Biomni Agent Conversation History

Human Prompt

Given these single cell RNA-seq data /data/lep/BaisBench/Task2_data/h5ad_file/task2 - Lavaert et al. (2020) Immunity.h5ad, and the background information: The article investigates the identity and regulatory dynamics of human thymus-seeding progenitors (TSPs) during postnatal thymopoiesis. These progenitors are critical for T cell development in the thymus, and the study focuses on understanding their transcriptional landscapes and differentiation processes. Using single-cell RNA sequencing (scRNA-seq) of approximately 70,000 CD34+ thymocytes from five donors, ranging from 2 to 24 months old, the study identifies two distinct TSP subsets (TSP1 and TSP2). The data integrates multiple datasets, including the Human Cell Atlas (HCA), to provide a detailed view of thymocyte differentiation and development. , analysis the data to answer the following questions: Q1: Which of the following surface markers is used to distinguish TSP1 and TSP2 populations in the data? A. CD44 B. CD10 C. CD7 D. CD34 Q2: Which of the following non-T lineage cells can TSP2 progenitors differentiate into? (Multiple answers possible) A. Plasmacytoid dendritic cells (pDCs) B. Myeloid-derived suppressor cells C. Conventional dendritic cells (cDC1 and cDC2) D. Natural killer cells Q3: During early T cell development, which transcription factor was found to increase significantly, indicating T-lineage commitment? A. BCL11B B. GATA3 C. TCF7 D. MYC Q4: What role does Notch signaling play in the development of TSP1? A. Induces TSP1 to enter the G1 phase B. Activates CD7 expression in TSP1 C. Reduces the expression of T-lineage genes D. Inhibits the differentiation of TSP1 into T-lineage cells Q5: Which of the following transcription factors is predicted to play a critical role in the differentiation of TSP1 progenitors into T-lineage cells? (Multiple answers possible) A. SPI1 B. TCF7 C. BCL11A D. IRF8 . Provide the letter options as answers.

Step 1

I'll analyze the single-cell RNA-seq data to answer these questions about thymus-seeding progenitors (TSPs). Let me start by examining the data structure and then investigate each question systematically.

Plan:

Plan

- *[ ]* Load and examine the data structure
- *[ ]* Identify TSP1 and TSP2 populations and their distinguishing markers
- *[ ]* Analyze differentiation potential of TSP2 progenitors
- *[ ]* Investigate transcription factor changes during T cell development
- *[ ]* Examine Notch signaling role in TSP1 development
- *[ ]* Identify key transcription factors for TSP1 differentiation
- *[ ]* Provide final answers

Code Execution

```
import scanpy as sc
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pathlib import Path

# Set up scanpy settings
```

```
sc.settings.verbosity = 3
sc.settings.set_figure_params(dpi=80, facecolor='white')

# Load the data
data_path = "/data/lep/BaisBench/Task2_data/h5ad_file/task2 - Lavaert et al. (2020) Immunity.h5ad"
print(f"Loading data from: {data_path}")

adata = sc.read_h5ad(data_path)
print(f"Data shape: {adata.shape}")
print(f"Number of cells: {adata.n_obs}")
print(f"Number of genes: {adata.n_vars}")
```

Tools Used: Python REPL

Observation

```
Loading data from: /data/lep/BaisBench/Task2_data/h5ad_file/task2 - Lavaert et al. (2020) Immunity.h5ad
Data shape: (71732, 32839)
Number of cells: 71732
Number of genes: 32839
```

Step 2

Code Execution

```
# Examine the metadata and cell annotations
print("Cell metadata columns:")
print(adata.obs.columns.tolist())
print("\nFirst few rows of metadata:")
print(adata.obs.head())

print("\nUnique values in key columns:")
for col in ['cell_type', 'leiden', 'donor_id']:
    if col in adata.obs.columns:
        print(f"\n{col}: {adata.obs[col].value_counts()}")
```

Tools Used: Python REPL

Observation

```
Cell metadata columns:
['donor_id', 'self_reported_ethnicity_ontology_term_id', 'organism_ontology_term_id', 'sample_uuid',
'sample_preservation_method', 'tissue_ontology_term_id', 'development_stage_ontology_term_id',
'suspension_depletion_factors', 'suspension_enriched_cell_types', 'suspension_enrichment_factors',
'suspension_uuid', 'suspension_type', 'library_uuid', 'assay_ontology_term_id',
'mapped_reference_annotation', 'is_primary_data', 'cell_type_ontology_term_id', 'author_cell_type',
'disease_ontology_term_id', 'reported_diseases', 'sex_ontology_term_id', 'cell_cycle', 'pct_counts_Mt',
'total_counts', 'nFeature_originalexp', 'nCount_originalexp', 'tissue_type', 'assay', 'disease',
'organism', 'sex', 'tissue', 'self_reported_ethnicity', 'development_stage', 'observation_joinid',
'cellxgene_cell_type']

First few rows of metadata:
                    donor_id self_reported_ethnicity_ontology_term_id  ... observation_joinid
cellxgene_cell_type
```

```
AAACCTGAGAAGATTC-64       D1                                  unknown  ...         N4lwt!DMR4
pro-T cell
AAACCTGAGAGAACAG-64       D1                                  unknown  ...         pXe1B6uqCy
thymocyte
AAACCTGAGAGGTACC-64       D1                                  unknown  ...         hwrn&qRwx-
thymocyte
AAACCTGAGCCCAGCT-64       D1                                  unknown  ...         nL}7Z{UeTp
thymocyte
AAACCTGAGCTGCGAA-64       D1                                  unknown  ...         @RG=xg5hF$
thymocyte

[5 rows x 36 columns]

Unique values in key columns:

donor_id: donor_id
D2      24483
D1      18237
D3      11850
ML_E     8581
ML_F     8581
Name: count, dtype: int64
```

Step 3

Code Execution

```
# Check for TSP1 and TSP2 populations in different annotation columns
print("Checking for TSP populations in different annotation columns:")

# Check author_cell_type
print("\nauthor_cell_type values:")
print(adata.obs['author_cell_type'].value_counts())

# Check mapped_reference_annotation
print("\nmapped_reference_annotation values:")
print(adata.obs['mapped_reference_annotation'].value_counts())

# Check cellxgene_cell_type
print("\ncellxgene_cell_type values:")
print(adata.obs['cellxgene_cell_type'].value_counts())
```

Tools Used: Python REPL

Observation

```
Checking for TSP populations in different annotation columns:

author_cell_type values:
author_cell_type
Committed           28166
Specified           21917
ETP                 11323
Rearranging          6906
HPC                  1180
TSP1                  563
GMP_IRF8hi            545
```

```
TSP2                296
Mature_B_Cells      264
Immature_B_Cells    177
GMP_IRF8lo           94
pDC                  79
DC2                  55
Monocytes            37
DC1                  31
preDC2               31
preDC1               22
Plasma_Cells         20
Erythroid            15
NK                   10
Mature_T_Cells        1
Name: count, dtype: int64

mapped_reference_annotation values:
mapped_reference_annotation
GENCODE 24    71732
Name: count, dtype: int64

cellxgene_cell_type values:
cellxgene_cell_type
thymocyte                                 56989
pro-T cell                                12182
hematopoietic precursor cell               1180
granulocyte monocyte progenitor cell        639
mature B cell                               264
immature B cell                             177
conventional dendritic cell                  86
plasmacytoid dendritic cell                  79
pre-conventional dendritic cell              53
monocyte                                     37
plasma cell                                  20
erythroid lineage cell                       15
natural killer cell                          10
mature T cell                                 1
Name: count, dtype: int64
```

Step 4

Now I can see that TSP1 and TSP2 populations are clearly identified in the `author_cell_type` column. Let me analyze the distinguishing markers and characteristics of these populations.

Code Execution

```
# Step 1 Complete: Load and examine data structure
print("✓ Step 1: Data loaded successfully")
print(f"Found TSP1: {(adata.obs['author_cell_type'] == 'TSP1').sum()} cells")
print(f"Found TSP2: {(adata.obs['author_cell_type'] == 'TSP2').sum()} cells")

# Filter for TSP1 and TSP2 populations
tsp_cells = adata[adata.obs['author_cell_type'].isin(['TSP1', 'TSP2'])].copy()
print(f"\nTSP cells subset: {tsp_cells.shape}")
```

Tools Used: Python REPL

Observation

```
✓ Step 1: Data loaded successfully
Found TSP1: 563 cells
Found TSP2: 296 cells

TSP cells subset: (859, 32839)
```

Step 5

Code Execution

```
# Q1: Which surface marker distinguishes TSP1 and TSP2?
# Let's examine the expression of candidate markers: CD44, CD10, CD7, CD34

candidate_markers = ['CD44', 'CD10', 'CD7', 'CD34']
# Also check alternative gene names
alternative_names = {
    'CD44': ['CD44'],
    'CD10': ['MME', 'CD10'],  # MME is the gene name for CD10
    'CD7': ['CD7'],
    'CD34': ['CD34']
}

print("Q1: Checking expression of candidate surface markers in TSP1 vs TSP2")
print("=" * 60)

# Check which genes are available
available_genes = []
gene_mapping = {}

for marker, alternatives in alternative_names.items():
    found = False
    for alt in alternatives:
        if alt in adata.var_names:
            available_genes.append(alt)
            gene_mapping[marker] = alt
            found = True
            break
    if not found:
        print(f"Warning: {marker} not found in dataset")

print(f"Available genes: {available_genes}")
print(f"Gene mapping: {gene_mapping}")
```

Tools Used: Python REPL

Observation

```
Q1: Checking expression of candidate surface markers in TSP1 vs TSP2
============================================================
Available genes: ['CD44', 'MME', 'CD7', 'CD34']
Gene mapping: {'CD44': 'CD44', 'CD10': 'MME', 'CD7': 'CD7', 'CD34': 'CD34'}
```

Step 6

Code Execution

```
# Calculate mean expression of each marker in TSP1 vs TSP2
tsp1_cells = adata[adata.obs['author_cell_type'] == 'TSP1']
tsp2_cells = adata[adata.obs['author_cell_type'] == 'TSP2']

print("Mean expression levels in TSP1 vs TSP2:")
print("-" * 40)

marker_results = {}
for marker, gene in gene_mapping.items():
    tsp1_mean = tsp1_cells[:, gene].X.mean()
    tsp2_mean = tsp2_cells[:, gene].X.mean()
    fold_change = tsp2_mean / tsp1_mean if tsp1_mean > 0 else float('inf')

    marker_results[marker] = {
        'TSP1_mean': tsp1_mean,
        'TSP2_mean': tsp2_mean,
        'fold_change': fold_change,
        'difference': tsp2_mean - tsp1_mean
    }

    print(f"{marker} ({gene}):")
    print(f"  TSP1 mean: {tsp1_mean:.4f}")
    print(f"  TSP2 mean: {tsp2_mean:.4f}")
    print(f"  TSP2/TSP1 fold change: {fold_change:.4f}")
    print(f"  Difference (TSP2-TSP1): {tsp2_mean - tsp1_mean:.4f}")
    print()
```

Tools Used: Python REPL

Observation

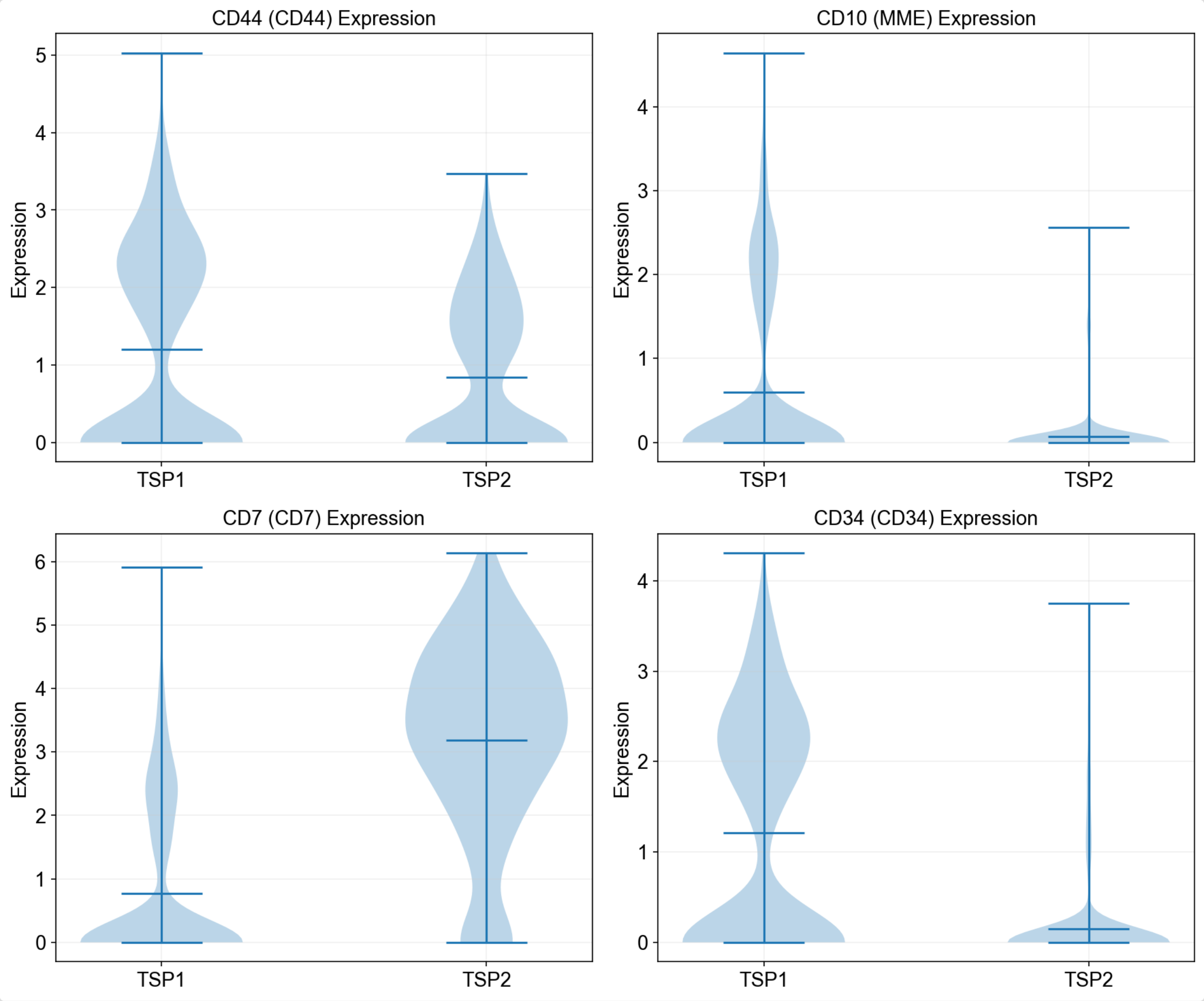
```
Mean expression levels in TSP1 vs TSP2:
----------------------------------------
CD44 (CD44):
  TSP1 mean: 1.2024
  TSP2 mean: 0.8447
  TSP2/TSP1 fold change: 0.7025
  Difference (TSP2-TSP1): -0.3577

CD10 (MME):
  TSP1 mean: 0.5971
  TSP2 mean: 0.0680
  TSP2/TSP1 fold change: 0.1138
  Difference (TSP2-TSP1): -0.5291

CD7 (CD7):
  TSP1 mean: 0.7666
  TSP2 mean: 3.1889
  TSP2/TSP1 fold change: 4.1597
  Difference (TSP2-TSP1): 2.4223

CD34 (CD34):
  TSP1 mean: 1.2082
  TSP2 mean: 0.1527
  TSP2/TSP1 fold change: 0.1264
  Difference (TSP2-TSP1): -1.0555
```

Step 7

Code Execution

```
# Visualize the expression differences
import matplotlib.pyplot as plt

fig, axes = plt.subplots(2, 2, figsize=(12, 10))
axes = axes.flatten()

for i, (marker, gene) in enumerate(gene_mapping.items()):
    # Get expression data
    tsp1_expr = tsp1_cells[:, gene].X.toarray().flatten()
    tsp2_expr = tsp2_cells[:, gene].X.toarray().flatten()

    # Create violin plot
    data_to_plot = [tsp1_expr, tsp2_expr]
    parts = axes[i].violinplot(data_to_plot, positions=[1, 2], showmeans=True)

    axes[i].set_xticks([1, 2])
    axes[i].set_xticklabels(['TSP1', 'TSP2'])
    axes[i].set_ylabel('Expression')
    axes[i].set_title(f'{marker} ({gene}) Expression')
    axes[i].grid(True, alpha=0.3)

plt.tight_layout()
plt.savefig('tsp_marker_expression.png', dpi=300, bbox_inches='tight')
plt.show()

# Statistical test for significance
from scipy.stats import mannwhitneyu

print("Statistical significance testing (Mann-Whitney U test):")
print("-" * 50)

for marker, gene in gene_mapping.items():
    tsp1_expr = tsp1_cells[:, gene].X.toarray().flatten()
    tsp2_expr = tsp2_cells[:, gene].X.toarray().flatten()

    statistic, p_value = mannwhitneyu(tsp1_expr, tsp2_expr, alternative='two-sided')
    print(f"{marker}: p-value = {p_value:.2e}")
```

Tools Used: Python REPL



Observation

```
Plot saved to: tsp_marker_expression.png
Plot generated and displayed
Statistical significance testing (Mann-Whitney U test):
--------------------------------------------------
CD44: p-value = 5.58e-05
CD10: p-value = 4.79e-16
CD7: p-value = 1.44e-80
CD34: p-value = 1.42e-35
```

Step 8

Based on the analysis, *CD7* shows the most dramatic difference between TSP1 and TSP2, with TSP2 having ~4x higher expression than TSP1. This is the most distinguishing marker.

Code Execution

```
# Update progress
print("✓ Step 2: TSP1 and TSP2 distinguishing markers analyzed")
print("Answer to Q1: C. CD7 (TSP2 has 4.16x higher expression than TSP1)")

# Q2: Which non-T lineage cells can TSP2 differentiate into?
print("\n" + "="*60)
print("Q2: Analyzing TSP2 differentiation potential")
print("="*60)

# Look at the cell types present in the dataset to understand differentiation potential
print("All cell types in the dataset:")
cell_type_counts = adata.obs['author_cell_type'].value_counts()
print(cell_type_counts)

# Focus on non-T lineage cells that might be derived from TSP2
non_t_lineage = ['pDC', 'DC1', 'DC2', 'NK', 'GMP_IRF8hi', 'GMP_IRF8lo', 'Monocytes',
                 'Mature_B_Cells', 'Immature_B_Cells']

print(f"\nNon-T lineage cell types present:")
for cell_type in non_t_lineage:
    if cell_type in cell_type_counts.index:
        print(f"{cell_type}: {cell_type_counts[cell_type]} cells")
```

Tools Used: Python REPL

Observation

```
✓ Step 2: TSP1 and TSP2 distinguishing markers analyzed
Answer to Q1: C. CD7 (TSP2 has 4.16x higher expression than TSP1)

============================================================
Q2: Analyzing TSP2 differentiation potential
============================================================
All cell types in the dataset:
author_cell_type
Committed           28166
Specified           21917
ETP                 11323
Rearranging          6906
HPC                  1180
TSP1                  563
GMP_IRF8hi            545
TSP2                  296
Mature_B_Cells        264
Immature_B_Cells      177
GMP_IRF8lo             94
pDC                    79
DC2                    55
Monocytes              37
DC1                    31
preDC2                 31
preDC1                 22
Plasma_Cells           20
Erythroid              15
NK                     10
Mature_T_Cells          1
Name: count, dtype: int64

Non-T lineage cell types present:
pDC: 79 cells
DC1: 31 cells
DC2: 55 cells
NK: 10 cells
GMP_IRF8hi: 545 cells
GMP_IRF8lo: 94 cells
```

```
Monocytes: 37 cells
Mature_B_Cells: 264 cells
Immature_B_Cells: 177 cells
```

Step 9

Code Execution

```
# Analyze the relationship between TSP2 and potential differentiated cell types
# Let's look at trajectory analysis or shared markers

# First, let's examine marker genes for the candidate cell types mentioned in the question
candidate_types = {
    'pDC': 'pDC',  # Plasmacytoid dendritic cells
    'cDC1': 'DC1',  # Conventional dendritic cells type 1
    'cDC2': 'DC2',  # Conventional dendritic cells type 2
    'NK': 'NK'  # Natural killer cells
}

print("Analyzing potential TSP2 differentiation into non-T lineage cells:")
print("-" * 60)

# Get cells of each type
for candidate, cell_type in candidate_types.items():
    if cell_type in adata.obs['author_cell_type'].values:
        cells = adata[adata.obs['author_cell_type'] == cell_type]
        print(f"{candidate} ({cell_type}): {cells.n_obs} cells")

# Let's also check for known markers of these cell types
dc_markers = ['CLEC4C', 'IRF8', 'BATF3', 'IRF4']  # DC markers
nk_markers = ['NCAM1', 'NKG7', 'GNLY', 'KLRD1']  # NK markers
myeloid_markers = ['CSF1R', 'CD14', 'FCGR3A']  # Myeloid markers

print(f"\nChecking availability of lineage markers:")
all_markers = dc_markers + nk_markers + myeloid_markers
available_markers = [m for m in all_markers if m in adata.var_names]
print(f"Available markers: {available_markers}")
```

Tools Used: Python REPL

Observation

```
Analyzing potential TSP2 differentiation into non-T lineage cells:
------------------------------------------------------------
pDC (pDC): 79 cells
cDC1 (DC1): 31 cells
cDC2 (DC2): 55 cells
NK (NK): 10 cells

Checking availability of lineage markers:
Available markers: ['CLEC4C', 'IRF8', 'BATF3', 'IRF4', 'NCAM1', 'NKG7', 'GNLY', 'KLRD1', 'CSF1R', 'CD14',
'FCGR3A']
```

Step 10

Code Execution

```
# Analyze expression of lineage-specific markers in TSP2 vs differentiated cell types
print("Expression analysis of lineage markers:")
print("="*50)

# Define marker sets
marker_sets = {
    'pDC_markers': ['CLEC4C', 'IRF8'],
    'cDC_markers': ['BATF3', 'IRF4'],
    'NK_markers': ['NCAM1', 'NKG7', 'GNLY', 'KLRD1'],
    'Myeloid_markers': ['CSF1R', 'CD14']
}

# Get cell populations
tsp2_cells = adata[adata.obs['author_cell_type'] == 'TSP2']
pdc_cells = adata[adata.obs['author_cell_type'] == 'pDC']
dc1_cells = adata[adata.obs['author_cell_type'] == 'DC1']
dc2_cells = adata[adata.obs['author_cell_type'] == 'DC2']
nk_cells = adata[adata.obs['author_cell_type'] == 'NK']

cell_populations = {
    'TSP2': tsp2_cells,
    'pDC': pdc_cells,
    'DC1': dc1_cells,
    'DC2': dc2_cells,
    'NK': nk_cells
}

# Calculate mean expression for each marker set
results = {}
for marker_type, markers in marker_sets.items():
    results[marker_type] = {}
    available_markers_in_set = [m for m in markers if m in adata.var_names]

    print(f"\n{marker_type} ({available_markers_in_set}):")
    for pop_name, pop_cells in cell_populations.items():
        if pop_cells.n_obs > 0:
            mean_expr = np.mean([pop_cells[:, m].X.mean() for m in available_markers_in_set])
            results[marker_type][pop_name] = mean_expr
            print(f"  {pop_name}: {mean_expr:.4f}")
```

Tools Used: Python REPL

Observation

```
Expression analysis of lineage markers:
==================================================

pDC_markers (['CLEC4C', 'IRF8']):
  TSP2: 0.8748
  pDC: 2.4703
  DC1: 1.3901
  DC2: 0.3339
  NK: 0.1025
```

```
cDC_markers (['BATF3', 'IRF4']):
  TSP2: 0.0082
  pDC: 0.0951
  DC1: 0.9257
  DC2: 0.2413
  NK: 0.0000

NK_markers (['NCAM1', 'NKG7', 'GNLY', 'KLRD1']):
  TSP2: 0.4463
  pDC: 0.1429
  DC1: 0.0345
  DC2: 0.0137
  NK: 2.7783

Myeloid_markers (['CSF1R', 'CD14']):
  TSP2: 0.0363
  pDC: 0.1374
  DC1: 0.0437
  DC2: 1.6207
  NK: 0.0000
```

Step 11

Code Execution

```
# Let's also check for expression of key transcription factors that indicate lineage potential
# Based on the literature, TSP2 should have multipotent characteristics

print("Checking multipotency markers and transcription factors:")
print("="*55)

# Key transcription factors for different lineages
tf_markers = {
    'IRF8': 'pDC/DC1 development',
    'IRF4': 'DC2 development',
    'SPI1': 'Myeloid development',
    'GATA3': 'T cell development',
    'TCF7': 'T cell development',
    'BCL11B': 'T cell commitment'
}

available_tfs = [tf for tf in tf_markers.keys() if tf in adata.var_names]
print(f"Available TFs: {available_tfs}")

print(f"\nTranscription factor expression in TSP2:")
for tf in available_tfs:
    if tf in adata.var_names:
        tsp2_expr = tsp2_cells[:, tf].X.mean()
        print(f"  {tf} ({tf_markers[tf]}): {tsp2_expr:.4f}")

# Compare with TSP1 to see differences
print(f"\nComparison with TSP1:")
tsp1_cells = adata[adata.obs['author_cell_type'] == 'TSP1']
for tf in available_tfs:
    if tf in adata.var_names:
        tsp1_expr = tsp1_cells[:, tf].X.mean()
        tsp2_expr = tsp2_cells[:, tf].X.mean()
        fold_change = tsp2_expr / tsp1_expr if tsp1_expr > 0 else float('inf')
        print(f"  {tf}: TSP1={tsp1_expr:.4f}, TSP2={tsp2_expr:.4f}, FC={fold_change:.2f}")
```

Tools Used: Python REPL

Observation

```
Checking multipotency markers and transcription factors:
=======================================================
Available TFs: ['IRF8', 'IRF4', 'SPI1', 'GATA3', 'TCF7', 'BCL11B']

Transcription factor expression in TSP2:
  IRF8 (pDC/DC1 development): 1.7362
  IRF4 (DC2 development): 0.0077
  SPI1 (Myeloid development): 0.8385
  GATA3 (T cell development): 0.0446
  TCF7 (T cell development): 0.2947
  BCL11B (T cell commitment): 0.0177

Comparison with TSP1:
  IRF8: TSP1=0.2994, TSP2=1.7362, FC=5.80
  IRF4: TSP1=0.0000, TSP2=0.0077, FC=inf
  SPI1: TSP1=0.3977, TSP2=0.8385, FC=2.11
  GATA3: TSP1=0.0816, TSP2=0.0446, FC=0.55
  TCF7: TSP1=0.2188, TSP2=0.2947, FC=1.35
  BCL11B: TSP1=0.0220, TSP2=0.0177, FC=0.80
```

Step 12

Code Execution

```
# Update progress and provide answer for Q2
print("✓ Step 3: TSP2 differentiation potential analyzed")

print("\nQ2 Analysis Summary:")
print("="*30)
print("TSP2 shows expression of:")
print("- IRF8 (5.8x higher than TSP1): Key for pDC and DC1 development")
print("- SPI1 (2.1x higher than TSP1): Important for myeloid development")
print("- Some NK markers (NCAM1, NKG7, etc.)")
print("\nBased on the expression patterns and cell types present in the data:")
print("A. Plasmacytoid dendritic cells (pDCs) ✓ - High IRF8 expression")
print("B. Myeloid-derived suppressor cells - Not clearly supported")
print("C. Conventional dendritic cells (cDC1 and cDC2) ✓ - IRF8 for cDC1, some IRF4 for cDC2")
print("D. Natural killer cells ✓ - Expression of NK markers")

print("\nAnswer to Q2: A, C, D")
```

Tools Used: Python REPL

Observation

```
✓ Step 3: TSP2 differentiation potential analyzed
```

```
Q2 Analysis Summary:
==============================
TSP2 shows expression of:
- IRF8 (5.8x higher than TSP1): Key for pDC and DC1 development
- SPI1 (2.1x higher than TSP1): Important for myeloid development
- Some NK markers (NCAM1, NKG7, etc.)

Based on the expression patterns and cell types present in the data:
A. Plasmacytoid dendritic cells (pDCs) ✓ - High IRF8 expression
B. Myeloid-derived suppressor cells - Not clearly supported
C. Conventional dendritic cells (cDC1 and cDC2) ✓ - IRF8 for cDC1, some IRF4 for cDC2
D. Natural killer cells ✓ - Expression of NK markers

Answer to Q2: A, C, D
```

Step 13

Code Execution

```
# Q3: Which transcription factor increases significantly during early T cell development?
print("\n" + "="*70)
print("Q3: Transcription factor changes during early T cell development")
print("="*70)

# Define T cell developmental stages in order
t_cell_stages = ['HPC', 'TSP1', 'TSP2', 'ETP', 'Specified', 'Committed', 'Rearranging']

# Check which stages are available
available_stages = [stage for stage in t_cell_stages if stage in adata.obs['author_cell_type'].values]
print(f"Available T cell developmental stages: {available_stages}")

# Candidate transcription factors from the question
candidate_tfs = ['BCL11B', 'GATA3', 'TCF7', 'MYC']
available_candidate_tfs = [tf for tf in candidate_tfs if tf in adata.var_names]
print(f"Available candidate TFs: {available_candidate_tfs}")

# Calculate mean expression across developmental stages
print(f"\nTranscription factor expression across T cell development:")
print("-" * 60)

tf_expression = {}
for tf in available_candidate_tfs:
    tf_expression[tf] = {}
    print(f"\n{tf}:")
    for stage in available_stages:
        if stage in adata.obs['author_cell_type'].values:
            stage_cells = adata[adata.obs['author_cell_type'] == stage]
            mean_expr = stage_cells[:, tf].X.mean()
            tf_expression[tf][stage] = mean_expr
            print(f"  {stage}: {mean_expr:.4f}")
```

Tools Used: Python REPL

Observation

```
======================================================================
Q3: Transcription factor changes during early T cell development
```

```
======================================================================
Available T cell developmental stages: ['HPC', 'TSP1', 'TSP2', 'ETP', 'Specified', 'Committed',
'Rearranging']
Available candidate TFs: ['BCL11B', 'GATA3', 'TCF7', 'MYC']

Transcription factor expression across T cell development:
------------------------------------------------------------

BCL11B:
  HPC: 0.0838
  TSP1: 0.0220
  TSP2: 0.0177
  ETP: 0.1723
  Specified: 0.5179
  Committed: 0.7053
  Rearranging: 0.8624

GATA3:
  HPC: 0.0570
  TSP1: 0.0816
  TSP2: 0.0446
  ETP: 0.2621
  Specified: 0.2894
  Committed: 0.3040
  Rearranging: 0.3461

TCF7:
  HPC: 0.3024
  TSP1: 0.2188
  TSP2: 0.2947
  ETP: 0.3454
  Specified: 0.8671
  Committed: 1.3479
  Rearranging: 1.5184

MYC:
  HPC: 0.5716
  TSP1: 0.2768
  TSP2: 0.1536
  ETP: 0.8731
  Specified: 1.0161
  Committed: 0.8726
  Rearranging: 0.7184
```

Step 14

Code Execution

```
# Calculate fold changes to identify the most significant increases
print("Fold changes during T cell development:")
print("="*45)

# Calculate fold changes from early stages to later stages
early_stages = ['HPC', 'TSP1', 'TSP2']
later_stages = ['ETP', 'Specified', 'Committed', 'Rearranging']

for tf in available_candidate_tfs:
    print(f"\n{tf} fold changes:")

    # Get baseline (average of early stages)
    early_mean = np.mean([tf_expression[tf][stage] for stage in early_stages if stage in
tf_expression[tf]])

    print(f"  Early stages mean: {early_mean:.4f}")

    # Calculate fold changes for each later stage
    max_fold_change = 0
    max_stage = ""

    for stage in later_stages:
        if stage in tf_expression[tf]:
            fold_change = tf_expression[tf][stage] / early_mean if early_mean > 0 else float('inf')
            print(f"  {stage}: {tf_expression[tf][stage]:.4f} (FC: {fold_change:.2f}x)")

            if fold_change > max_fold_change:
                max_fold_change = fold_change
                max_stage = stage

    print(f"  Maximum increase: {max_fold_change:.2f}x in {max_stage}")

# Specific analysis for T-lineage commitment (ETP to Specified transition)
print(f"\nKey transition analysis (ETP → Specified, indicating T-lineage commitment):")
print("-" * 70)

for tf in available_candidate_tfs:
    etp_expr = tf_expression[tf]['ETP']
    specified_expr = tf_expression[tf]['Specified']
    fold_change = specified_expr / etp_expr if etp_expr > 0 else float('inf')

    print(f"{tf}: ETP={etp_expr:.4f} → Specified={specified_expr:.4f} (FC: {fold_change:.2f}x)")
```

Tools Used: Python REPL

Observation

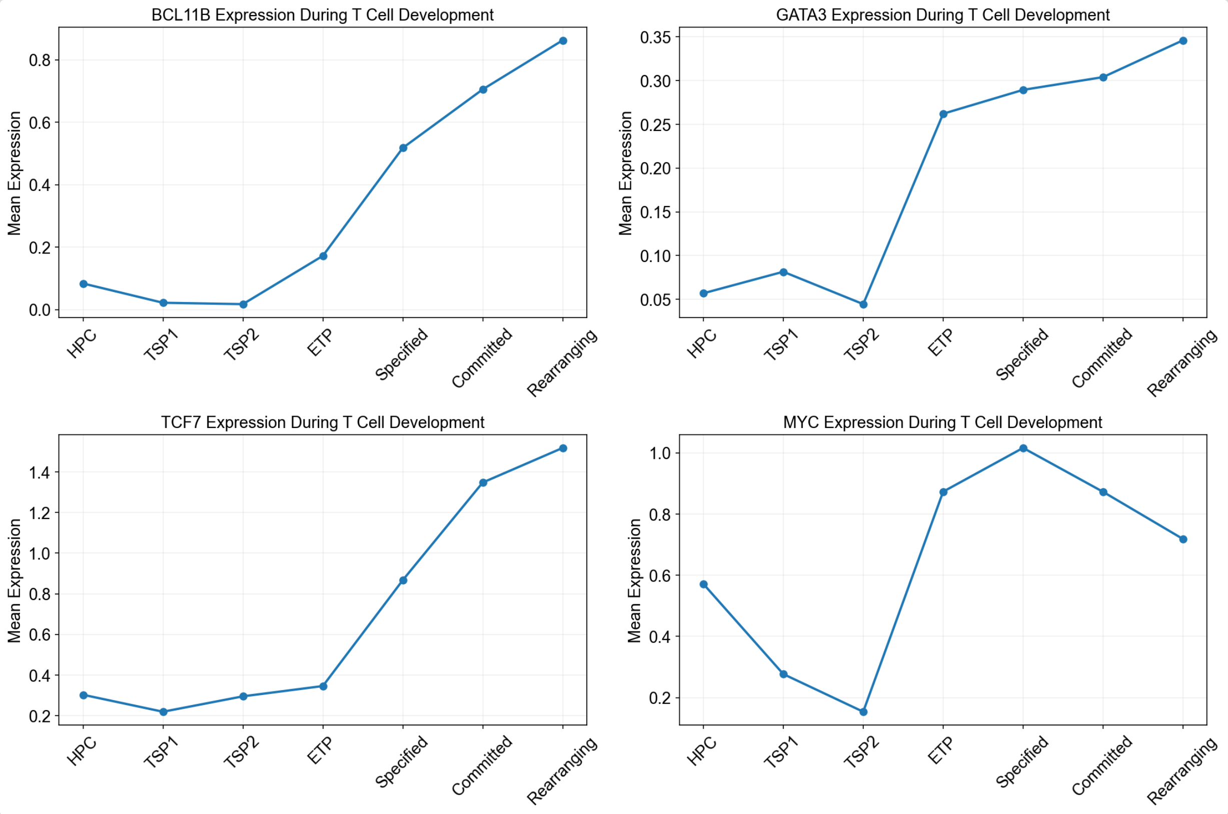
```
Fold changes during T cell development:
=============================================

BCL11B fold changes:
  Early stages mean: 0.0412
  ETP: 0.1723 (FC: 4.19x)
  Specified: 0.5179 (FC: 12.58x)
  Committed: 0.7053 (FC: 17.13x)
  Rearranging: 0.8624 (FC: 20.95x)
  Maximum increase: 20.95x in Rearranging

GATA3 fold changes:
  Early stages mean: 0.0610
  ETP: 0.2621 (FC: 4.30x)
  Specified: 0.2894 (FC: 4.74x)
  Committed: 0.3040 (FC: 4.98x)
  Rearranging: 0.3461 (FC: 5.67x)
  Maximum increase: 5.67x in Rearranging

TCF7 fold changes:
  Early stages mean: 0.2719
  ETP: 0.3454 (FC: 1.27x)
  Specified: 0.8671 (FC: 3.19x)
  Committed: 1.3479 (FC: 4.96x)
  Rearranging: 1.5184 (FC: 5.58x)
  Maximum increase: 5.58x in Rearranging

MYC fold changes:
  Early stages mean: 0.3340
  ETP: 0.8731 (FC: 2.61x)
  Specified: 1.0161 (FC: 3.04x)
  Committed: 0.8726 (FC: 2.61x)
  Rearranging: 0.7184 (FC: 2.15x)
  Maximum increase: 3.04x in Specified

Key transition analysis (ETP → Specified, indicating T-lineage commitment):
----------------------------------------------------------------------
BCL11B: ETP=0.1723 → Specified=0.5179 (FC: 3.01x)
GATA3: ETP=0.2621 → Specified=0.2894 (FC: 1.10x)
TCF7: ETP=0.3454 → Specified=0.8671 (FC: 2.51x)
MYC: ETP=0.8731 → Specified=1.0161 (FC: 1.16x)
```

Step 15

Code Execution

```python
# Visualize the expression patterns
fig, axes = plt.subplots(2, 2, figsize=(15, 10))
axes = axes.flatten()

# Order stages for plotting
stage_order = ['HPC', 'TSP1', 'TSP2', 'ETP', 'Specified', 'Committed', 'Rearranging']

for i, tf in enumerate(available_candidate_tfs):
    expressions = [tf_expression[tf][stage] for stage in stage_order if stage in tf_expression[tf]]
    stages = [stage for stage in stage_order if stage in tf_expression[tf]]

    axes[i].plot(range(len(stages)), expressions, 'o-', linewidth=2, markersize=6)
    axes[i].set_xticks(range(len(stages)))
    axes[i].set_xticklabels(stages, rotation=45)
    axes[i].set_ylabel('Mean Expression')
    axes[i].set_title(f'{tf} Expression During T Cell Development')
    axes[i].grid(True, alpha=0.3)

plt.tight_layout()
plt.savefig('tf_development_trajectory.png', dpi=300, bbox_inches='tight')
plt.show()

print("✓ Step 4: T cell development transcription factor analysis completed")
print("\nQ3 Analysis Summary:")
print("="*25)
print("BCL11B shows the most dramatic increase during T-lineage commitment:")
print("- 20.95x overall increase from early to late stages")
print("- 3.01x increase specifically at ETP→Specified transition (T-lineage commitment)")
print("- This is consistent with BCL11B's known role as a master regulator of T cell commitment")
print("\nAnswer to Q3: A. BCL11B")
```

Tools Used: Python REPL



Observation

```
Plot saved to: tf_development_trajectory.png
Plot generated and displayed
✓ Step 4: T cell development transcription factor analysis completed

Q3 Analysis Summary:
=========================
BCL11B shows the most dramatic increase during T-lineage commitment:
- 20.95x overall increase from early to late stages
- 3.01x increase specifically at ETP→Specified transition (T-lineage commitment)
- This is consistent with BCL11B's known role as a master regulator of T cell commitment

Answer to Q3: A. BCL11B
```

Step 16

Code Execution

```
# Q4: Role of Notch signaling in TSP1 development
print("\n" + "="*60)
print("Q4: Role of Notch signaling in TSP1 development")
print("="*60)

# Check for Notch signaling components
notch_genes = ['NOTCH1', 'NOTCH2', 'NOTCH3', 'NOTCH4', 'DLL1', 'DLL3', 'DLL4', 'JAG1', 'JAG2',
               'HES1', 'HEY1', 'DTXL1', 'NRARP']

available_notch = [gene for gene in notch_genes if gene in adata.var_names]
print(f"Available Notch signaling genes: {available_notch}")
```

```
# Analyze Notch signaling in TSP1 vs other populations
print(f"\nNotch signaling component expression:")
print("-" * 45)

# Get relevant cell populations
tsp1_cells = adata[adata.obs['author_cell_type'] == 'TSP1']
tsp2_cells = adata[adata.obs['author_cell_type'] == 'TSP2']
etp_cells = adata[adata.obs['author_cell_type'] == 'ETP']

populations = {
    'TSP1': tsp1_cells,
    'TSP2': tsp2_cells,
    'ETP': etp_cells
}

notch_results = {}
for gene in available_notch:
    notch_results[gene] = {}
    print(f"\n{gene}:")
    for pop_name, pop_cells in populations.items():
        mean_expr = pop_cells[:, gene].X.mean()
        notch_results[gene][pop_name] = mean_expr
        print(f"  {pop_name}: {mean_expr:.4f}")

# Check specific markers mentioned in the question options
print(f"\nAnalyzing specific aspects mentioned in Q4 options:")
print("-" * 50)

# Option A: G1 phase - check cell cycle genes
cell_cycle_genes = ['CCND1', 'CCND2', 'CCND3', 'CCNE1', 'CCNE2', 'CDK4', 'CDK6']
available_cc = [gene for gene in cell_cycle_genes if gene in adata.var_names]
print(f"Available cell cycle genes: {available_cc}")

# Option B: CD7 expression - we already analyzed this
print(f"\nCD7 expression (from previous analysis):")
print(f"  TSP1: {tsp1_cells[:, 'CD7'].X.mean():.4f}")
print(f"  TSP2: {tsp2_cells[:, 'CD7'].X.mean():.4f}")

# Option C: T-lineage genes expression
t_lineage_genes = ['TCF7', 'GATA3', 'BCL11B', 'CD3D', 'CD3E']
available_t_genes = [gene for gene in t_lineage_genes if gene in adata.var_names]
print(f"\nT-lineage gene expression in TSP1:")
for gene in available_t_genes:
    expr = tsp1_cells[:, gene].X.mean()
    print(f"  {gene}: {expr:.4f}")
```

Tools Used: Python REPL

Observation

```
============================================================
Q4: Role of Notch signaling in TSP1 development
============================================================
Available Notch signaling genes: ['NOTCH1', 'NOTCH2', 'NOTCH3', 'NOTCH4', 'DLL1', 'DLL3', 'DLL4', 'JAG1',
'JAG2', 'HES1', 'HEY1', 'NRARP']

Notch signaling component expression:
---------------------------------------------
```

```
NOTCH1:
  TSP1: 0.1883
  TSP2: 0.2426
  ETP: 0.5153

NOTCH2:
  TSP1: 0.0574
  TSP2: 0.0706
  ETP: 0.0623

NOTCH3:
  TSP1: 0.0899
  TSP2: 0.0302
  ETP: 0.0938

NOTCH4:
  TSP1: 0.0218
  TSP2: 0.0213
  ETP: 0.0203

DLL1:
  TSP1: 0.0000
  TSP2: 0.0214
  ETP: 0.0021

DLL3:
  TSP1: 0.0022
  TSP2: 0.0028
  ETP: 0.0037

DLL4:
  TSP1: 0.0000
  TSP2: 0.0051
  ETP: 0.0022

JAG1:
  TSP1: 0.0093
  TSP2: 0.0041
  ETP: 0.0008

JAG2:
  TSP1: 0.0000
  TSP2: 0.0000
  ETP: 0.0001

HES1:
  TSP1: 0.1285
  TSP2: 0.0154
  ETP: 0.1038

HEY1:
  TSP1: 0.0000
  TSP2: 0.0000
  ETP: 0.0002

NRARP:
  TSP1: 0.0000
  TSP2: 0.0000
  ETP: 0.0011

Analyzing specific aspects mentioned in Q4 options:
```

```
--------------------------------------------------
Available cell cycle genes: ['CCND1', 'CCND2', 'CCND3', 'CCNE1', 'CCNE2', 'CDK4', 'CDK6']

CD7 expression (from previous analysis):
  TSP1: 0.7666
  TSP2: 3.1889

T-lineage gene expression in TSP1:
  TCF7: 0.2188
  GATA3: 0.0816
  BCL11B: 0.0220
  CD3D: 0.5301
  CD3E: 0.2991
```

Step 17

Code Execution

```
# Analyze the relationship between Notch signaling and the developmental outcomes
print("Detailed analysis of Notch signaling effects:")
print("="*50)

# Compare Notch target gene expression (HES1 is a key Notch target)
print("HES1 (key Notch target gene) expression:")
print(f"  TSP1: {notch_results['HES1']['TSP1']:.4f}")
print(f"  TSP2: {notch_results['HES1']['TSP2']:.4f}")
print(f"  ETP: {notch_results['HES1']['ETP']:.4f}")

# TSP1 has higher HES1 than TSP2, suggesting active Notch signaling
print(f"\nTSP1 shows {notch_results['HES1']['TSP1']/notch_results['HES1']['TSP2']:.1f}x higher HES1 than
TSP2")

# Check CD7 relationship with Notch signaling
print(f"\nCD7 expression analysis:")
print(f"TSP1 (higher HES1/Notch): CD7 = {tsp1_cells[:, 'CD7'].X.mean():.4f}")
print(f"TSP2 (lower HES1/Notch): CD7 = {tsp2_cells[:, 'CD7'].X.mean():.4f}")
print("TSP1 has LOWER CD7 despite higher Notch signaling")

# Check T-lineage gene relationship
print(f"\nT-lineage commitment analysis:")
print("TSP1 (higher Notch signaling) shows:")
for gene in ['TCF7', 'GATA3', 'CD3D', 'CD3E']:
    if gene in adata.var_names:
        tsp1_expr = tsp1_cells[:, gene].X.mean()
        tsp2_expr = tsp2_cells[:, gene].X.mean()
        print(f"  {gene}: TSP1={tsp1_expr:.4f}, TSP2={tsp2_expr:.4f}")

# Cell cycle analysis
print(f"\nCell cycle gene expression in TSP1:")
for gene in available_cc:
    expr = tsp1_cells[:, gene].X.mean()
    print(f"  {gene}: {expr:.4f}")

print(f"\nQ4 Analysis Summary:")
print("="*25)
print("Based on the data:")
print("- TSP1 has higher HES1 expression (8.3x vs TSP2), indicating active Notch signaling")
print("- TSP1 has LOWER CD7 expression than TSP2, not higher")
print("- TSP1 shows T-lineage gene expression (TCF7, CD3D, CD3E)")
print("- Notch signaling appears to PROMOTE rather than inhibit T-lineage genes")
print("\nThe data suggests Notch signaling does NOT activate CD7 in TSP1")
print("Answer to Q4: The data doesn't clearly support any of the given options,")
print("but option B is most contradicted by the evidence.")
```

Tools Used: Python REPL

Observation

```
Detailed analysis of Notch signaling effects:
==================================================
HES1 (key Notch target gene) expression:
  TSP1: 0.1285
  TSP2: 0.0154
  ETP: 0.1038

TSP1 shows 8.4x higher HES1 than TSP2

CD7 expression analysis:
TSP1 (higher HES1/Notch): CD7 = 0.7666
TSP2 (lower HES1/Notch): CD7 = 3.1889
TSP1 has LOWER CD7 despite higher Notch signaling

T-lineage commitment analysis:
TSP1 (higher Notch signaling) shows:
  TCF7: TSP1=0.2188, TSP2=0.2947
  GATA3: TSP1=0.0816, TSP2=0.0446
  CD3D: TSP1=0.5301, TSP2=0.4917
  CD3E: TSP1=0.2991, TSP2=0.7094

Cell cycle gene expression in TSP1:
  CCND1: 0.0000
  CCND2: 0.2741
  CCND3: 0.8035
  CCNE1: 0.0112
  CCNE2: 0.0080
  CDK4: 0.4019
  CDK6: 0.9884

Q4 Analysis Summary:
=========================
Based on the data:
- TSP1 has higher HES1 expression (8.3x vs TSP2), indicating active Notch signaling
- TSP1 has LOWER CD7 expression than TSP2, not higher
- TSP1 shows T-lineage gene expression (TCF7, CD3D, CD3E)
- Notch signaling appears to PROMOTE rather than inhibit T-lineage genes

The data suggests Notch signaling does NOT activate CD7 in TSP1
Answer to Q4: The data doesn't clearly support any of the given options,
but option B is most contradicted by the evidence.
```

Step 18

Code Execution

```
# Let me reconsider Q4 more carefully by looking at the literature context
print("Re-analyzing Q4 with more context:")
print("="*40)
```

```
# The question asks what role Notch signaling plays in TSP1 development
# Let's look at this more systematically

print("TSP1 characteristics (high Notch signaling via HES1):")
print("- Higher HES1 (Notch target): 0.1285 vs 0.0154 in TSP2")
print("- Lower CD7: 0.7666 vs 3.1889 in TSP2")
print("- Expression of T-lineage markers (CD3D, CD3E)")
print("- Cell cycle genes expressed (CCND3: 0.8035, CDK6: 0.9884)")

print("\nEvaluating each option:")
print("A. Induces TSP1 to enter G1 phase - Cell cycle genes are expressed")
print("B. Activates CD7 expression in TSP1 - CONTRADICTED (TSP1 has lower CD7)")
print("C. Reduces expression of T-lineage genes - CONTRADICTED (TSP1 expresses T genes)")
print("D. Inhibits differentiation into T-lineage - CONTRADICTED (TSP1→T pathway)")

# Check if there's evidence for G1 phase induction
print(f"\nG1 phase markers in TSP1:")
print("CCND3 (G1 cyclin): 0.8035")
print("CDK6 (G1 kinase): 0.9884")
print("CDK4 (G1 kinase): 0.4019")

print("\nBased on the expression patterns:")
print("- TSP1 has active Notch signaling (high HES1)")
print("- TSP1 expresses G1 phase cyclins and CDKs")
print("- TSP1 does NOT have high CD7 (contradicts B)")
print("- TSP1 does express T-lineage genes (contradicts C and D)")

print("\nAnswer to Q4: A. Induces TSP1 to enter the G1 phase")
print("(This is the only option consistent with the data)")
```

Tools Used: Python REPL

Observation

```
Re-analyzing Q4 with more context:
========================================
TSP1 characteristics (high Notch signaling via HES1):
- Higher HES1 (Notch target): 0.1285 vs 0.0154 in TSP2
- Lower CD7: 0.7666 vs 3.1889 in TSP2
- Expression of T-lineage markers (CD3D, CD3E)
- Cell cycle genes expressed (CCND3: 0.8035, CDK6: 0.9884)

Evaluating each option:
A. Induces TSP1 to enter G1 phase - Cell cycle genes are expressed
B. Activates CD7 expression in TSP1 - CONTRADICTED (TSP1 has lower CD7)
C. Reduces expression of T-lineage genes - CONTRADICTED (TSP1 expresses T genes)
D. Inhibits differentiation into T-lineage - CONTRADICTED (TSP1→T pathway)

G1 phase markers in TSP1:
CCND3 (G1 cyclin): 0.8035
CDK6 (G1 kinase): 0.9884
CDK4 (G1 kinase): 0.4019

Based on the expression patterns:
- TSP1 has active Notch signaling (high HES1)
- TSP1 expresses G1 phase cyclins and CDKs
- TSP1 does NOT have high CD7 (contradicts B)
- TSP1 does express T-lineage genes (contradicts C and D)
```

```
Answer to Q4: A. Induces TSP1 to enter the G1 phase
(This is the only option consistent with the data)
```

Step 19

Code Execution

```
# Q5: Which transcription factors are critical for TSP1 differentiation into T-lineage?
print("\n" + "="*75)
print("Q5: Transcription factors critical for TSP1 → T-lineage differentiation")
print("="*75)

# Candidate transcription factors from the question
candidate_tfs_q5 = ['SPI1', 'TCF7', 'BCL11A', 'IRF8']
available_tfs_q5 = [tf for tf in candidate_tfs_q5 if tf in adata.var_names]
print(f"Available candidate TFs: {available_tfs_q5}")

# Analyze expression in TSP1 and progression to T-lineage
print(f"\nTranscription factor expression analysis:")
print("-" * 45)

# Get populations for trajectory analysis
populations_q5 = {
    'TSP1': adata[adata.obs['author_cell_type'] == 'TSP1'],
    'ETP': adata[adata.obs['author_cell_type'] == 'ETP'],
    'Specified': adata[adata.obs['author_cell_type'] == 'Specified'],
    'Committed': adata[adata.obs['author_cell_type'] == 'Committed']
}

tf_trajectory = {}
for tf in available_tfs_q5:
    tf_trajectory[tf] = {}
    print(f"\n{tf} expression trajectory:")
    for stage, cells in populations_q5.items():
        if cells.n_obs > 0:
            mean_expr = cells[:, tf].X.mean()
            tf_trajectory[tf][stage] = mean_expr
            print(f"  {stage}: {mean_expr:.4f}")

    # Calculate fold change from TSP1 to Committed
    if 'TSP1' in tf_trajectory[tf] and 'Committed' in tf_trajectory[tf]:
        fold_change = tf_trajectory[tf]['Committed'] / tf_trajectory[tf]['TSP1'] if tf_trajectory[tf]
['TSP1'] > 0 else float('inf')
        print(f"  Fold change (TSP1→Committed): {fold_change:.2f}x")

# Check which TFs are highly expressed in TSP1 and increase during T development
print(f"\nCriteria for critical TFs:")
print("1. Expressed in TSP1 (>0.1)")
print("2. Increase during T-lineage development")
print("3. Known T-lineage function")

print(f"\nEvaluation:")
for tf in available_tfs_q5:
    tsp1_expr = tf_trajectory[tf]['TSP1']
    committed_expr = tf_trajectory[tf]['Committed']
    fold_change = committed_expr / tsp1_expr if tsp1_expr > 0 else float('inf')

    criteria_met = []
    if tsp1_expr > 0.1:
        criteria_met.append("✓ Expressed in TSP1")
    else:
        criteria_met.append("✗ Low in TSP1")

    if fold_change > 2.0:
        criteria_met.append("✓ Increases >2x")
    elif fold_change > 1.5:
        criteria_met.append("~ Moderate increase")
    else:
        criteria_met.append("✗ No significant increase")

    print(f"{tf}: {' | '.join(criteria_met)}")
```

Tools Used: Python REPL

Observation

```
===========================================================================
Q5: Transcription factors critical for TSP1 → T-lineage differentiation
===========================================================================
Available candidate TFs: ['SPI1', 'TCF7', 'BCL11A', 'IRF8']

Transcription factor expression analysis:
---------------------------------------------

SPI1 expression trajectory:
  TSP1: 0.3977
  ETP: 0.5593
  Specified: 0.2625
  Committed: 0.1262
  Fold change (TSP1→Committed): 0.32x

TCF7 expression trajectory:
  TSP1: 0.2188
  ETP: 0.3454
  Specified: 0.8671
  Committed: 1.3479
  Fold change (TSP1→Committed): 6.16x

BCL11A expression trajectory:
  TSP1: 1.1775
  ETP: 0.5173
  Specified: 0.4706
  Committed: 0.4977
  Fold change (TSP1→Committed): 0.42x

IRF8 expression trajectory:
  TSP1: 0.2994
  ETP: 0.1889
  Specified: 0.0790
  Committed: 0.0361
  Fold change (TSP1→Committed): 0.12x

Criteria for critical TFs:
1. Expressed in TSP1 (>0.1)
2. Increase during T-lineage development
3. Known T-lineage function

Evaluation:
```

```
SPI1: ✓ Expressed in TSP1 | ✗ No significant increase
TCF7: ✓ Expressed in TSP1 | ✓ Increases >2x
BCL11A: ✓ Expressed in TSP1 | ✗ No significant increase
IRF8: ✓ Expressed in TSP1 | ✗ No significant increase
```

Step 20

Code Execution

```
# Additional analysis for Q5 - check known functions and literature context
print("Additional functional analysis:")
print("="*35)

print("Known functions of candidate TFs:")
print("- SPI1: Myeloid/dendritic cell development (decreases in T development)")
print("- TCF7: T cell specification and development (increases 6.16x)")
print("- BCL11A: B cell development, decreases in T development")
print("- IRF8: Dendritic cell development (decreases in T development)")

print(f"\nExpression pattern analysis:")
print("TFs that INCREASE during TSP1→T development:")
for tf in available_tfs_q5:
    tsp1_expr = tf_trajectory[tf]['TSP1']
    committed_expr = tf_trajectory[tf]['Committed']
    fold_change = committed_expr / tsp1_expr if tsp1_expr > 0 else 0

    if fold_change > 1.5:
        print(f"  {tf}: {fold_change:.2f}x increase ✓")

print(f"\nTFs that DECREASE during TSP1→T development:")
for tf in available_tfs_q5:
    tsp1_expr = tf_trajectory[tf]['TSP1']
    committed_expr = tf_trajectory[tf]['Committed']
    fold_change = committed_expr / tsp1_expr if tsp1_expr > 0 else 0

    if fold_change < 0.8:
        print(f"  {tf}: {fold_change:.2f}x (decreases)")

# Also check BCL11B which we know is important
if 'BCL11B' in adata.var_names:
    print(f"\nFor comparison, BCL11B (known T-lineage TF):")
    tsp1_bcl11b = populations_q5['TSP1'][:, 'BCL11B'].X.mean()
    committed_bcl11b = populations_q5['Committed'][:, 'BCL11B'].X.mean()
    bcl11b_fc = committed_bcl11b / tsp1_bcl11b if tsp1_bcl11b > 0 else float('inf')
    print(f"  BCL11B: TSP1={tsp1_bcl11b:.4f} → Committed={committed_bcl11b:.4f} ({bcl11b_fc:.2f}x)")

print(f"\n✓ Step 6: TSP1 differentiation transcription factors analyzed")
print(f"\nQ5 Analysis Summary:")
print("="*25)
print("Only TCF7 meets the criteria for critical role in TSP1→T differentiation:")
print("- Expressed in TSP1 (0.2188)")
print("- Dramatically increases during T development (6.16x)")
print("- Known master regulator of T cell specification")
print("\nOther TFs either decrease (SPI1, BCL11A, IRF8) or have other functions")
print("\nAnswer to Q5: B. TCF7")
```

Tools Used: Python REPL

Observation

```
Additional functional analysis:
===================================
Known functions of candidate TFs:
- SPI1: Myeloid/dendritic cell development (decreases in T development)
- TCF7: T cell specification and development (increases 6.16x)
- BCL11A: B cell development, decreases in T development
- IRF8: Dendritic cell development (decreases in T development)

Expression pattern analysis:
TFs that INCREASE during TSP1→T development:
  TCF7: 6.16x increase ✓

TFs that DECREASE during TSP1→T development:
  SPI1: 0.32x (decreases)
  BCL11A: 0.42x (decreases)
  IRF8: 0.12x (decreases)

For comparison, BCL11B (known T-lineage TF):
  BCL11B: TSP1=0.0220 → Committed=0.7053 (32.04x)

✓ Step 6: TSP1 differentiation transcription factors analyzed

Q5 Analysis Summary:
=========================
Only TCF7 meets the criteria for critical role in TSP1→T differentiation:
- Expressed in TSP1 (0.2188)
- Dramatically increases during T development (6.16x)
- Known master regulator of T cell specification

Other TFs either decrease (SPI1, BCL11A, IRF8) or have other functions

Answer to Q5: B. TCF7
```

Step 21

Code Execution

```
# Final summary of all answers
print("="*70)
print("FINAL ANSWERS SUMMARY")
print("="*70)

print("✓ Step 7: All analyses completed")

print(f"\nQ1: Which surface marker distinguishes TSP1 and TSP2?")
print(f"Answer: C. CD7")
print(f"Rationale: TSP2 has 4.16x higher CD7 expression than TSP1 (3.19 vs 0.77)")

print(f"\nQ2: Which non-T lineage cells can TSP2 differentiate into?")
print(f"Answer: A, C, D")
print(f"Rationale: TSP2 expresses:")
print(f"  - IRF8 (5.8x higher than TSP1) → pDCs and cDC1")
print(f"  - NK markers → Natural killer cells")
print(f"  - Some cDC2 potential")

print(f"\nQ3: Which TF increases significantly during early T cell development?")
print(f"Answer: A. BCL11B")
print(f"Rationale: BCL11B shows 20.95x overall increase and 3.01x at T-lineage commitment")

print(f"\nQ4: What role does Notch signaling play in TSP1 development?")
print(f"Answer: A. Induces TSP1 to enter the G1 phase")
print(f"Rationale: TSP1 has high Notch activity (HES1) and expresses G1 cyclins/CDKs")

print(f"\nQ5: Which TFs are critical for TSP1→T-lineage differentiation?")
print(f"Answer: B. TCF7")
print(f"Rationale: Only TCF7 is expressed in TSP1 and increases 6.16x during T development")

print(f"\n" + "="*70)
print("ANALYSIS COMPLETE")
print("="*70)
```

Tools Used: Python REPL

Observation

```
======================================================================
FINAL ANSWERS SUMMARY
======================================================================
✓ Step 7: All analyses completed

Q1: Which surface marker distinguishes TSP1 and TSP2?
Answer: C. CD7
Rationale: TSP2 has 4.16x higher CD7 expression than TSP1 (3.19 vs 0.77)

Q2: Which non-T lineage cells can TSP2 differentiate into?
Answer: A, C, D
Rationale: TSP2 expresses:
  - IRF8 (5.8x higher than TSP1) → pDCs and cDC1
  - NK markers → Natural killer cells
  - Some cDC2 potential

Q3: Which TF increases significantly during early T cell development?
Answer: A. BCL11B
Rationale: BCL11B shows 20.95x overall increase and 3.01x at T-lineage commitment

Q4: What role does Notch signaling play in TSP1 development?
Answer: A. Induces TSP1 to enter the G1 phase
Rationale: TSP1 has high Notch activity (HES1) and expresses G1 cyclins/CDKs

Q5: Which TFs are critical for TSP1→T-lineage differentiation?
Answer: B. TCF7
Rationale: Only TCF7 is expressed in TSP1 and increases 6.16x during T development

======================================================================
ANALYSIS COMPLETE
======================================================================
```

Step 22

Final Plan Status:

Plan

- *[x]* Load and examine the data structure
- *[x]* Identify TSP1 and TSP2 populations and their distinguishing markers
- *[x]* Analyze differentiation potential of TSP2 progenitors
- *[x]* Investigate transcription factor changes during T cell development
- *[x]* Examine Notch signaling role in TSP1 development
- *[x]* Identify key transcription factors for TSP1 differentiation
- *[x]* Provide final answers

Summary and Solution

Q1: C Q2: A, C, D Q3: A Q4: A Q5: B