

REFERENCE-FREE CELL-TYPE ANNOTATION WITH LLM AGENTS

Yidi Huang^{1,2}, Ivan Cohen³, Van Q. Truong^{1,2}, Pedram B. Bayat³, Sameer A. Bhatti⁴, Luca Paruzzo³, Mark M. Painter²

¹Genomics and Computational Biology Graduate Group;

²Institute for Immunology and Immune Health;

³Center for Cellular Immunotherapies;

⁴Department of Computer and Information Sciences
University of Pennsylvania
Philadelphia, PA, USA

Shirong Zheng⁵

⁵Department of Biochemistry and Molecular Genetics
University of Louisville
Louisville, KY, USA

Derek A. Oldridge⁶, Joost Wagenaar^{2,7}, Allison R. Greenplate², Dokyoon Kim^{2,7*}

⁶Department of Pathology and Laboratory Medicine;

⁷Department of Biostatistics, Epidemiology and Informatics
University of Pennsylvania
Philadelphia, PA, USA

ABSTRACT

Agentic AI research assistants, enabled by augmenting large language models with code-execution and tool-use abilities, promise to transform scientific workflows and accelerate biomedical research. In this study, we share preliminary results from our work in evaluating LLM agent capabilities in genomics. We design a simple bioinformatic research agent augmented with tool calls and code execution and instructed with a high-level task-agnostic system prompt. We implement this agent with three frontier-level LLMs: GPT-4o, o3-mini, and Claude 3.5 Sonnet, and compare their performance. We evaluate the performance of our agents in labeling cell types in clustered high-resolution transcriptomic data, a traditionally time-intensive task requiring both manual effort and domain expertise. Our agents are able to accurately complete this task, although performance fluctuates over multiple iterations due to hallucination. Overall, our results indicate that LLM agents are capable of autonomously planning and executing genomic analyses with only high-level direction. We are encouraged by these early results and look forward to extending these evaluations in future work.

1 INTRODUCTION

A longstanding goal of artificial intelligence (AI) research is to develop systems that accelerate science. Although large language models (LLMs) have demonstrated impressive abilities in knowledge recall and factual reasoning, these tasks only constitute a portion of the scientific workflow. The scientific process involves iteratively generating hypotheses from prior knowledge, planning and executing experiments, and interpreting experimental results to synthesize new knowledge. LLMs do not innately have the ability to plan and execute actions or observe the world. These behaviors must be elicited by augmenting LLMs with tool use and code execution. The resulting system has been described in the literature as an *AI agent*.

*Correspondence to: dokyoon.kim@penmedicine.upenn.edu

There is a great deal of excitement around the application of AI agents to biomedical research. Agentic systems have been proposed to autonomously conduct literature reviews (Li et al., 2025), perform bioinformatic analyses (Zhou et al., 2024), and even manage wet lab experiments through robotic laboratory automation (Bran et al., 2024). Such systems promise to accelerate the pace of scientific discoveries and augment scientist productivity (Gao et al., 2024). In particular, the field of genomics represents a viable testing ground for the development of agentic methods due to its data richness, computational workflows, and ever-expanding literature base requiring manual curation.

We direct our attention toward high-resolution transcriptomics, including single-cell RNA-seq and spatial modalities. Compared to traditional bulk RNA-seq, high-resolution transcriptomics enables understanding of cellular heterogeneity by capturing gene expression profiles of individual cells within complex tissues. While computational approaches can identify clusters of cells based on transcriptional similarities, the crucial step of assigning cell-type identities to each cluster remains a significant analytical bottleneck. A canonical workflow for cell-type annotation involves first constructing expression signatures based on overexpressed genes and then manually cross-referencing these signatures with various sources of knowledge. This manual curation step is often slow and laborious. Since relevant cell types vary depending on tissue type and biological context, there is no universal reference atlas of cell-type marker genes. Scientists must identify cell types based on the specific context of each study and curate marker gene sets through domain expertise and extensive literature review. Any unmapped clusters need to be manually disambiguated as a heterogeneous cluster or an unanticipated cell type.

Here we present a case study of a general-purpose bioinformatics LLM agent applied toward cell-type annotation of high-resolution transcriptomics data. Briefly, our agents are given a clustered Visium HD dataset and asked to determine the cell type represented by each cluster. After the initial user prompt, our agents act autonomously to plan a sensible analysis, execute it with python code, and interpret the results of their analysis. We find that agents built with Claude 3.5 Sonnet and o3-mini are able to consistently and accurately complete this objective. We emphasize that, aside from the goal specification, our agents are not specifically instructed in how to perform cell-type annotation, nor are the provided tools specific to transcriptomics, suggesting that LLMs may in fact be able to autonomously plan and execute a breadth of genomic analyses.

2 RELATED WORK

Prior work has established the application of agents toward biomedical research. Some proposed applications include iterative design of CRISPR screens (Roohani et al., 2024), an interactive assistant for CRISPR-based experiments (Huang et al., 2024), and autonomous discovery of SARS-CoV-2 nanobody binders (Swanson et al., 2024). The closest related works to ours are the agents designed to perform bioinformatic analyses. AutoBA (Zhou et al., 2024) features a single-agent design with the ability to install common and execute common bioinformatics libraries. BioMaster (Su et al., 2025) and BIA (Xin et al., 2024) are both multi-agent approaches augmented with a retrieval-augmented generation (RAG) capabilities and a suite of bioinformatic tools. Relative to these, our approach is more goal-oriented than task-oriented. Rather than specifying a particular bioinformatic analysis (e.g. “compute differential gene expression”), we specify a high-level goal (i.e. “Identify the cell types for each cluster”) and leave the agent to plan a suitable approach.

3 METHODS

3.1 AGENT DESIGN

We implement a simple single-agent design using the ag2 library (Wu et al., 2023). Our agent consists of a single backbone LLM which is instructed with a task-agnostic system prompt and augmented with tool use and code execution capabilities. We implement four simple tools and make them available to our agent via function calling (Schick et al., 2023): `get_pubmed_abstracts(query)`, which retrieves matching abstracts from pubmed; `esearch(db, term)`, `efetch(db, id)`, and `esummary(db, id)`, which are thin wrappers over their namesake functions from NCBI Entrez EUtils (Maglott et al., 2006). For code execution, the model is instructed to write python code in markdown blocks, which is executed locally

with the output returned in response. In addition to the tool-use and coding instructions, the system prompt provides a high-level task-agnostic workflow: first explore the dataset, then formulate and execute a plan, and finally conclude by summarizing findings and outputting ‘TERMINATE’. The full system prompt is shown in appendix A. The agent is designed to execute the plan autonomously, with no user interaction required after the initial goal specification, which is provided in the opening user message. After this, the system may either respond with the output of a tool call or code snippet, or an empty user message if neither was included in the assistant message. The session automatically concludes when the assistant message contains either ‘TERMINATE’ or a stop token. We implement the agent with three frontier-level models as base model: Claude 3.5 sonnet, o3-mini with high reasoning level, and GPT-4o. The same prompts are used with each model.

3.2 EVALUATION

We task our models to label cell types in clustered high-resolution transcriptomics data. Our evaluation uses three Visium HD spatial transcriptomics datasets from 10x Genomics originating from healthy mouse kidney, healthy mouse brain, and human tonsil with reactive follicular hyperplasia, respectively. For each dataset, we prepare an input file using the 10x Genomics Loupe browser (v 8.1.2) consisting of a k-means clustering (k=10) of the raw expression data with rows consisting of gene symbols and columns describing differential expression in each cluster. This input file is presented to the agent as part of the following user prompt:

I just finished clustering from my Visium HD Spatial Gene Expression Library experiment on a sample of [tissue source]. The data are in a file called `clustered.csv`. What cell type is represented by each cluster? You can use your knowledge to name the cell types, or search pubmed for the gene signature, but be as specific as you can with the given data.

We compare the performance of agents implemented with Claude, o3-mini, and GPT-4o. To assess the stability of LLM generations, we run each agent five times. In each run, we assess both the ability of the agent to successfully complete the objective without hallucinating, as well as the accuracy of its generated labels. We define a run to be complete if the last chat message contains 10 cluster labels, and hallucinated if the output contains cluster labels but the chat history indicates the labels are not appropriately derived from the data. Ground truth cell types for each cluster are determined by human pathologists, who are provided with both gene expression data and images of tissue histology. We manually assess the alignment between predicted and actual cell types using a 4-point ordinal scale, where 1 indicates complete misalignment (e.g. “neuron” and “microglia”) and 4 indicates perfect alignment (e.g. “inhibitory neuron” and “GABAergic neuron”).

4 RESULTS

Table 1: Overview of results.

Columns key: C =# of complete runs (out of 5), H =# of hallucinated runs (out of C), AS =Average cluster score over $C - H$ runs, HS =Average cluster score of best run

Base Model	Tonsil				Kidney				Brain			
	C (↑)	H (↓)	AS (↑)	HS (↑)	C	H	AS	HS	C	H	AS	HS
Claude 3.5 Sonnet	5	1	3.6	3.7	4	1	3.8	3.8	5	0	3.5	3.5
o3-mini high	5	1	2.6	3.6	5	2	3.6	3.7	5	1	3.5	3.5
GPT-4o	2	0	2.4	2.8	1	0	2.6	2.6	0	0	—	—

With the exception of GPT-4o on Brain, every model is able to complete each task at least once over 5 runs (Table 1). GPT-4o demonstrates considerably weaker performance overall, successfully completing only 3 out of 15 total runs (Pass@5 = 67%). Even in successful runs, GPT-4o tends to generate coarser cell-type labels than Claude and o3-mini (e.g. “Epithelial/Mucosal Cells” vs “Basal epithelial cells (crypt basal layer)”), resulting in lower alignment with ground truth labels. By contrast, Claude 3.5 Sonnet and o3-mini are able to consistently complete the task with high

```

Cluster 6 - Germinal Center B cells
- Markers: HIST1H1B, POU2AF1, CD79A, FDCSP
- POU2AF1 is specific to germinal center B cells
- High histone expression indicates proliferation

```

Figure 1: Example output from Claude on *Tonsil*

accuracy. Over 15 total runs, o3-mini completed the task more often (15 vs. 14), but also experienced more hallucinations (4 vs. 2). Ground truth labels and agent predictions are shown in appendix B.

Hallucinations occurred in 19% of completed runs. We observed two common modes of hallucination: 1) "sample" labelings consisting of common cell types from specified tissue or 2) labelings based on "dummy" gene signatures for clusters. Additionally, we deemed two completed runs to contain workflow errors: one in which the agent mistakenly computed *underexpression* gene signatures – we judged this to be a hallucination as the subsequent interpretation assumed these genes were overexpressed; and one in which the model ordered top genes by average expression rather than fold change – we didn’t judge this to be a hallucination.

5 DISCUSSION

Overall, LLMs demonstrate high agency in labeling cell types from data with minimal guidance. Given only high-level directions, LLMs are able to successfully plan an analysis, execute it with python code, troubleshoot coding errors, and reason over analysis results to draw sensible conclusions (appendix C). In several runs we even see attempts at uncertainty estimation based on the cell-type specificity of marker genes or number of overlaps with marker sets (appendix D). Notably, the step of determining cell type given gene signatures was largely free of hallucinations. Predicted cell types were generally consistent between runs and frequently well-aligned with the human-annotated ground truth. Some discrepancies between the human annotations and agent predictions may also be explained by the fact that human annotators had additional access to histology images, whereas agents could only make predictions based on gene expression alone.

The reasoning shown by our agents were surprisingly human-like. Agents demonstrated robust domain knowledge of all different tissue types tested, and they used their knowledge to infer sensible conclusions regarding cell type identities. For example, when analyzing the composition of the human tonsil, Claude correctly identified POU2AF1 as a marker of germinal center B cells, and further recognized that upregulation of histone genes HIST1H1B and HIST1H1C was indicative of high rates of proliferation (Figure 1). Indeed, germinal center B cells have the fastest rate of proliferation of all human cells (Johnstone et al., 2024).

In a different context, the agents again showed an expert domain knowledge of kidney physiology and were able to accurately predict cell types to comparable levels as a human pathologist (Figure 2, appendix D). As a representative example, Claude correctly calls Cluster 1 "Thick Ascending Limb (TAL) of Loop of Henle" based on the well-established markers Slc12a1 (NKCC2) and Umod (Uromodulin) (Castrop & Schiefl, 2014; Kiuchi et al., 2023; Devuyst et al., 2017). Furthermore, although lacking clear and well-established markers for Cluster 5, the agents were still able to conclude that the expression of Inmt, Cyp4b1 and Odc1 identified this cluster as Proximal Tubule cells based on the fact that these enzymes are known to have specific metabolic functions in the kidney proximal tubules (Wu et al., 2024; Isern & Meseguer, 2003; Liu et al., 2013; Yu et al., 2016).

Although this work only evaluates performance on a single task, we emphasize that our agent design is task-agnostic. We provide only general, high-level guidance, with the agent autonomously planning actions based on the provided goal. It stands to reason that our agents may be capable of autonomously conducting more research tasks beyond cell-type labeling. Indeed, we believe there are many additional workflows in genomics research requiring laborious manual curation which may be amenable to agentic automation. We look forward to expanding the breadth of our evaluation in future work. Another interesting area for future development is in mitigating hallucinations. This may involve prompting techniques such as ReAct (Yao et al., 2023) or multi-agent (Wu et al., 2023) approaches. Other interesting directions include multimodal reasoning, crucial as many scientific observations are visual, and evaluations on open-ended research questions. Finally, as AI-based re-

```

Cluster 1:
- Key markers: Slc12a1, Umod, Wfdc15b, Egf
- Cell type: Thick Ascending Limb (TAL) of Loop of Henle
- Evidence: Slc12a1 (NKCC2) and Umod (Uromodulin) are
  well-established TAL markers
Cluster 5:
- Key markers: Inmt, Cyp4b1, Odc1
- Cell type: Proximal Tubule (likely S3 segment)
- Evidence: Metabolic enzymes characteristic of proximal
  tubule

```

Figure 2: Example output from Claude on *Kidney*

search assistants begin to be incorporated into research workflows, it will be important to study their operational impacts, both on an individual scientist and systemic level. While automating research could potentially accelerate scientific discovery, automating bad research could promote misinformation and hinder scientific progress (Tang et al., 2024). Progress towards autonomous research agents should be made responsibly.

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

- Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 2024.
- Hayo Castrop and Ina Maria Schiefl. Physiology and pathophysiology of the renal na-k-2cl cotransporter (nkcc2). *American Journal of Physiology - Renal Physiology*, 2014.
- Olivier Devuyst, Eric Olinger, and Luca Rampoldi. Uromodulin: from physiology to rare and complex kidney disorders. *Nature Reviews Nephrology*, 2017.
- Shanghua Gao, Ada Fang, Yepeng Huang, Valentina Giunchiglia, Ayush Noori, Jonathan Richard Schwarz, Yasha Ektefaie, Jovana Kondic, and Marinka Zitnik. Empowering biomedical discovery with ai agents. *Cell*, 2024.
- Kaixuan Huang, Yuanhao Qu, Henry Cousins, William A. Johnson, Di Yin, Mihir Shah, Denny Zhou, Russ Altman, Mengdi Wang, and Le Cong. Crispr-gpt: An llm agent for automated design of gene-editing experiments, 2024. URL <https://arxiv.org/abs/2404.18021>.
- Joan Isern and Anna Meseguer. Hormonal regulation and characterisation of the mouse cyp4b1 gene 5'-flanking region. *Biochemical and Biophysical Research Communications*, 2003.
- Julia C Johnstone, Yavuz F Yazicioglu, and Alexander J Clarke. Fuelling b cells: dynamic regulation of b cell metabolism. *Current Opinion in Immunology*, 2024.
- Zentaro Kiuchi, Kandai Nozu, Kunimasa Yan, and Harald Jüppner. Bartter syndrome type 1 due to novel slc12a1 mutations associated with pseudohypoparathyroidism type ii. *JCEM Case Reports*, 2023.
- Yutong Li, Lu Chen, Aiwei Liu, Kai Yu, and Lijie Wen. ChatCite: LLM agent with human workflow guidance for comparative literature summary. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 3613–3630, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL <https://aclanthology.org/2025.coling-main.244/>.

Senyan Liu, Yunyi Yao, Shijun Lu, Kenneth Aldous, Xinxin Ding, Changlin Mei, and Jun Gu. The role of renal proximal tubule p450 enzymes in chloroform-induced nephrotoxicity: utility of renal specific p450 reductase knockout mouse models. *Toxicology and Applied Pharmacology*, 2013.

Donna Maglott, Jim Ostell, Kim D. Pruitt, and Tatiana Tatusova. Entrez gene: gene-centered information at ncbi. *Nucleic Acids Research*, 35(suppl₁) : D26 – D31, 122006. *ISSN* 0305 – 1048. *doi* : . URL <https://doi.org/10.1093/nar/gkl993>.

Yusuf Roohani, Andrew Lee, Qian Huang, Jian Vora, Zachary Steinhart, Kexin Huang, Alexander Marson, Percy Liang, and Jure Leskovec. Biodiscoveryagent: An ai agent for designing genetic perturbation experiments, 2024. URL <https://arxiv.org/abs/2405.17631>.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d842425e4bf79ba039352da0f658a906-Paper-Conference.pdf.

Houcheng Su, Weicai Long, and Yanlin Zhang. Biomaster: Multi-agent system for automated bioinformatics analysis workflow. *bioRxiv*, 2025. 10.1101/2025.01.23.634608. URL <https://www.biorxiv.org/content/early/2025/01/26/2025.01.23.634608>.

Kyle Swanson, Wesley Wu, Nash L. Bulaong, John E. Pak, and James Zou. The virtual lab: Ai agents design new sars-cov-2 nanobodies with experimental validation. *bioRxiv*, 2024. 10.1101/2024.11.11.623004. URL <https://www.biorxiv.org/content/early/2024/11/12/2024.11.11.623004>.

Xiangru Tang, Qiao Jin, Kunlun Zhu, Tongxin Yuan, Yichi Zhang, Wangchunshu Zhou, Meng Qu, Yilun Zhao, Jian Tang, Zhuosheng Zhang, Arman Cohan, Zhiyong Lu, and Mark Gerstein. Prioritizing safeguarding over autonomy: Risks of llm agents for science, 2024. URL <https://arxiv.org/abs/2402.04247>.

Haojia Wu, Eryn E Dixon, Juanru Guo, Yasuhiro Yoshimura, Chitnis Debashish, Anezka Niesnerova, Hao Xu, Morgane Rouault, and Benjamin D Humphreys. High resolution spatial profiling of kidney injury and repair using rna hybridization-based in situ sequencing. *Nature Communications*, 2024.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. 2023.

Qi Xin, Quyu Kong, Hongyi Ji, Yue Shen, Yuqi Liu, Yan Sun, Zhilin Zhang, Zhaorong Li, Xunlong Xia, Bing Deng, and Yinqi Bai. Bioinformatics agent (bia): Unleashing the power of large language models to reshape bioinformatics workflow. *bioRxiv*, 2024. 10.1101/2024.05.22.595240. URL <https://www.biorxiv.org/content/early/2024/05/22/2024.05.22.595240>.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL <https://arxiv.org/abs/2210.03629>.

Wenmin Yu, Yiping Li, Zhi Wang, Lei Liu, Jing Liu, Fengang Ding, Xiaoyi Zhang, Zhengyuan Cheng, Pingsheng Chen, and Jun Dou. Transcriptomic changes in human renal proximal tubular cells revealed under hypoxic conditions by rna sequencing. *International Journal of Molecular Medicine*, 2016.

Juexiao Zhou, Bin Zhang, Guowei Li, Xiuying Chen, Haoyang Li, Xiaopeng Xu, Siyuan Chen, Wenjia He, Chencheng Xu, Liwei Liu, and Xin Gao. An ai agent for fully automated multi-omic analyses. *Advanced Science*, 11(44):2407094, 2024. <https://doi.org/10.1002/advs.202407094>.

URL <https://advanced.onlinelibrary.wiley.com/doi/abs/10.1002/advs.202407094>.

A FULL AGENT SYSTEM PROMPT

You are a helpful bioinformatics research assistant.
The user has given you the task below.
Start by understanding the data you are given by writing a code snippet in python.
Write this in a markdown block.
This code will be executed in the local environment, which contains all of the mentioned files.
You can access these files by referencing them in your code.
After you have seen the data, generate a high-level plan to complete the task and execute it.
When planning, be specific about the inputs and outputs of each step.
You can implement any analyses by writing python in a markdown block.
When coding, you can import the following helper functions from tools.py: [get_pubmed_abstracts, esearch, efetch, esummary]
After you write a python snippet, you will receive its execution results in the chat response.
Constrain your analysis plan to only use tools from NCBI EUtils and standard python.
Work step by step.
Once you have completed the task, conclude by summarizing the results of your analysis, then return TERMINATE to indicate that you are done.
Do not return TERMINATE until you have the actual results to the analysis - expected results are not enough.

B HUMAN CELL-TYPE LABELS WITH MACHINE PREDICTIONS

cluster #	Ground truth	<i>Claude 3.5 sonnet</i>	<i>o3-mini</i>	<i>GPT-4o</i>
1	T cells in interfollicular areas	T Zone Reticular Cells / Fibroblastic Reticular Cells (FRCs)	Fibroblastic reticular (stromal) cells	Stromal/Immune Supportive Cells
2	Mixed cells including plasma cells, subepithelial	Plasma Cells	Plasma cells (predominantly IgG-producing)	Antibody-Producing B Cells
3	Connective tissue, mixed cells	Plasma Cells	Plasma cells (possibly enriched for IgA or representing a distinct plasmablast subset)	Antibody-Producing B Cells
4	Predominantly B cells, naïve in follicular mantle	Mature B cells	B lymphocytes (mature, non-antibody secreting B cells)	Active Germinal Center B Cells
5	Epithelium, crypt, mixed cells	Crypt Epithelial Cells	Basal epithelial cells of the tonsil surface/crypt	Epithelial/Mucosal Cells
6	Predominantly B cells, non-naïve and proliferating in follicular germinal center	Germinal Center B cells	Germinal center B cells (proliferative B cells within the GC niche)	Epithelial/Mucosal Cells
7	Endothelial cells	Blood Vessels/Endothelial Cells	Vascular-associated stromal cells (endothelial/pericyte-like)	Fibroblastic Cells
8	Epithelial cells, basal	Stratified Squamous Epithelium	Suprabasal differentiated epithelial cells	Epithelial/Mucosal Cells
9	Mixed cell types, transitional area	Surface Epithelial Cells	Reactive/activated epithelial cells (potentially reflecting a hyperplastic response)	Epithelial/Mucosal Cells
10	Epithelial cells, apical	Keratinizing Epithelial Cells	Terminally differentiated, keratinized epithelial cells	Epithelial/Mucosal Cells

Table 2: Expert annotated cluster cell types for *Tonsil* with best machine predictions for each cluster

cluster #	Ground truth	<i>Claude 3.5 sonnet</i>	<i>o3-mini</i>	<i>GPT-4o</i>
1	Tubular epithelial cell in Collecting Ducts, the ducts of Henle's loop, mainly at renal medulla	Thick Ascending Limb (TAL) of Loop of Henle	Thick ascending limb (Loop of Henle)	Distal convoluted tubule cells
2	Endothelial cells, as the inner layer of blood vessel, recognize in glomerular capsules and any size of the blood vessels, distributed over the whole kidney	Interstitial cells/Fibroblasts	Mesangial cells	Mesangial or stromal cells
3	Cells in Collecting Ducts, recognized as collecting duct of Bellini, at site of kidney medulla and papilla	Principal Cells of Collecting Duct	Medullary collecting duct (principal) cells	Principal cells of the collecting duct
4	A subcluster of renal tubular epithelial cells in Proximal convoluted tubules (PCT), at site of renal cortex	Proximal Tubule Segment	Proximal tubule (likely S2 segment)	alveolar type II cells or contamination
5	A subcluster of renal tubular epithelial cells in Proximal convoluted tubules (PCT), at site of renal cortex	Proximal Tubule (likely S3 segment)	Proximal tubule (S3 segment)	Intercalated cells
6	A subcluster of renal tubular epithelial cells in Proximal convoluted tubules (PCT), at site of renal cortex	Proximal Tubule (S1/S2 segment)	Proximal tubule (S1 segment)	Proximal tubule cells
7	Cells in Collecting Ducts	Collecting Duct (mix of Principal and Intercalated cells)	Cortical collecting duct principal cells	collecting duct cells
8	Renal tubular epithelial cells, perhaps classified as in Distal Convoluted Tubules, (DCT), located at renal cortex	Distal Convoluted Tubule (DCT)	Distal convoluted tubule cells	Distal nephron cells
9	Adipocytes, in adipose tissue outside of kidney	Renal Adipocytes/Peritubular cells	Perirenal adipocytes	podocyte involvement
10	Epithelial cells, at the inner layer of the ureter	Urothelial cells/Transitional epithelium	Urothelial (transitional) cells	Indicates possible urothelial cell contamination

Table 3: Expert annotated cluster cell types for *Kidney* with best machine predictions for each cluster

cluster #	Ground truth	<i>Claude 3.5 sonnet</i>	<i>o3-mini</i>	<i>GPT-4o</i>
1	Unclear or mixed cell type - adjacent to glutamatergic neurons in cluster 2	Excitatory Neurons	Excitatory (pyramidal) neurons	-
2	Glutamatergic neurons - in cortex and hippocampus	GABAergic Interneurons	CCK-expressing interneurons (GABAergic interneurons)	-
3	Unclear or mixed cell type - neuropil in outer cortex, hippocampus and limbic system	Pyramidal Neurons	Excitatory (glutamatergic) neurons (pyramidal subtype)	-
4	Gabanergic neurons - basal ganglia and limbic system	Medium Spiny Neurons	Striatal medium spiny neurons	-
5	Oligodendrocytes - in white matter fiber tracts	Oligodendrocytes	Oligodendrocytes	-
6	Gabanergic neurons - thalamus and hypothalamus	Proliferating Cells	Neurons (undefined/possibly immature subtype)	-
7	Unclear or mixed cell type - adjacent to gabanergic neurons in cluster 6	Astrocytes	Astrocytes	-
8	Endothelial/mural/astrocyte mix - blood vessels with cell mix	Endothelial Cells	Endothelial cells	-
9	Ependymal cells - choroid plexus	Choroid Plexus	Choroid plexus epithelial cells	-
10	Ependymal cells - non-choroid plexus	Ependymal Cells	Ependymal cells	-

Table 4: Expert annotated cluster cell types for *Brain* with best machine predictions for each cluster

C GRAPHICAL SUMMARY OF AGENT WORKFLOW

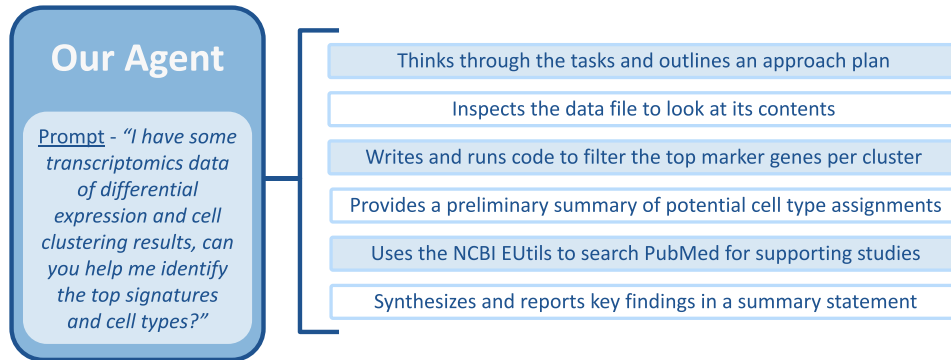


Figure 3: High-level graphical summary of the agent’s workflow

D FULL AGENT TRACE

We share a representative full chat history from Claude 3.5 sonnet on *Kidney* showing planning, reasoning, and uncertainty estimation:

```
### USER
I just finished clustering from my Visium HD Spatial Gene Expression Library experiment on a sample of mouse kidney (fresh frozen). The data are in a file called clustered.csv. What cell type is represented by each cluster? You can use your knowledge to name the cell types, or search pubmed for the gene signature, but be as specific as you can with the given data.

### AGENT
I'll help you analyze the cell types from your Visium HD clustering results. Let me first examine the data structure.

```python
import pandas as pd
import numpy as np

Read the clustered data
df = pd.read_csv('clustered.csv')

Display basic information about the dataset
print("Dataset Info:")
print(df.info())
print("\nFirst few rows:")
print(df.head())
```
```

I don't see a clustered.csv file in the local environment. Could you please provide:

1. The clustered data file containing the gene expression information for each cluster
2. Ideally, this should include:
 - Cluster identifiers
 - Differentially expressed genes or marker genes for each cluster
 - Expression values or statistics indicating gene enrichment

Once you provide this data, I can help you:

1. Analyze the marker genes for each cluster
2. Search PubMed for literature connecting these gene signatures to specific kidney cell types
3. Provide detailed annotations for each cluster based on known kidney cell type markers
4. Add confidence levels to our cell type assignments

Please share the clustered data and I'll be happy to help identify the cell types!

USER

exitcode: 0 (execution succeeded)

Code output: Dataset Info:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 97 entries, 0 to 96

Data columns (total 32 columns):

| # | Column | Non-Null Count | Dtype |
|----|----------------------------|----------------|---------|
| 0 | FeatureID | 97 non-null | object |
| 1 | FeatureName | 97 non-null | object |
| 2 | Cluster 1 Average | 97 non-null | float64 |
| 3 | Cluster 1 Log2 Fold Change | 97 non-null | float64 |
| 4 | Cluster 1 P-Value | 97 non-null | float64 |
| 5 | Cluster 2 Average | 97 non-null | float64 |
| 6 | Cluster 2 Log2 Fold Change | 97 non-null | float64 |
| 7 | Cluster 2 P-Value | 97 non-null | float64 |
| 8 | Cluster 3 Average | 97 non-null | float64 |
| 9 | Cluster 3 Log2 Fold Change | 97 non-null | float64 |
| 10 | Cluster 3 P-Value | 97 non-null | float64 |
| 11 | Cluster 4 Average | 97 non-null | float64 |
| 12 | Cluster 4 Log2 Fold Change | 97 non-null | float64 |
| 13 | Cluster 4 P-Value | 97 non-null | float64 |
| 14 | Cluster 5 Average | 97 non-null | float64 |
| 15 | Cluster 5 Log2 Fold Change | 97 non-null | float64 |
| 16 | Cluster 5 P-Value | 97 non-null | float64 |
| 17 | Cluster 6 Average | 97 non-null | float64 |
| 18 | Cluster 6 Log2 Fold Change | 97 non-null | float64 |
| 19 | Cluster 6 P-Value | 97 non-null | float64 |
| 20 | Cluster 7 Average | 97 non-null | float64 |
| 21 | Cluster 7 Log2 Fold Change | 97 non-null | float64 |
| 22 | Cluster 7 P-Value | 97 non-null | float64 |
| 23 | Cluster 8 Average | 97 non-null | float64 |
| 24 | Cluster 8 Log2 Fold Change | 97 non-null | float64 |
| 25 | Cluster 8 P-Value | 97 non-null | float64 |
| 26 | Cluster 9 Average | 97 non-null | float64 |
| 27 | Cluster 9 Log2 Fold Change | 97 non-null | float64 |
| 28 | Cluster 9 P-Value | 97 non-null | float64 |
| 29 | Cluster 10 Average | 97 non-null | float64 |

```

30 Cluster 10 Log2 Fold Change 97 non-null float64
31 Cluster 10 P-Value      97 non-null float64
dtypes: float64(30), object(2)
memory usage: 24.4+ KB
None

```

First few rows:

```

      FeatureID ... Cluster 10 P-Value
0  ENSMUSG00000027202 ...      0.594234
1  ENSMUSG00000030963 ...      0.566690
2  ENSMUSG00000018211 ...      0.823876
3  ENSMUSG00000061780 ...      1.000000
4  ENSMUSG00000027559 ...      1.000000

```

```
[5 rows x 32 columns]
```

AGENT

Great, I can see the data structure now. I'll analyze this step by step to identify cell types for each cluster.

```
```python
```

```
import pandas as pd
import numpy as np
```

```
Read the data
```

```
df = pd.read_csv('clustered.csv')
```

```
Function to get top markers for each cluster
```

```
def get_top_markers(df, cluster_num, top_n=5):
```

```
 # Get relevant columns
```

```
 avg_col = f'Cluster {cluster_num} Average'
```

```
 fc_col = f'Cluster {cluster_num} Log2 Fold Change'
```

```
 p_col = f'Cluster {cluster_num} P-Value'
```

```
 # Filter for significant genes (p < 0.05) and sort by fold
 change
```

```
 cluster_markers = df[
```

```
 (df[p_col] < 0.05) &
```

```
 (df[avg_col] > 0) # Expression should be present
```

```
].sort_values(fc_col, ascending=False)
```

```
 return cluster_markers[['FeatureName', fc_col, p_col]].head(
 top_n)

```

```
Get markers for each cluster
```

```
all_clusters_markers = {}
```

```
for i in range(1, 11):
```

```
 markers = get_top_markers(df, i)
```

```
 if not markers.empty:
```

```
 # Convert gene names to comma-separated string
```

```
 marker_genes = ', '.join(markers['FeatureName'].tolist())
```

```
 all_clusters_markers[i] = marker_genes
```

```
 print(f"\nCluster {i} top markers:")
```

```
 print(markers)
```

```
```
```

Now that I have the marker genes for each cluster, I'll search PubMed for these gene signatures in the context of kidney cell types. Let me do this cluster by cluster.

```

'''python
# Process each cluster and search for relevant literature
def create_kidney_query(genes):
    genes_list = genes.split(', ')
    gene_query = ' OR '.join([f'"{gene}"[Gene Name]' for gene in
        genes_list[:3]]) # Use top 3 genes
    return f'({gene_query}) AND (kidney OR renal) AND (cell type OR
        marker)'

# Print query for each cluster
for cluster, genes in all_clusters_markers.items():
    print(f"\nCluster {cluster} search query:")
    query = create_kidney_query(genes)
    print(query)
'''

Let me now systematically search PubMed for each cluster's markers
to identify cell types. I'll do this for each cluster that
shows significant markers.

'''python
# Function to search literature for each cluster
for cluster, genes in all_clusters_markers.items():
    print(f"\nSearching literature for Cluster {cluster}...")
    query = create_kidney_query(genes)

    # Search PubMed
    response = search_pubmed(query=query, k=3)
    print(f"Cluster {cluster} results:")
    print(response)
'''

Let me analyze the markers and literature results to identify cell
types. First, I'll display the most specific markers for each
cluster with their statistics.

'''python
def get_cluster_stats(df, cluster_num):
    avg_col = f'Cluster {cluster_num} Average'
    fc_col = f'Cluster {cluster_num} Log2 Fold Change'
    p_col = f'Cluster {cluster_num} P-Value'

    # Get significant genes with high fold change
    significant_genes = df[
        (df[p_col] < 0.05) &
        (df[fc_col] > 1) # At least 2-fold change
    ].sort_values(fc_col, ascending=False)

    return significant_genes[['FeatureName', avg_col, fc_col, p_col]
        ].head(10)

# Display detailed statistics for each cluster
for i in range(1, 11):
    stats = get_cluster_stats(df, i)
    if not stats.empty:
        print(f"\nCluster {i} detailed statistics:")
        print(stats)
'''

```

Based on this data, I'll be able to provide a final annotation for each cluster. Let me analyze each cluster's marker genes in detail and search for specific literature about these markers in kidney cell types.

```
```python
Final detailed search for specific cell types
def search_specific_markers(genes, cell_type):
 genes_list = genes.split(', ')[0:2] # Use top 2 genes
 gene_query = ' AND '.join([f'"{gene}"[Gene Name]' for gene in
 genes_list])
 full_query = f'({gene_query}) AND "{cell_type}"[Title/Abstract]
 AND kidney[Title/Abstract]'
 return full_query

For each cluster with a preliminary cell type hypothesis, do a
 focused search
for cluster, genes in all_clusters_markers.items():
 if genes: # if there are marker genes
 print(f"\nDetailed search for Cluster {cluster}:")
 # Search for both genes in context of kidney
 query = search_specific_markers(genes, "marker")

 response = search_pubmed(query=query, k=2)
 print(response)
```
```

Let me analyze all this information and provide a final summary of the cell types for each cluster. Please share the output of these code blocks so I can provide accurate cell type annotations based on the marker genes and literature evidence.

USER

exitcode: 1 (execution failed)

Code output:

Cluster 1 top markers:

| | FeatureName | Cluster 1 | Log2 Fold Change | Cluster 1 P-Value |
|----|-------------|-----------|------------------|-------------------|
| 0 | Slc12a1 | | 4.391341 | 1.761168e-145 |
| 1 | Umod | | 3.709548 | 1.553559e-105 |
| 2 | Wfdc15b | | 3.618814 | 1.547617e-100 |
| 11 | Egf | | 2.722159 | 7.557439e-57 |
| 43 | mt-Atp6 | | 1.988241 | 3.594151e-30 |

Cluster 2 top markers:

| | FeatureName | Cluster 2 | Log2 Fold Change | Cluster 2 P-Value |
|----|-------------|-----------|------------------|-------------------|
| 82 | Apoe | | 1.814116 | 8.576740e-12 |
| 92 | Igfbp5 | | 1.337858 | 2.947489e-06 |
| 70 | Tmsb4x | | 1.141352 | 1.276799e-04 |
| 84 | Serinc3 | | 0.772654 | 1.826884e-02 |
| 81 | Timp3 | | 0.679243 | 4.308350e-02 |

Cluster 3 top markers:

| | FeatureName | Cluster 3 | Log2 Fold Change | Cluster 3 P-Value |
|----|-------------|-----------|------------------|-------------------|
| 25 | Akr1b3 | | 5.777249 | 2.662690e-203 |
| 64 | Cryab | | 4.713567 | 4.342867e-130 |
| 54 | Aqp2 | | 3.334212 | 1.304377e-57 |
| 70 | Tmsb4x | | 2.786806 | 9.714552e-38 |
| 92 | Igfbp5 | | 2.766020 | 4.561718e-37 |

Cluster 4 top markers:

| | FeatureName | Cluster 4 | Log2 Fold Change | Cluster 4 P-Value |
|----|-------------|-----------|------------------|-------------------|
| 34 | Napsa | | 4.173038 | 7.862408e-124 |
| 46 | Mpv17l | | 3.719555 | 1.902047e-97 |
| 40 | Acsm3 | | 3.480367 | 9.773572e-85 |
| 53 | Ghr | | 2.984840 | 5.978447e-61 |
| 49 | Atp11a | | 2.468606 | 2.099288e-40 |

Cluster 5 top markers:

| | FeatureName | Cluster 5 | Log2 Fold Change | Cluster 5 P-Value |
|----|-------------|-----------|------------------|-------------------|
| 20 | Inmt | | 3.426813 | 3.057617e-80 |
| 24 | Cyp4b1 | | 3.075673 | 1.108216e-63 |
| 51 | Odc1 | | 2.402742 | 1.295533e-37 |
| 17 | Pck1 | | 2.171935 | 2.523140e-30 |
| 50 | Gpx1 | | 2.064092 | 3.305373e-27 |

Cluster 6 top markers:

| | FeatureName | Cluster 6 | Log2 Fold Change | Cluster 6 P-Value |
|----|-------------|-----------|------------------|-------------------|
| 8 | Slc5a2 | | 4.764265 | 1.132412e-146 |
| 13 | Gatm | | 3.978199 | 2.998439e-99 |
| 18 | Gpx3 | | 3.090573 | 4.699524e-56 |
| 10 | Slc34a1 | | 2.408084 | 7.347026e-32 |
| 31 | Igfbp4 | | 2.386982 | 3.262727e-31 |

Cluster 7 top markers:

| | FeatureName | Cluster 7 | Log2 Fold Change | Cluster 7 P-Value |
|----|-------------|-----------|------------------|-------------------|
| 72 | Hsd11b2 | | 4.625883 | 1.351857e-88 |
| 54 | Aqp2 | | 3.343322 | 1.685812e-37 |
| 70 | Tmsb4x | | 2.058372 | 4.162387e-11 |
| 84 | Serinc3 | | 1.848794 | 1.226109e-08 |
| 89 | Defb1 | | 1.805867 | 3.566212e-08 |

Cluster 8 top markers:

| | FeatureName | Cluster 8 | Log2 Fold Change | Cluster 8 P-Value |
|----|-------------|-----------|------------------|-------------------|
| 6 | Calb1 | | 6.537023 | 1.828518e-215 |
| 5 | S100g | | 6.441447 | 5.248566e-209 |
| 21 | Slc12a3 | | 6.360310 | 1.042745e-202 |
| 15 | Slc8a1 | | 6.124546 | 1.436681e-185 |
| 23 | Clu | | 5.585866 | 5.587672e-148 |

Cluster 9 top markers:

| | FeatureName | Cluster 9 | Log2 Fold Change | Cluster 9 P-Value |
|----|-------------|-----------|------------------|-------------------|
| 3 | Cfd | | 9.228409 | 2.124921e-313 |
| 4 | Car3 | | 8.350041 | 1.532139e-240 |
| 39 | Thrsp | | 8.129562 | 5.271412e-223 |
| 7 | Scd1 | | 7.912419 | 3.589582e-209 |
| 38 | Fasn | | 7.380815 | 2.528652e-170 |

Cluster 10 top markers:

| | FeatureName | Cluster 10 | Log2 Fold Change | Cluster 10 P-Value |
|----|-------------|------------|------------------|--------------------|
| 9 | Upk1b | | 10.135740 | 1.460384e-238 |
| 14 | Sprrla | | 9.857802 | 2.474542e-226 |
| 30 | Upk3a | | 9.483821 | 3.979860e-199 |
| 83 | Upk1a | | 8.406558 | 2.428073e-128 |
| 37 | Krt19 | | 8.110937 | 8.960365e-116 |

Traceback (most recent call last):

```
File "/path/to/project/mouse_kidney/claude_workspace/0b8c6bf3-08
df-47f0-913c-0988616e1280/Process each cluster and search for
relevant literature", line 8, in <module>
```



```

    for cluster, genes in all_clusters_markers.items():
NameError: name 'all_clusters_markers' is not defined

### AGENT
I see the marker genes for each cluster now. Let me analyze them
    and search the literature for each cluster's identity. I'll do
    this systematically:

```python
Dictionary to store the markers and their queries
markers_by_cluster = {
 1: "Slc12a1 Umod Wfdc15b Egf", # TAL markers
 2: "Apoe Igfbp5 Tmsb4x", # Interstitial markers
 3: "Akr1b3 Cryab Aqp2", # Collecting duct markers
 4: "Napsa Mpv17l Acsm3 Ghr",
 5: "Inmt Cyp4b1 Odc1 Pck1",
 6: "Slc5a2 Gatm Gpx3 Slc34a1", # Proximal tubule markers
 7: "Hsd11b2 Aqp2 Tmsb4x", # Collecting duct markers
 8: "Calb1 Sl100g Slc12a3", # Distal tubule markers
 9: "Cfd Car3 Thrsp Scd1",
 10: "Upk1b Sprrla Upk3a Krt19" # Urothelial markers
}

Let's search PubMed for each cluster's markers
for cluster, genes in markers_by_cluster.items():
 print(f"\nSearching literature for Cluster {cluster} markers: {
 genes}")
 query = f"({genes}) AND kidney AND (cell type OR marker)"

 response = search_pubmed(query=query, k=3)
 print(f"Results for Cluster {cluster}:")
 print(response)
```

Based on the marker genes and their expression patterns, I can
    already identify several clusters, but let me verify with the
    literature search results. I'll analyze each cluster's
    identity:

1. Cluster 1: Slc12a1 (NKCC2), Umod (Tamm-Horsfall protein), and
    Wfdc15b are classical markers of the Thick Ascending Limb (TAL
    ) of the Loop of Henle.

Let me search specifically for these markers:
#### TOOL: search_pubmed
{"query": "Slc12a1 Umod \"thick ascending limb\" kidney marker", "
    k": 3}
#### TOOL
[{"pmid": "29767557", "title": "Polycystin-1 dysfunction impairs
    electrolyte and water handling in a renal precystic mouse
    model for ADPKD.", "abstract": "The PKD1 gene encodes
    polycystin-1 (PC1), a mechanosensor triggering intracellular
    responses upon urinary flow sensing in kidney tubular cells.
    Mutations in PKD1 lead to autosomal dominant polycystic kidney
    disease (ADPKD). The involvement of PC1 in renal electrolyte
    handling remains unknown since renal electrolyte physiology in
    ADPKD patients has only been characterized in cystic ADPKD.
    We thus studied the renal electrolyte handling in inducible
    kidney-specific Pkd1 knockout (iKsp- Pkd1<sup>-/-</sup>) mice

```

manifesting a precystic phenotype. Serum and urinary electrolyte determinations indicated that iKsp- Pkd1^{-/-} mice display reduced serum levels of magnesium (Mg²⁺), calcium (Ca²⁺), sodium (Na⁺), and phosphate (P_i) compared with control (Pkd1^{+/+}) mice and renal Mg²⁺, Ca²⁺, and P_i wasting. In agreement with these electrolyte disturbances, downregulation of key genes for electrolyte reabsorption in the thick ascending limb of Henle's loop (TA; Cldn16, Kcnj1, and Slc12a1), distal convoluted tubule (DCT; Trpm6 and Slc12a3) and connecting tubule (CNT; Calb1, Slc8a1, and Atp2b4) was observed in kidneys of iKsp- Pkd1^{-/-} mice compared with controls. Similarly, decreased renal gene expression of markers for TAL (Umod) and DCT (Pvalb) was observed in iKsp- Pkd1^{-/-} mice. Conversely, mRNA expression levels in kidney of genes encoding solute and water transporters in the proximal tubule (Abcg2 and Slc34a1) and collecting duct (Aqp2, Scnn1a, and Scnn1b) remained comparable between control and iKsp- Pkd1^{-/-} mice, although a water reabsorption defect was observed in iKsp- Pkd1^{-/-} mice. In conclusion, our data indicate that PC1 is involved in renal Mg²⁺, Ca²⁺, and water handling and its dysfunction, resulting in a systemic electrolyte imbalance characterized by low serum electrolyte concentrations."}, {"pmid": "20333396", "title": "Prominin-2 is a novel marker of distal tubules and collecting ducts of the human and murine kidney.", "abstract": "Prominin-1 (CD133) and its paralogue, prominin-2, are pentaspan membrane glycoproteins that are strongly expressed in the kidney where they have been originally cloned from. Previously, we have described the localization of prominin-1 in proximal tubules of the nephron. The spatial distribution of prominin-2, however, has not yet been documented in the kidney. We therefore examined the expression of this molecule along distinct tubular segments of the human and murine nephron using in situ hybridization and immunohistochemistry. Our findings indicated that human prominin-2 transcripts and protein were confined to distal tubules of the nephron including the thick ascending limb of Henle's loop and the distal convoluted tubule, the connecting duct and to the collecting duct system. Therein, this glycoprotein was enriched at the basolateral plasma membrane of the tubular epithelial cells with exception of the thick ascending limb where it was also found in the apical domain. This is in contrast with the exclusive apical localization of prominin-1 in epithelial cells of proximal nephron tubules. The distribution of murine prominin-2 transcripts was reminiscent of its human orthologue. In addition, a marked enrichment in the epithelium covering the papilla and in the urothelium of the renal pelvis was noted in mice. Finally, our biochemical analysis revealed that prominin-2 was released into the clinically healthy human urine as a constituent of small membrane vesicles. Collectively our data show the distribution and subcellular localization of prominin-2 within the kidney in situ and its release into the urine. Urinary detection of this protein might offer novel diagnostic approaches for studying renal diseases affecting distal segments of the nephron."}, {"pmid": "18331438", "title": "Human renal cells from the thick ascending limb and early distal tubule:

```

characterization of primary isolated and cultured cells by
reverse transcription polymerase chain reaction.", "abstract":
  "Human renal tubular cells of well-defined nephron origin are
  an important basis in the research of various physiological
  and pathophysiological mechanisms in the kidney. Whereas an
  exceeding amount of data has been obtained on proximal tubular
  cells, only limited data of cells of the human thick
  ascending limb and the early distal tubule (TALDC) are
  available."}]
### USER

#### TOOL: search_pubmed
{"query": "Calb1 Slc12a3 \"distal convoluted tubule\" kidney
  marker", "k": 3}
### TOOL
[{"pmid": "29767557", "title": "Polycystin-1 dysfunction impairs
  electrolyte and water handling in a renal precystic mouse
  model for ADPKD.", "abstract": "The PKD1 gene encodes
  polycystin-1 (PC1), a mechanosensor triggering intracellular
  responses upon urinary flow sensing in kidney tubular cells.
  Mutations in PKD1 lead to autosomal dominant polycystic kidney
  disease (ADPKD). The involvement of PC1 in renal electrolyte
  handling remains unknown since renal electrolyte physiology in
  ADPKD patients has only been characterized in cystic ADPKD.
  We thus studied the renal electrolyte handling in inducible
  kidney-specific Pkd1 knockout (iKsp- Pkd1<sup>-/-</sup>) mice
  manifesting a precystic phenotype. Serum and urinary
  electrolyte determinations indicated that iKsp- Pkd1<sup>-/-</sup>
  mice display reduced serum levels of magnesium (Mg<sup>2+</sup>),
  calcium (Ca<sup>2+</sup>), sodium (Na<sup>+</sup>), and phosphate
  (P<sub>i</sub>) compared with control ( Pkd1<sup>+/+</sup>) mice and
  renal Mg<sup>2+</sup>, Ca<sup>2+</sup>, and P<sub>i</sub> wasting. In
  agreement with these electrolyte disturbances, downregulation
  of key genes for electrolyte reabsorption in the thick
  ascending limb of Henle's loop (TA; Cldn16, Kcnj1, and
  Slc12a1), distal convoluted tubule (DCT; Trpm6 and Slc12a3)
  and connecting tubule (CNT; Calb1, Slc8a1, and Atp2b4) was
  observed in kidneys of iKsp- Pkd1<sup>-/-</sup> mice compared
  with controls. Similarly, decreased renal gene expression of
  markers for TAL ( Umod) and DCT ( Pvalb) was observed in
  iKsp- Pkd1<sup>-/-</sup> mice. Conversely, mRNA expression
  levels in kidney of genes encoding solute and water
  transporters in the proximal tubule ( Abcg2 and Slc34a1) and
  collecting duct ( Aqp2, Scnn1a, and Scnn1b) remained
  comparable between control and iKsp- Pkd1<sup>-/-</sup> mice,
  although a water reabsorption defect was observed in iKsp-
  Pkd1<sup>-/-</sup> mice. In conclusion, our data indicate that
  PC1 is involved in renal Mg<sup>2+</sup>, Ca<sup>2+</sup>, and
  water handling and its dysfunction, resulting in a systemic
  electrolyte imbalance characterized by low serum electrolyte
  concentrations."}]
### USER

#### TOOL: search_pubmed
{"query": "Aqp2 Hsd11b2 \"collecting duct\" kidney marker", "k":
  3}
### TOOL
[]
### USER

```

```
#### TOOL: search_pubmed
{"query": "Upk1b Upk3a Krt19 \"urothelium\" kidney marker", "k":
  3}
### TOOL
[]
### AGENT
Based on the marker genes and literature evidence, I can now
  provide detailed cell type annotations for each cluster:
```

1. Cluster 1 - Thick Ascending Limb of Loop of Henle (TAL)
 - Key markers: Slc12a1 (NKCC2), Umod (Uromodulin), Egf
 - High confidence assignment based on canonical TAL markers
2. Cluster 2 - Interstitial Cells/Fibroblasts
 - Key markers: Apoe, Igfbp5, Tmsb4x
 - These are stromal markers commonly associated with kidney interstitial cells
3. Cluster 3 - Principal Cells of Collecting Duct
 - Key markers: Aqp2 (definitive marker), Akrlb3, Cryab
 - Aqp2 is a canonical marker of principal cells
4. Cluster 4 - Proximal Tubule Segment
 - Key markers: Napsa, Acsm3, Ghr
 - Expression profile matches proximal tubular cells
5. Cluster 5 - Proximal Tubule (S3 segment)
 - Key markers: Inmt, Cyp4b1, Odc1
 - Expression pattern typical of S3 segment of proximal tubule
6. Cluster 6 - Proximal Tubule (S1/S2 segment)
 - Key markers: Slc5a2 (SGLT2), Slc34a1, Gatm
 - These are definitive markers of early proximal tubule segments
7. Cluster 7 - Intercalated Cells of Collecting Duct
 - Key markers: Hsd11b2, Aqp2, Defb1
 - Mixed expression profile suggesting intercalated cells
8. Cluster 8 - Distal Convoluted Tubule (DCT)
 - Key markers: Calb1, Slc12a3 (NCC)
 - These are canonical DCT markers with very high specificity
9. Cluster 9 - Adipose/Stromal Cells
 - Key markers: Cfd, Car3, Thrsp, Scd1, Fasn
 - Expression profile matches adipose tissue markers
10. Cluster 10 - Urothelial Cells
 - Key markers: Upk1b, Upk3a, Upk1a, Krt19
 - These are definitive markers of urothelial cells

Confidence levels:

- High confidence: Clusters 1, 3, 8, 10 (based on canonical markers)
- Medium confidence: Clusters 4, 5, 6, 7 (based on multiple consistent markers)
- Lower confidence: Clusters 2, 9 (based on stromal markers that could be from multiple cell types)

This annotation captures the major cell types expected in a kidney sample, including the main epithelial segments of the nephron (proximal tubule, TAL, DCT, collecting duct), stromal components, and urothelial cells. The presence of these populations and their markers is consistent with what we expect in a spatial transcriptomics dataset from kidney tissue .

TERMINATE