**BenchMate Tool Mapping Document**

**Purpose**

This document defines the scientific purpose, expected data, processing logic, visualization settings, and implementation details for each initial tool available on BenchMate Benchtop.  
It ensures consistent backend and frontend development across all scientific domains, techniques, and tools.

**How to Read and Use This Document**

* Tools are organized as **Domain → Technique → Tool/Visualization**.
* Each Tool is mapped with the following fields:
  + Scientific purpose
  + Required file types and structures
  + Backend packages and processes
  + Frontend customization options
  + Statistical analyses where relevant
  + Download and AI extension possibilities
* Each Tool's mapping indicates which parts of the codebase it impacts.

**Architecture Context: Where Mapped Items Belong**

| **Document Field** | **Backend Folder** | **Frontend Folder** |
| --- | --- | --- |
| Config YAML (defaults/settings) | /backend/app/config/... | /frontend/src/config/... (or loaded dynamically) |
| Backend Data Processing | /backend/app/utils/... | N/A |
| API Endpoint (run analysis) | /backend/app/endpoints/... | /frontend/src/services/... (calls API) |
| Frontend UI Components | N/A | /frontend/src/components/..., /frontend/src/pages/... |
| Frontend Visualization | Data passed from backend | Plot rendering handled in frontend components |
| Downloadable Results | Backend prepares files | Frontend triggers download |
| AI Assistant Notes | Later AI integration | Later AI interaction modules |

**Tool Development Automation Strategy**

**Purpose**

As BenchMate expands across multiple scientific domains and techniques, we anticipate building and maintaining a large number of tools and visualizations.  
To streamline development, improve consistency, and enable rapid scaling, we are establishing an automation strategy for creating and updating new tools.

**Development Phases**

| **Phase** | **Description** | **When Implemented** |
| --- | --- | --- |
| **Phase 1: Golden Templates** | Build one fully clean, working set of backend and frontend files for a complete tool (e.g., Volcano Plot). This will serve as the standard for new tools. | Immediate |
| **Phase 2: Semi-Automated Generator Script** | Build a Python script that automatically copies the Golden Template, renames the files and classes/functions based on a new tool name (e.g., from "Volcano" to "Heatmap"), and fills basic placeholders. | After initial tool mapping and first few tools built |
| **Phase 3: Config-Driven Dynamic Loading** | Move toward a system where frontend and backend dynamically read YAML configuration files to define tool behavior, minimizing the need for hardcoded pages or endpoints for each new tool. | After MVP |

**Initial Tools and Associated Data Visualizations**

**1. Domain: Omics**

**1.1 Technique: Transcriptomics**

* Bulk RNA-seq
  + Volcano Plot
  + PCA Plot
  + Heatmap
  + MA Plot
  + Box Plot
* scRNA-seq (Single-cell RNA-seq)
  + UMAP Plot
  + t-SNE Plot
  + Cluster Heatmap
  + Violin Plot
  + Feature Scatter Plot

**1.2 Technique: Proteomics**

* Mass Spectrometry-Based Proteomics
  + Volcano Plot
  + PCA Plot
  + Heatmap
  + Protein-Protein Interaction Network (future)

**1.3 Technique: Metabolomics**

* LC-MS/MS Metabolomics
  + Volcano Plot
  + PCA Plot
  + Correlation Heatmap
  + Box Plot

**2. Domain: Imaging**

**Technique: Microscopy Imaging (Fiji/ImageJ Processing)**

* Bar Graphs (e.g., Cell counts, Area measurements)
* Scatter Plots (Object measurements)
* Line Graphs (Intensity profiles)
* Image Quantifications (Area, Intensity, Count summaries)

**3. Domain: Medical Imaging**

**Techniques: MRI, CT, Ultrasound**

* Segmentation Heatmaps
* Overlays/Comparison Maps
* Region of Interest (ROI) Analysis

**4. Domain: Calcium Imaging**

**Technique: GCaMP Recordings**

* ΔF/F0 Trace Plots
* Peak Detection Graphs
* Area Under Curve Graphs
* Event Frequency Histograms

**5. Domain: Electrophysiology**

**Techniques: Patch Clamp, MEA Recordings**

* Spike Raster Plots
* Firing Rate Histograms
* Burst Analysis Graphs
* I/V Curve Graphs

**6. Domain: Flow Cytometry**

**Technique: FACS Data Analysis**

* Dot Plots (2D scatter by markers)
* Gating Visualizations
* Histograms (Single-channel distributions)
* Pie Charts (Population percentages)

**Tool Mapping**

**Volcano Plot (Bulk RNA-Seq)**

| **Section** | **Description** |
| --- | --- |
| **Tool/Plot Name** | Volcano Plot |
| **Domain** | Omics |
| **Technique** | Bulk RNA-seq (Transcriptomics) |
| **Scientific Purpose** | Visualize differential expression by plotting log2 fold change against significance (-log10 p-value or adjusted p-value). Highlights upregulated and downregulated genes versus neutral genes. |
| **Expected Data Upload** | - File format: CSV or TSV  - Required columns:  • Gene Name  • Log2 Fold Change (log2FC)  • P-value  • Adjusted P-value (optional but recommended) |
| **User Variability Handling** | - Allow user to map uploaded columns to:  • X-axis (log2FC)  • Y-axis (p-value or adjusted p-value)  • Label column (gene names)  - Allow handling missing adjusted p-values (fallback to p-values) |
| **Required Backend Packages** | pandas, numpy, statsmodels, scipy |
| **Backend Processing Steps** | 1. Parse uploaded file. 2. Map user-selected columns. 3. Apply user-defined thresholds (e.g., log2FC threshold, p-value threshold). 4. Categorize genes into 'Upregulated', 'Downregulated', or 'Neutral'. 5. Return JSON with categorized data points and thresholds used. |
| **Related Statistical Analyses** | - P-value significance (typically < 0.05)  - Fold-change thresholds (log2FC > 1 or < -1, but customizable)  - Optional: Adjusted p-value filtering |
| **Frontend Required User Inputs** | - Upload file  - Column mapping selectors (X-axis, Y-axis, Labels)  - Set log2FC threshold  - Set p-value (or adjusted p-value) threshold  - Optional filters (e.g., minimum count, expression cutoff) |
| **Frontend Customization Options** | - Plot Title  - X and Y Axis Labels  - Point Size and Color Settings  - Upregulated/Downregulated/Neutral Point Colors  - Show/Hide gridlines  - Figure dimensions (height, width)  - Threshold lines visible or hidden |
| **Frontend Tooltips and Learning Notes** | - **Log2 Fold Change (log2FC):** Represents the magnitude of expression change between two groups. Positive means upregulated, negative means downregulated.  - **P-value:** Statistical probability the observed difference is random. Lower p-values suggest stronger evidence.  - **Threshold Lines:** Show cutoff points to highlight significantly changed genes visually.  - **Adjusted P-value (padj):** Corrects p-values for multiple comparisons, reducing false positives. |
| **Dynamic Recognition Rules** | - If uploaded file includes columns that match patterns like "log2FC", "pval", or "padj", suggest Volcano Plot automatically. |
| **Manual Selection Availability** | Yes — users can manually choose Volcano Plot if it is not auto-suggested. |
| **Download Options** | - Processed data table (CSV) showing categorization  - Volcano plot image (PNG, SVG)  - Filtered gene lists (e.g., Upregulated-only) |
| **AI Assistant Extensions (Future)** | - Summarize number of significant genes.  - Highlight top 10 upregulated and downregulated genes.  - Suggest additional analyses (e.g., pathway enrichment on significant genes). |
| **Notes/Future Extensions** | - Batch processing: Upload multiple comparisons and generate volcano plots side-by-side.  - Session versioning: Record thresholds and settings used.  - Metadata attachment: Tag experimental conditions (e.g., "Treatment vs Control").  - Example Dataset: DESeq2 sample output to demonstrate upload and plotting. |

**PCA Plot (Bulk RNA-seq)**

| **Section** | **Description** |
| --- | --- |
| **Tool/Plot Name** | PCA Plot (Principal Component Analysis) |
| **Domain** | Omics |
| **Technique** | Bulk RNA-seq (Transcriptomics) |
| **Scientific Purpose** | Visualizes sample clustering and variance across principal components. Reduces high-dimensional expression data into major patterns to detect batch effects, treatment separations, or sample similarities. |
| **Expected Data Upload** | - File format: CSV or TSV  - Required columns:  • Sample Names (row/column header)  • Gene expression values across samples |
| **User Variability Handling** | - Allow user to specify if samples are rows or columns (orientation toggle).  - Allow optional metadata upload (group labels: condition, treatment).  - Allow user to map group variable (for color-coding points by condition). |
| **Required Backend Packages** | pandas, numpy, sklearn (for PCA), matplotlib (optional for figure rendering if needed) |
| **Backend Processing Steps** | 1. Parse uploaded expression matrix.  2. Normalize data (optional scaling).  3. Run PCA (using scikit-learn).  4. Output principal components (PC1, PC2) for each sample.  5. Return JSON of sample coordinates and explained variance percentages. |
| **Related Statistical Analyses** | - Variance explained by PC1, PC2, etc.  - Optional group separation visualization (e.g., color by treatment group)  - Optional cluster analysis (future expansion) |
| **Frontend Required User Inputs** | - Upload expression matrix  - Toggle row vs column orientation  - Map metadata if available (for color grouping)  - Select which two PCs to plot (e.g., PC1 vs PC2) |
| **Frontend Customization Options** | - Plot Title  - X and Y Axis Labels (PC1 %, PC2 % displayed)  - Point Color Settings (by group)  - Point Size Settings  - Gridlines on/off  - Figure size (height/width) |
| **Frontend Tooltips and Learning Notes** | - **Principal Component (PC):** A new variable created by combining multiple features to explain maximum variance.  - **Variance Explained:** Percent of total data variation captured by each PC. Higher variance = more important pattern.  - **Grouping by Metadata:** Helps visualize whether experimental groups cluster together. |
| **Dynamic Recognition Rules** | - If uploaded matrix contains large numeric table (>50 features) → suggest PCA Plot automatically. |
| **Manual Selection Availability** | Yes — users can manually choose PCA Plot even if not auto-suggested. |
| **Download Options** | - PCA Coordinates table (CSV)  - PCA Plot image (PNG, SVG) |
| **AI Assistant Extensions (Future)** | - Summarize variance captured by first two PCs.  - Highlight any visible group separation.  - Suggest exploring additional PCs if variance is low. |
| **Notes/Future Extensions** | - Batch PCA: Run PCA across multiple batches automatically.  - 3D PCA: Allow PC1 vs PC2 vs PC3 visualizations.  - Link samples to metadata hover-over tooltips.  - Provide example dataset (RNA-seq normalized count table). |

**Heatmap (Bulk RNA-seq)**

| **Section** | **Description** |
| --- | --- |
| **Tool/Plot Name** | Heatmap |
| **Domain** | Omics |
| **Technique** | Bulk RNA-seq (Transcriptomics) |
| **Scientific Purpose** | Visualize gene expression patterns across samples using a color-coded matrix. Helps detect clustering, co-expression, and overall expression trends. |
| **Expected Data Upload** | - File format: CSV or TSV  - Required structure:  • Rows = Genes  • Columns = Samples  • Values = Expression levels (e.g., normalized counts, TPM, FPKM) |
| **User Variability Handling** | - Allow user to toggle row vs column orientation if necessary.  - Allow user to select subsets of genes or samples for visualization.  - Allow user to upload optional metadata for side annotations (e.g., treatment groups, batches). |
| **Required Backend Packages** | pandas, numpy, seaborn (for heatmap generation), matplotlib |
| **Backend Processing Steps** | 1. Parse uploaded expression matrix. 2. Optionally scale or normalize data. 3. Allow subsetting of selected genes or samples. 4. Return JSON: gene names, sample names, matrix values. 5. If server-side plotting needed: generate clustered heatmap (clustermap) if requested. |
| **Related Statistical Analyses** | - Optional: Hierarchical clustering of genes and/or samples (e.g., dendrograms). - Optional: Z-score normalization across rows (genes) for standardization. |
| **Frontend Required User Inputs** | - Upload expression matrix.  - Select genes or samples (optional filtering).  - Choose clustering: genes only, samples only, both, or none.  - Select scaling: none, row normalization, column normalization. |
| **Frontend Customization Options** | - Plot Title  - Color Palette Selector (e.g., Viridis, Coolwarm, Reds)  - Show/Hide Dendrograms  - Axis Font Size Adjustment  - Cell Color Range (custom min/max scaling)  - Figure Size (Height, Width) |
| **Frontend Tooltips and Learning Notes** | - **Heatmap:** A graphical representation where individual values are represented as colors to show relative magnitude.  - **Clustering:** Groups genes or samples based on similarity in expression profiles. Useful for discovering patterns.  - **Scaling/Normalization:** Adjusts values so that patterns are easier to compare visually across genes or samples. |
| **Dynamic Recognition Rules** | - If uploaded data matrix has >2 samples and numeric values → suggest Heatmap automatically. |
| **Manual Selection Availability** | Yes — users can manually select Heatmap generation. |
| **Download Options** | - Heatmap Plot image (PNG, SVG)  - Scaled expression matrix (optional)  - Cluster order (list of genes/samples by cluster) |
| **AI Assistant Extensions (Future)** | - Summarize major gene clusters.  - Identify top upregulated/downregulated gene groups.  - Suggest cluster-based downstream analysis (e.g., pathway enrichment). |
| **Notes/Future Extensions** | - Multi-heatmap comparison (e.g., multiple conditions side-by-side).  - Interactive zoom into sub-clusters.  - Metadata-driven annotation bars alongside rows/columns (e.g., Treatment Group).  - Example dataset: Small normalized RNA-seq matrix with 50 genes × 6 samples. |

**MA Plot (Bulk RNA-seq)**

| **Section** | **Description** |
| --- | --- |
| **Tool/Plot Name** | MA Plot (Mean vs. log ratio plot) |
| **Domain** | Omics |
| **Technique** | Bulk RNA-seq (Transcriptomics) |
| **Scientific Purpose** | Visualizes the relationship between the average expression (mean) and the log fold change across genes. Helps detect systematic bias and overall differential expression patterns between conditions. |
| **Expected Data Upload** | - File format: CSV or TSV  - Required columns:  • Gene Name  • Mean Expression (A-values)  • Log2 Fold Change (M-values)  • P-value or Adjusted P-value (optional but recommended) |
| **User Variability Handling** | - Allow user to map uploaded columns:  • X-axis (Mean Expression)  • Y-axis (Log2 Fold Change)  • Label (Gene Name)  • P-value (optional) for coloring significant genes. |
| **Required Backend Packages** | pandas, numpy, statsmodels (optional if thresholding by significance) |
| **Backend Processing Steps** | 1. Parse uploaded file. 2. Map user-selected columns. 3. (Optional) Apply significance filtering based on p-value. 4. Categorize genes if thresholds are provided (optional up/down coloring). 5. Return JSON with plotting points, labels, and categorization. |
| **Related Statistical Analyses** | - Fold Change Assessment (M-values)  - Significance by p-value (optional highlighting) |
| **Frontend Required User Inputs** | - Upload file  - Map X-axis and Y-axis columns  - Set optional p-value threshold (if coloring by significance)  - Toggle whether to highlight significant genes or not |
| **Frontend Customization Options** | - Plot Title  - X and Y Axis Labels (editable)  - Point Size and Color Settings  - Highlight Color for Significant Genes  - Gridlines on/off  - Figure Dimensions (Height/Width) |
| **Frontend Tooltips and Learning Notes** | - **MA Plot:** A plot of log2 fold change (M) vs average expression (A). Used to detect differential expression patterns, systematic biases, and spread of data.  - **Mean Expression (A-value):** The average expression level of a gene across samples.  - **Log2 Fold Change (M-value):** The ratio of expression between two conditions, logged to base 2 for scale symmetry. |
| **Dynamic Recognition Rules** | - If uploaded file has one column with "mean" and one with "logFC" or similar → suggest MA Plot automatically. |
| **Manual Selection Availability** | Yes — users can manually select MA Plot even if not suggested. |
| **Download Options** | - Processed dataset (CSV)  - MA Plot image (PNG, SVG) |
| **AI Assistant Extensions (Future)** | - Summarize global fold change trends (e.g., "Most genes unchanged, small subset highly upregulated.")  - Identify bias or shift if detected (e.g., more genes downregulated) |
| **Notes/Future Extensions** | - Option to display LOWESS smoothing curve (to show overall trend).  - Batch MA plots for multiple comparisons.  - Metadata tags for comparisons shown (e.g., "Treatment A vs Treatment B").  - Example dataset: Simulated RNA-seq differential expression table with mean and log2FC columns. |

**Box Plot (Bulk RNA-seq)**

| **Section** | **Description** |
| --- | --- |
| **Tool/Plot Name** | Box Plot |
| **Domain** | Omics |
| **Technique** | Bulk RNA-seq (Transcriptomics) |
| **Scientific Purpose** | Compare the expression levels of specific genes (or groups of genes) across different experimental conditions. Summarizes data distribution (median, quartiles, outliers) visually. |
| **Expected Data Upload** | - File format: CSV or TSV  - Required columns:  • Sample Identifier  • Condition/Group  • Gene Expression Value  - Optionally: multiple genes if multi-gene comparison needed (advanced later) |
| **User Variability Handling** | - Allow user to map columns:  • Sample ID  • Group/Condition  • Expression Value  - Allow user to select one or multiple genes if dataset is long-form |
| **Required Backend Packages** | pandas, numpy, scipy.stats (for statistical testing, optional) |
| **Backend Processing Steps** | 1. Parse uploaded data. 2. Map user-selected columns. 3. Prepare data for box plot aggregation by group. 4. (Optional) Run statistical tests (e.g., t-test, ANOVA) if requested. 5. Return JSON with grouping, summary statistics, and optionally p-values. |
| **Related Statistical Analyses** | - t-test (2 groups)  - ANOVA (multiple groups)  - Mann-Whitney U-test (non-parametric) |
| **Frontend Required User Inputs** | - Upload file  - Select sample ID, group label, expression value columns  - Choose which genes (or all) to display  - Optionally enable statistical comparison |
| **Frontend Customization Options** | - Plot Title  - X-axis (Group) Label  - Y-axis (Expression) Label  - Box color by group  - Point overlay (show individual data points) on/off  - Statistical test selection (if available)  - Figure size (height/width) |
| **Frontend Tooltips and Learning Notes** | - **Box Plot:** Displays distribution summary statistics (median, quartiles) for each group. - **Median:** Middle value of the dataset. - **IQR (Interquartile Range):** Range between first and third quartiles (25%-75%). - **Outliers:** Points outside 1.5× IQR are often plotted separately. - **Statistical Test:** Tests whether differences between groups are statistically significant. |
| **Dynamic Recognition Rules** | - If uploaded file contains 'group' and 'expression' type columns → suggest Box Plot automatically. |
| **Manual Selection Availability** | Yes — users can manually select Box Plot even if not suggested. |
| **Download Options** | - Processed data summary (CSV)  - Box Plot image (PNG, SVG)  - Statistical comparison results (optional) |
| **AI Assistant Extensions (Future)** | - Summarize whether groups show significant differences.  - Recommend additional subgroup comparisons.  - Explain choice between t-test, ANOVA, or non-parametric tests based on data characteristics. |
| **Notes/Future Extensions** | - Multi-gene comparison box plots (faceted or grouped box plots).  - Interactive statistical test result overlays (e.g., p-value stars above boxes).  - Example dataset: Expression levels of 2–3 genes across 2–3 conditions. |

**Imaging**

**Purpose**

BenchMate’s Imaging Module builds upon the foundational scientific imaging workflows established by ImageJ and Fiji, with significant enhancements to usability, reproducibility, integration, and scientific data analysis.

We are **not copying** ImageJ/Fiji interfaces or internal code directly,  
instead, we are **recreating the functional workflows and improving them** in a modern, user-friendly, no-code environment.

**Open-Source Licensing Context**

* ImageJ and Fiji are open-source under the GPL license.
* Access to ImageJ/Fiji source code is public:
  + [ImageJ Source Code Documentation](https://imagej.net/develop/source)
  + [ImageJ GitHub Repository](https://imagej.net/develop/github)
* Under GPL, we are allowed to:
  + Access and study the original source code.
  + Adapt the workflow logic for our platform.
  + Commercialize our platform as long as derivative GPL components (if any are directly reused) respect GPL distribution obligations.
* **BenchMate will write all original code** (backend and frontend) but **use ImageJ/Fiji’s scientific workflows as references** for ensuring scientific rigor and completeness.

**1. Core Principles**

| **Principle** | **Description** |
| --- | --- |
| Modular Processing System | Users apply modular image operations in any order. Each step recorded if recording enabled. |
| Live Workflow Recording | Recording is attached to the Modular Panel with an ON/OFF toggle. Steps saved as macros (JSON/YAML). |
| Immediate Data Extraction and Analysis | Measurements directly available for visualization/statistical analysis without external exports. |
| Metadata Preservation | Pixel size, units, channels, bit depth metadata extracted and preserved across all processing and exports. |
| Deep Learning Model Support | Optional StarDist and Cellpose deep learning models downloaded on demand for advanced segmentation. |
| Full Visualization and Analysis Customization | Users can fully customize graphs and choose statistical tests appropriate to their experiment design. |
| Hover Tooltips and Teaching | Every tool, setting, and analysis option has an explainer tooltip. |
| Lightweight Core with Advanced Mode | Optional Fiji plugin access and 3D imaging extensions planned after MVP polish. |

**2. Supported File Formats**

* TIFF, multi-page TIFF
* CZI (Zeiss)
* JPEG, JPG
* PNG
* BMP
* GIF
* PGM
* DICOM
* FITS
* Others via Bio-Formats Plugin (future expansions)

Users can **upload, process, and export** in these formats.

**3. User Journey Flow**

| **Step** | **Description** |
| --- | --- |
| 1. Upload Images | Upload single/multiple images; extract metadata. |
| 2. (Optional) Start Workflow Recording | Toggle recording ON/OFF, record all operations with parameters. |
| 3. Apply Modular Functions | Choose any processing function in any order; parameter customizable. |
| 4. Data Extraction | Measurements (Area, Intensity, Shape Descriptors) generated when applicable. |
| 5. Visualization and Analysis | Users toggle option to visualize data; select columns; customize plots; run statistical tests. |
| 6. Save/Export | Export processed images, tables, plots, workflows. |
| 7. Macro Replay | Replay saved workflows across batches or single images. |

Everything modular, flexible, reproducible.

**4. Processing Functions Planned**

(Already listed extensively above.)

Includes all Fiji core functions + modern DL segmentation.

**5. Visualization and Analysis**

* Full control over plot types, axes, colors, figure size.
* Integrated basic and advanced statistical tests.
* Teach users what tests/plots are appropriate (hover tooltips + optional "Learn More" popups).
* Future AI suggestion system.

**6. Advanced User Support**

* Fiji plugin access planned (optional Advanced Mode toggle).
* Future community plugin system.
* 3D visualization and analysis support for Z-stacks.

**Developer Notes**

* Follow modular coding structure: upload layer → processing layer → analysis layer → export layer.
* Reference ImageJ/Fiji GitHub/documentation for each re-implemented function.
* Validate key measurement functions scientifically.
* Keep system lightweight unless Advanced Mode activated.