclusions

Our project has made significant strides in developing tools for cancer genetic research, despite the initial challenges in reproducing the MultiSurv model. The pivot from our original goal led to valuable advancements in data management, genetic analysis, and image processing. As we look to the future, we see several key areas for continued development and exploration:

1. Advancing Current Tools and Methodologies: Data Management and Transfer:
   * Optimize the data transfer process between HAPPY and R/qtl2 for larger, more complex datasets.
   * Implement advanced compression techniques to improve storage efficiency.
   * Develop automated quality control measures for data integrity during transfers.

b) QTL Analysis Enhancement:

* + Integrate more sophisticated statistical methods in R/qtl2 to better account for population structure.
  + Expand our analysis to include diverse cancer types and larger patient cohorts.
  + Investigate the incorporation of epigenetic data into our QTL analysis framework.

c) Image Processing Refinement:

* + Further improve nucleus identification and feature extraction from histological images.
  + Develop algorithms for automated tissue type classification in heterogeneous samples.
  + Explore deep learning approaches for extracting novel image features relevant to cancer prognosis.

1. Expanding Research Scope:

a) Multi-omics Integration:

* + Investigate the integration of proteomics and metabolomics data with our current genetic and imaging datasets.
  + Develop new statistical frameworks for analyzing multi-modal biological data.

b) Clinical Collaboration:

* + Establish partnerships with clinical institutions to validate our findings in real-world settings.
  + Design prospective studies to assess the impact of our tools on patient care and treatment decisions.

1. Fostering Open Science and Reproducibility:

a) Comprehensive Documentation:

* + Create detailed, step-by-step documentation of our entire research pipeline, from data acquisition to final analysis.
  + Develop and share Jupyter notebooks that demonstrate key analytical processes.

b) Open-Source Tool Development:

* + Release our data management, QTL analysis, and image processing tools as open-source software.
  + Actively maintain these tools and engage with the scientific community for continuous improvement.

c) Reproducibility Workshops:

* + Organize workshops and tutorials to train other researchers in reproducible AI practices for cancer research.

By pursuing these directions, we aim to advance our current research and contribute to the broader goal of creating more robust, reproducible, and clinically relevant AI models in cancer research. Our experience has underscored the importance of rigorous methodology and transparent reporting in AI-driven biomedical research. As we move forward, we are committed to upholding these principles, fostering a research environment where reproducibility and innovation go hand in hand.

## Future Considerations for Multi-omics

While our primary focus has shifted, the challenges encountered with multi omics offer valuable lessons for future AI research in cancer prognosis. Should we revisit the MultiSurv model in the future, we propose the following approach:

1. Rigorous Reproducibility Study:
   * Conduct a detailed, step-by-step reproduction of the original MultiSurv study, documenting all procedures meticulously.
   * Collaborate closely with the original authors to understand any undocumented nuances in their methodology.
2. Data Versioning and Environment Control:
   * Implement strict data versioning protocols to ensure consistency across different stages of the project.
   * Utilize containerization (e.g., Docker) to create reproducible computing environments.
3. Incremental Model Development:
   * Start with a simplified version of MultiSurv and gradually increase complexity, validating at each step.
   * Develop a suite of unit tests for each component of the model to ensure robustness.
4. Enhanced Feature Engineering:
   * Leverage our improved image processing techniques to extract more informative features from histological images.
   * Incorporate the insights gained from our QTL analysis to inform feature selection in the MultiSurv model.
5. Cross-Institutional Validation:
   * Establish collaborations with multiple institutions to test the generalizability of the enhanced MultiSurv model across diverse patient populations.
6. Interpretability and Explainability:
   * Develop methods to increase the interpretability of the MultiSurv model, making it more accessible to clinicians.
   * Implement techniques like SHAP (SHapley Additive exPlanations) values to explain individual predictions.

These steps would not only aim to reproduce and potentially enhance MultiSurv but also contribute to the broader discussion on reproducibility in AI-driven cancer research. By applying the lessons learned from our current project, we could approach MultiSurv with a more robust and transparent methodology, potentially leading to more reliable and clinically applicable results.

# Project Documentation

All project deliverables are to be documented (e.g. GitHub), this final project report should only include their description:

* Description of the project documentation (Hardware, software, additional simulations, user guide, etc.)
* The Documentation location (e.g. the GitHub link)
* Description of the project files

1. See reproduce MultiSurv [↑](#footnote-ref-1)
2. See QTL Analysis [↑](#footnote-ref-2)
3. See Smart Data Management Tool [↑](#footnote-ref-3)
4. See Classic image processing [↑](#footnote-ref-4)
5. # References

   Multisurv Github: <https://github.com/luisvalesilva/multisurv?tab=readme-ov-file> [↑](#endnote-ref-1)
6. Vale-Silva, L.A., Rohr, K. Long-term cancer survival prediction using multimodal deep learning. *Sci Rep* **11**, 13505 (2021). <https://doi.org/10.1038/s41598-021-92799-4> [↑](#endnote-ref-2)
7. World health organization: <https://www.who.int/news-room/fact-sheets/detail/breast-cancer>. [↑](#endnote-ref-3)
8. Ruifrok AC, Johnston DA. Quantification of histochemical staining by color deconvolution. Anal Quant Cytol Histol. 2001 Aug;23(4):291-9. PMID: 11531144. [↑](#endnote-ref-4)
9. X. Ji, Y. Li, J. Cheng, Y. Yu and M. Wang, "Cell image segmentation based on an improved watershed algorithm," 2015 8th International Congress on Image and Signal Processing (CISP), Shenyang, China, 2015, pp. 433-437, doi: 10.1109/CISP.2015.7407919. [↑](#endnote-ref-5)
10. Internet geography drainage basins: <https://www.internetgeography.net/wp-content/uploads/2023/03/The-main-features-of-a-drainage-basin.png> [↑](#endnote-ref-6)
11. MRC CBU Wiki University of Cambridge: <https://imaging.mrc-cbu.cam.ac.uk/imaging/PrinciplesSmoothing> [↑](#endnote-ref-7)
12. Locci-Lopez, Daniel & Zhang, Rui & Oyem, Arnold & Castagna, John. (2018). The Multi-Scale Fourier Transform. 10.1190/segam2018-2994723.1. [↑](#endnote-ref-8)
13. R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural Features for Image Classification," in IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, no. 6, pp. 610-621, Nov. 1973, doi: 10.1109/TSMC.1973.4309314. [↑](#endnote-ref-9)
14. Elshakhs, Yahia S., et al. "A comprehensive survey on Delaunay triangulation: applications, algorithms, and implementations over CPUs, GPUs, and FPGAs." *IEEE Access* (2024) [↑](#endnote-ref-10)