Spatial Lymph Node Stratified Map

An Interactive Atlas Resolving Overplotting in High-Dimensional Tissue Datasets

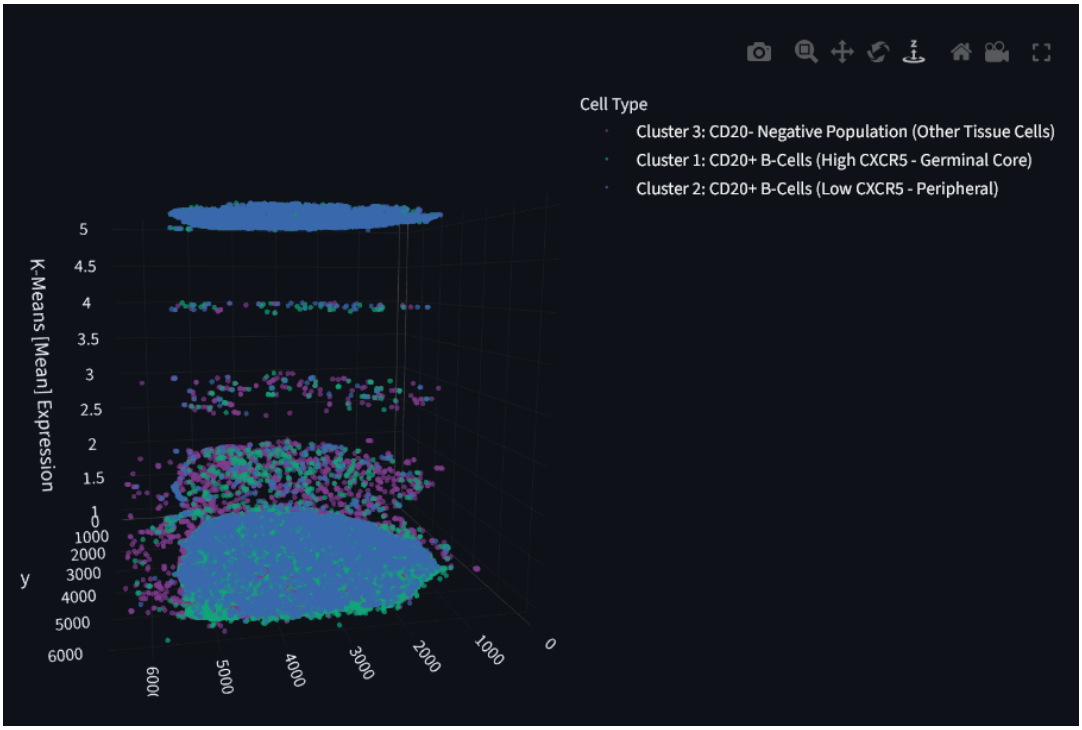
Visualization Tools:	Python, Streamlit, Plotly Express
Intended Audience:	Translational Researchers and Clinical Immunologists
Specific Dataset:	CFDE / HuBMAP Spatial Proteomics Reconstructive Micrography (SPRM) pipeline (Dataset ID: HBM927.KDXB.445)

Description

What questions were you trying to answer?

When I first opened the spreadsheet of 36,124 individual cells from the HuBMAP pipeline, looking at endless rows of numbers and complex column names like `K-means [tsne_all_features]` left me feeling a bit out of my depth. Without real-world context, numbers alone cannot tell a story. Even though I understood the basic ideas behind mathematical clustering, I couldn't picture how those abstract values related to the physical structure of a human organ. I wanted to answer a fundamental question: how can we translate complex, raw pipeline data into an impactful visual interface that lets a clinician see underlying diagnostic patterns in a patient's tissue?

Figure 1: Interactive 3D Stratified Streamlit Dashboard Output



How did you analyze or process the data?

The data engineering workflow was built entirely in Python. To bring the data together, the physical cell location indices (`cell_centers.csv`) and the high-dimensional expression matrices (`cell_clusters.csv`) were combined using an inner join on their unique cellular tracking IDs.

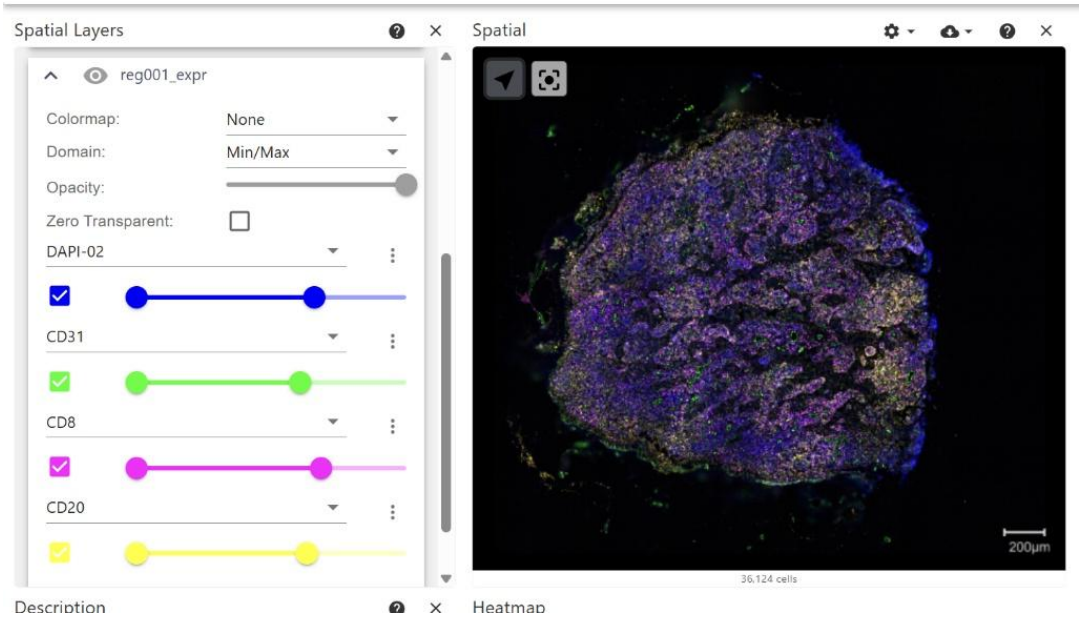
Since this dataset is so massive, trying to view all 36,124 cells on a normal, flat 2D plot creates severe overplotting, which resulted in a crowded, messy blur where different cell populations completely mask one another. To solve this, I decided to incorporate a 3D aspect to give the data more breathing room.

I wrote a dictionary to map the abstract string categories to their true biological names, filtering the data to isolate the critical B-cell populations:

- **Cluster 1:** CD20+ B-Cells (High CXCR5 – Germinal Core)
- **Cluster 2:** CD20+ B-Cells (Low CXCR5 – Peripheral Border)
- **Cluster 3:** CD20– Negative Environment

By using the cluster assignments as distinct floating layers along the vertical Z-axis, the overlapping cell groups pull apart. I optimized the marker sizes to a crisp, microscopic radius (`size=2`) and applied high-contrast qualitative colors to make the different protein expressions stand out distinctly.

Figure 2: Official HuBMAP Fluorescent Microscopy Spatial Map Overlay



What should viewers learn from this?

Viewers will learn how the distribution of different proteins changes across a tissue sample in real time. By manipulating the simple sidebar filters in the Streamlit app, users can peel back the crowded layers to observe exactly where specific cell types gather.

When you look at the app, the floating dots instantly map out the true microanatomy of a human lymph node. It clearly reveals a dense central core called the Lymphoid Follicle (Germinal Center), driven by the high CXCR5 expression of Cluster 1, that is surrounded by the migrating paths of Cluster 2. Viewers will see that by working with this data, they are essentially performing a visual inspection of a real tissue slice to find correlations. Ultimately, creating this image matters because it has real impact: it bridges the gap between cold code and actual meaning, turning a technical spreadsheet into an intuitive, visual tool that can help clinicians find answers to big questions and guide patient diagnoses.

References

National Institutes of Health Common Fund Data Ecosystem. (2026). *Common Fund Data Ecosystem Portal. Multi-omics Data Infrastructure.* https://www.cfde.science/

HuBMAP Consortium. (2023). *Spatial Proteomics Reconstructive Micrography (SPRM) Pipeline Dataset for Human Lymph Node* (Dataset ID: HBM927.KDXB.445). Global Common Fund Data Ecosystem. https://doi.org/10.35079/HBM927.KDXB.445

Appendix: Technical Source Code

Script 1: interpret_data.py

This script loads the structural cell center matrix metadata and cell cluster identification files, then executes an inner join on the indexed cell identity boundaries.

```
import pandas as pd

# Load the cell center and cell cluster data from CSV files
cell_center_df = pd.read_csv('reg001_expr.ome.tiff-cell_centers.csv')
cell_cluster_df = pd.read_csv('reg001_expr.ome.tiff-cell_cluster.csv')

# Merge the two dataframes on the 'cell_id' column
merged_df = pd.merge(cell_center_df, cell_cluster_df, on='ID', how='inner')

# Print column names and first few rows of the dataframes
print('\n----- Cell Center DataFrame -----')
print(cell_center_df.columns)
print('\n----- Cell Cluster DataFrame -----')
print(cell_cluster_df.columns)
print('\n----- Merged DataFrame -----')
print(merged_df.columns)

merged_df.to_csv('merged_data.csv', index=False)
```

Script 2: spatial_lymph_node_stratified_map.py

This interface script processes the integrated files, assigns real biological cell phenotype titles via dictionary mapping, and deploys the final interactive Streamlit web dashboard visualization framework.

```
import streamlit as st
import plotly.express as px
import pandas as pd

# 1. Load the merged DataFrame
merged_df = pd.read_csv('merged_data.csv')

# Define our key data column name
cluster_column = 'K-Means [tSNE_All_Features]'

# 2. Map the raw cluster numbers to their decoded protein identities
protein_names = {
    1: "Cluster 1: CD20+ B-Cells (High CXCR5 - Germinal Core)",
    2: "Cluster 2: CD20+ B-Cells (Low CXCR5 - Peripheral)",
    3: "Cluster 3: CD20- Negative Population (Other Tissue Cells)"
}

# Create a brand new column named 'Cell Type' filled with the clean biological names
merged_df['Cell Type'] = merged_df[cluster_column].map(protein_names)

# Force the column to be a string so Plotly treats it as high-contrast categories
merged_df['Cell Type'] = merged_df['Cell Type'].astype(str)

# Create Streamlit app title and description
st.title('Spatial Lymph Node Protein Stratified Map')

st.markdown("""
### Interactive Cellular Atlas
This dashboard visualizes **36,124 individual cells** from a human lymph node dataset,
processed via the HuBMAP SPRM pipeline.

* **Physical Layout (X & Y Axes):** These map out the exact location of every single
  cell on the original tissue slide.
* **The 3D Lever (Z Axis):** You can toggle between a flat, traditional view and a 3D
  view. The 3D view separates overlapping cell clusters for clarity.
```

```
* **Color Profiles:** The colors highlight distinct groups of cells based on the
  unique protein mixture found in each individual cell.
""")

# Sidebar Widget: Let users filter what cell types they want to display
selected_display_names = st.sidebar.multiselect(
    "Select Cellular Phenotype / Protein Filter:",
    options=list(protein_names.values()),
    default=list(protein_names.values())
)

# Sidebar: Let the user choose what the Z-axis represents!
z_axis_choice = st.sidebar.selectbox(
    "Select Z-Axis Perspective",
    options=["K-Means [Mean] Expression", "Flat 2D View"]
)

# If 'Flat 2D View' is chosen, we pass 'z' (default flat plane coordinate)
z_val = z_axis_choice if z_axis_choice != "Flat 2D View" else 'z'

# Filter the dataframe based on what the user selected in the multiselect sidebar
filtered_df = merged_df[merged_df['Cell Type'].isin(selected_display_names)]

# Create the optimized 3D scatter plot
fig = px.scatter_3d(
    filtered_df,
    x='x',
    y='y',
    z=z_val,
    color='Cell Type',
    hover_data=['ID'],
    color_discrete_sequence=px.colors.qualitative.Bold  # Uses high-contrast categorical
        colors
)

# Shrink the marker size so 36,000+ points look crisp and distinct
fig.update_traces(marker=dict(size=2, opacity=0.7))

# Clean up layout dimensions to maximize visual space
fig.update_layout(margin=dict(l=0, r=0, b=0, t=40))

# 6. Display the plot in the app
st.plotly_chart(fig, use_container_width=True)
```