

1 **Dendrite: A Structured, Accessible, and Queryable Pathology Search Database for**
2 **Streamlined Experiment Planning**

3 Yunrui Lu^{1,*}, Robert Hamilton^{1,*}, Jack Greenberg², Gokul Srinivasan¹, Parth Shah¹, Sarah
4 Preum³, Jason Pettus¹, Louis Vaickus¹, Joshua Levy^{1,4,5,6,7,8,**}

- 5
- 6 1. Department of Pathology and Laboratory Medicine, Dartmouth Health, Lebanon, NH
7 03766 USA
 - 8 2. Department of Computer Science, Middlebury College, Middlebury, VT 05753 USA
 - 9 3. Department of Computer Science, Dartmouth College, Hanover NH 03756 USA
 - 10 4. Department of Dermatology, Dartmouth Health, Lebanon, NH 03766 USA
 - 11 5. Department of Epidemiology, Dartmouth Geisel School of Medicine, Lebanon, NH 03766
12 USA
 - 13 6. Program in Quantitative Biomedical Sciences, Dartmouth Geisel School of Medicine,
14 Lebanon, NH 03766 USA
 - 15 7. Department of Pathology and Laboratory Medicine, Cedars Sinai Medical Center, Los
16 Angeles, CA 90048 USA
 - 17 8. Department of Computational Biomedicine, Cedars Sinai Medical Center, Los Angeles,
18 CA 90048 USA

19

20 * Authors contributed equally as co-first authors.

21 ** To whom correspondence should be addressed: joshua.j.levy@dartmouth.edu

22

23 **Corresponding Author Contact Information:**

24 Joshua J. Levy PhD

25 Assistant Professor of Pathology and Dermatology

26 Adjunct Assistant Professor of Epidemiology

27 Faculty, Quantitative Biomedical Sciences

28 Machine Learning Co-Director, Emerging Diagnostic and Investigative Technologies

29 Biostatistics and Bioinformatics Shared Resource, Dartmouth Cancer Center

30 Dartmouth-Hitchcock Medical Center

31 1 Medical Center Drive, Department of Pathology and Laboratory Medicine, Lebanon, NH

32 03756

33 Phone: (925) 457-5752 | Email: joshua.j.levy@dartmouth.edu

34

35 **Abstract**

36 Pathology reports contain vital information, yet a significant portion of this data remains
37 underutilized in electronic medical record systems due to the unstructured and varied nature of
38 reporting. Although synoptic reporting has introduced reporting standards, the majority of
39 pathology text remains free-form, necessitating additional processing to enable accessibility for
40 research and clinical applications. This paper presents Dendrite, a web application designed to
41 enhance pathology research by providing intelligent search capabilities and streamlining the
42 creation of study cohorts. Leveraging expert knowledge and natural language processing
43 algorithms, Dendrite converts free-form pathology reports into structured formats, facilitating
44 easier querying and analysis. Using a custom Python script, Dendrite organizes pathology
45 report data, enabling record linkages, text searches, and structured drop-down menus for
46 information filtering and integration. A companion web application enables data exploration and
47 export, showcasing its potential for further analysis and research. Dendrite, derived from
48 existing laboratory information systems, outperforms existing implementations in terms of
49 speed, responsiveness, and flexibility. With its efficient search functionality and support for
50 clinical research and quality improvement efforts in the pathology field, Dendrite proves to be a
51 valuable tool for pathologists. Future enhancements encompass user management integration,
52 integration of natural language processing and machine learning to enhance structured
53 reporting capabilities and seamless integration of Dendrite with the vast repository of genomics
54 and imaging data.

55

56 **Keywords:** natural language processing; laboratory information systems; bladder cancer; colon
57 cancer; pathology reports

58

59 Introduction

60 Pathology, derived from the word *pathos*, is the study of disease. Disease progression is largely
61 characterized by the use of increasingly sophisticated histological and molecular assays.
62 Pathological examination of these assays plays a critical role in diagnosis, prognostication, and
63 treatment, potentially enhancing prevention and screening efforts¹. For instance, histological
64 and molecular information often serves as the baseline and endpoint of drug trials for clinical
65 oncology applications². Pathology reports play a crucial role in capturing the clinical narrative
66 reflecting these assessments, encompassing vital information related to diagnosis, prognosis,
67 and specimen processing. Traditional approaches in natural language processing (NLP) have
68 employed rule-based or machine-learning analytics to extract valuable insights from textual
69 patterns contained in these reports, enabling clinical endpoints and biomarker information to be
70 derived from these reports³.

71
72 Leveraging the wealth of information in pathology reports for large-scale databases is costly and
73 labor-intensive, but if harnessed, has the potential to contribute significantly to comprehensive
74 cancer registries, enabling more precise population-level studies and identification of novel
75 associations between clinical features and outcomes. The subjective nature and use of
76 nonstandard mapping terminology in anatomic pathology reports limit their interoperability with
77 electronic medical record (EMR) systems— many efforts have been taken to further structure this
78 information into digestible queryable formats such as the migration of synoptic reporting
79 information into EPIC Beaker^{4,5}.

80
81 Structuring this data into more standardized formats has the potential to improve the utilization
82 of pathological reporting information. Such resources could be readily pooled into larger
83 electronic health record systems such as EPIC, which could spur future research/clinical
84 applications. Currently, many clinical research studies still rely on manual chart review, which is
85 often inefficient and prone to error. To overcome these challenges, there is a growing need for
86 advanced database management tools that can efficiently store, manage, and analyze
87 pathology data for research purposes. The use of database resources can enhance the
88 efficiency of clinical research and quality improvement planning. Collaborating with a domain
89 expert pathologist in developing these databases can ensure that the captured information is
90 pertinent and applicable to the work of various diagnostic subspecialties. Furthermore, several
91 recent studies have sought to leverage restructured pathology text reports to perform various
92 classification (e.g., study case complexity for reimbursement and RVUs) and information
93 extraction tasks (e.g., extracting gross/histo-morphology). Many of these algorithms operate on
94 unstructured free text^{6,7}.

95
96 Our pathology department at a mid-sized academic center, is transitioning from an Oracle
97 Cerner laboratory information system (LIS) to EPIC Beaker LIS. We were motivated by previous
98 attempts to create business intelligence systems that could swiftly retrieve structured and
99 unstructured pathology reporting data⁸. As a result, we designed our own in-house solution with
100 enhanced search capabilities, called Dendrite. Dendrite leverages an internal pathology
101 reporting database structured from pathology notes.

102

103 Dendrite refers to both the pathology reporting database, the codebase used to generate this
104 database as well as its front facing web interface. This database encompasses nearly 1 million
105 pathology reports, extracted from 2008 to 2022 across various diagnostic subspecialties.
106 Dendrite, designed by pathologists for pathologists, employs traditional extraction methods
107 based on expert domain knowledge to capture reporting information such as staining results
108 and procedural codes along with nearly one hundred additional reporting fields. Dendrite allows
109 for the arbitrary combination of multiple search criteria spanning nearly all facets of the
110 pathology search, with capabilities similar to systems like Cerner, allowing for the rapid display
111 of disparate information sources. The platform was developed to enable a real-time human-
112 computer web application interface that supports interactive search, filtering and aggregation.
113 Reports can be viewed through an output display table that can be edited in real-time and
114 exported. Dendrite has been integrated with the Dartmouth Cloud or Augmet, an AWS cloud
115 resource that seamlessly integrates with the electronic health record (EHR) system⁹. This
116 integration has further improved Dendrite's capabilities for imaging and genomics applications,
117 enabling swift querying and exporting of genomics and imaging data in addition to a wide array
118 of pathology reporting fields.

119
120 In the future, Dendrite can be used to supplement innovative cancer screening and surveillance
121 approaches, enhance patient outcomes, facilitate the rapid prototyping and development of
122 advanced machine learning algorithms for text, genomics, and imaging, and broaden our
123 understanding of cancer epidemiology¹⁰⁻¹². The aim of this study is to demonstrate the
124 functionality of this search tool. Subsequent studies will focus on documenting its
125 implementation.
126

127 **Methods**

128 **Data Collection and Development/Description of Dendrite Database**

129 After IRB approval, we developed a set of custom Python scripts to process ten years worth of
130 pathology reporting data, corresponding to 749,136 reports from the end of 2011 to 2021. We
131 adopted an extract, transform, load (ETL) framework that used regex for report deidentification
132 and stain dictionaries for the accurate reporting of 738 stains. It also parsed and delineated
133 different report sections (e.g., dividing diagnostic text by specimen source and free text).
134 Reporting information was reorganized into the following tables:

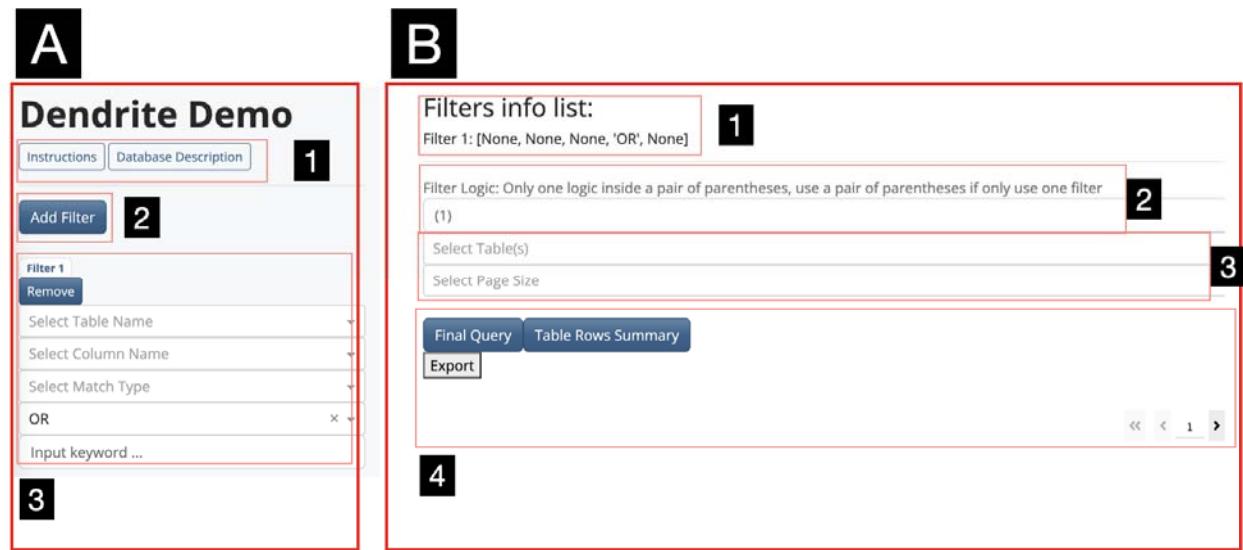
- 135
- 136 ● Stain orders and results
 - 137 ● Slide imaging information
 - 138 ● Cytology preparation methods
 - 139 ● Cytological findings
 - 140 ● Parsed and free diagnostic text
 - 141 ● Parsed and free discussion text
 - 142 ● Flow cytometry results
 - 143 ● Synoptics reporting
 - 144 ● Clinical history

- 145 • HPV results (cytology)
- 146 • Specimen adequacy
- 147 • Gross specimen reports
- 148 • Molecular findings
- 149 • Additional relevant anatomic pathology information (e.g., practicing pathologist, outside
- 150 consult, EDTA, subspecialty identifiers, addendum, etc.)

151
152 **Supplementary Table 1** provides the schema / reporting fields for each table. This information
153 can be filtered and joined through record linkages, text searches and structured drop-down
154 menus (e.g., subsetting by stain). The dendrite database can be readily updated through
155 automatic export of Cerner stored pathology reports and passing these reports through the
156 Python script. All reporting fields were deidentified using the Python script that uses regex to
157 remove information using a database of tens of thousands of first and last names. Dates were
158 stripped although a custom index date was stored which allows for the recovery of this
159 information.

160
161 This pathology database serves as the backend for an accompanying web-based tool that can
162 query this information. A web application was developed using Plotly Dash ¹³, a Python tool that
163 generates interactive websites– this interactive web application allows for collaborating
164 pathologists to explore this database through multiple search tables, to be described in
165 subsequent sections. We hosted this database and accompanying web application using an
166 Amazon Web Services (AWS) instance for internal access.

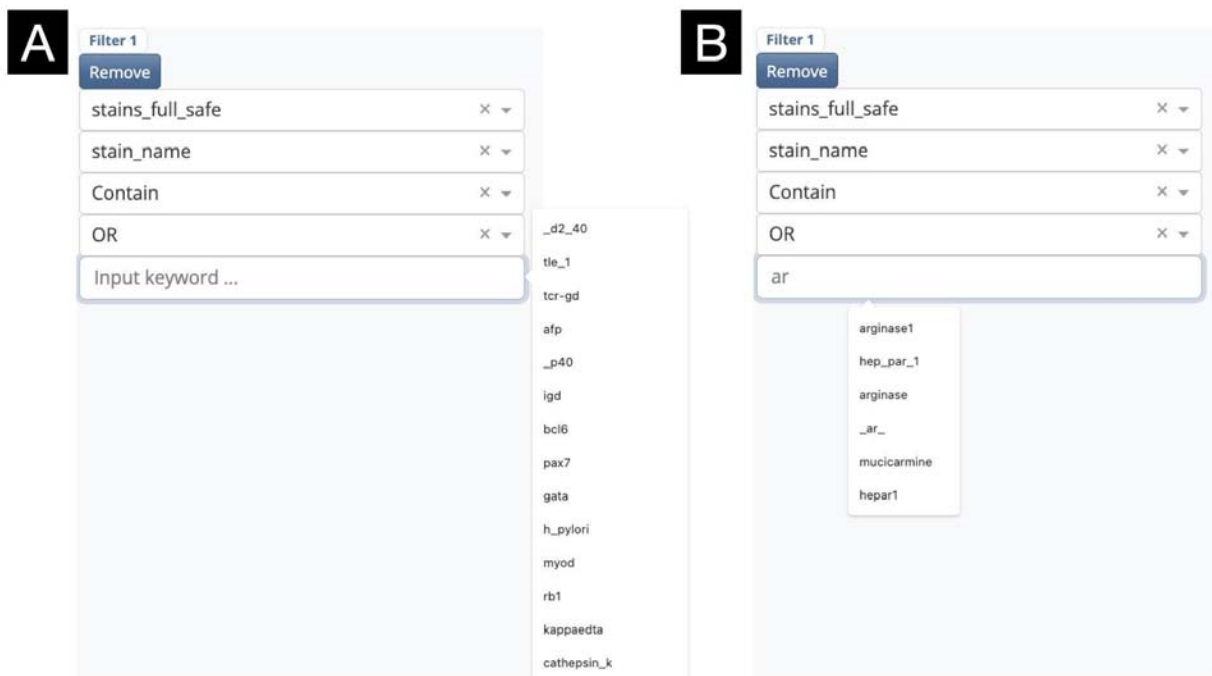
167 Web Application Description



168
169 **Figure 1: Dendrite Initial Interface: A)** Filters component. (1) Instruction button and database
170 description button. (2) Adding filters button. (3) Filter unit. **B)** Query component. (1) Filter list
171 information display. (2) Logic statement input. (3) Target table(s) selection and page size
172 selection. (4) Table query, information summary and display.

173 Multiple Search Criteria

174 The Dendrite web application layout was inspired by the web design of NCBI's search tool and
175 is divided into two sections (**Figure 1**): a) a tool for constructing filters to query the database (left
176 panel) and b) a tool to explore the queried results and browse/export rapidly compiled
177 information tables (right panel). The left panel is vertically scrollable to allow the addition of
178 multiple filters while the right panel is static. Several buttons have been added which provide
179 detailed instructions (a dynamically extending instructions tab on the right-hand side for
180 additional usage details) for operating the application and providing further description on the
181 database via a popup (**Figure 1A**). Below the instruction buttons are various buttons which
182 control the addition of report database filters (i.e., Add/Remove filter) and the subunits of the
183 filter, which allow for the selection of reporting criteria from multiple search tables (**Figure 1A**).
184 By clicking on the Add Filter, physicians can add multiple filter units in order to build multiple
185 search conditions.



186
187 **Figure 2: Keyword Input Prompts: (A) Initial prompts. (B) Prompts automatically refined**
188 **through user input**

189
190 *Description of Filter:* Filters are specified by the user to query a *table* (e.g., all stains contained
191 in stains_full_safe) for a specific field (e.g., stains as selected from stain_name) (**Figure 2**). The
192 user is then prompted via an *input keyword* argument to enter free text to query the table field
193 with (e.g., searching for carcinoma) should the field contain free text, else the user can select
194 from a list of unique values from a dropdown menu that dynamically suggests these values
195 based on an initial input. Physicians can either choose an individual term from a dropdown
196 menu or input any keyword. Dendrite will then continue to use this keyword to generate further
197 accurate suggestions. Physicians can, again, choose one of the suggestions or input any
198 additional keywords to yield the final result. Furthermore, users can select whether query text
199 from the selected field should exactly match or contain the keyword (the latter option produces

200 more flexible searches at the expense of specificity). The creation of a filter will select all
201 associated pathology reports by unique identifiers.

202 Adjustable Search Logic

203 Filters are indexed (i.e., assigned a number) by the order they are added and can be removed
204 to delete the filter. Additional functionality is provided to combine the search results from
205 multiple tables. This is provided through a logical dropdown menu, with the following options: 1)
206 “AND”– the intersection of reports from the current and previously defined filter, 2) “OR”– the
207 union of reports from the current and previous filter, and 3) “NOT”– the set difference of reports
208 from the current and previous filters. A list of available filters is made available to the user with
209 along an editable logic statement that specifies the combined search criteria, generated from
210 the initial logical dropdown menus (e.g., ((1 AND 2) OR 3) represents the union between filter 3
211 and the intersection between filters 1 and 2). This logical statement can be adjusted based on
212 any possible combination/permutation of filters– for instance, ((1 AND 3) OR (2 AND 4)),
213 representing the union between the intersection between filters 1 and 3 and separately the
214 intersection between filters 2 and 4. Once the final conditional logic has been specified,
215 physicians can select one or more target table(s) for viewing. Running the final query will
216 generate the final set of pathology reports selected using the conditional logic states and merge
217 together the selected tables by these unique identifiers. Further description of the search
218 process can be found in the supplementary materials (**Supplementary Figure 1**).

219 Interactive Result Table

220 The final table display is visualized in the lower right of the web application, generated after
221 running the query (**Figure 3**). The number of unique patients / reports can be found using the
222 “Table Rows Summary” button. The final table is organized into multiple pages. Sorting and
223 filtering operations, similar to that featured in an excel spreadsheet, can be used to further sort,
224 query or subset the table by multiple columns. All the results can be exported/downloaded into
225 multiple spreadsheet formats for further analysis (**Supplementary Figure 2**).

226

The screenshot displays the 'Dendrite Demo' web application. On the left, a sidebar contains an 'Add Filter' button and a list of filters: 'discussions_safe', 'ds_txt', 'Contain', 'OR', and 'specimen'. A 'Filter' label is overlaid on the sidebar. The main content area shows 'Filters info list:' with 'Filter 1: ['discussions_safe', 'ds_txt', 'Contain', 'OR', 'specimen']'. Below this, the 'Filter Logic' is defined as '(1) stains_full_safe' with a value of '5'. A 'Final Query' button is visible. The 'Table Rows Summary' button is also present. A 'Sorting' label is overlaid on the table header. The table has columns: 'id_safe', 'block_number', 'stain_name', and 'description'. The data rows are as follows:

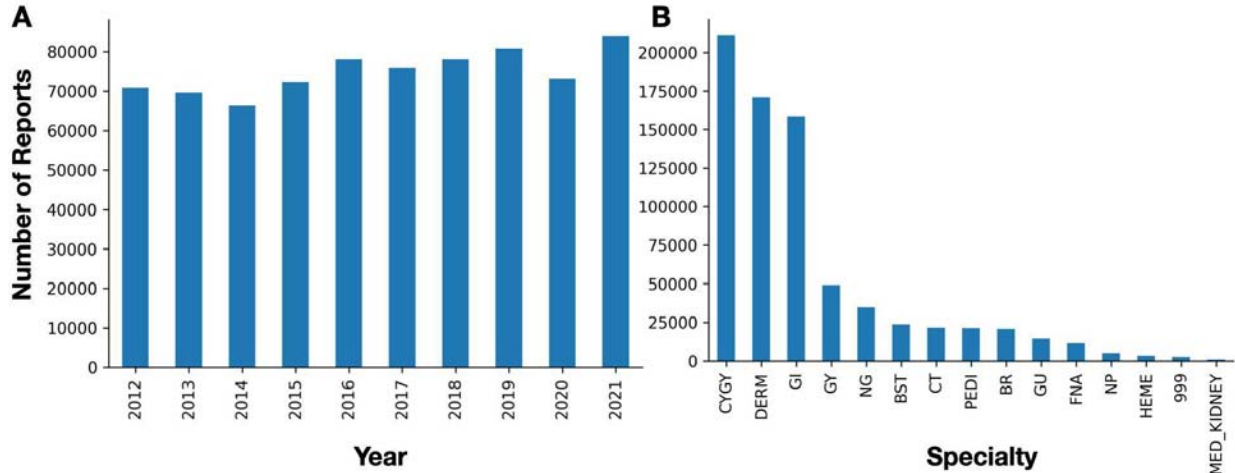
id_safe	block_number	stain_name	description
ec91d046-ee62-11ec-9e49-8c8caa4dd5eb	a1	_cd3_	see above
ec91d046-ee62-11ec-9e49-8c8caa4dd5eb	a1	cd20	
ec91d046-ee62-11ec-9e49-8c8caa4dd5eb	a1	bc12	
ec91d046-ee62-11ec-9e49-8c8caa4dd5eb	a1	bc16	
ec91d046-ee62-11ec-9e49-8c8caa4dd5eb	a1	cyclin_d1	

227

228

229 **Figure 3: Illustration of filtering and sorting functionality in results display table**

230 **Results**



231 **Figure 4: Breakdown of number of included pathology reports by: A) Year, B) Subspecialty**

232

233

234

Description of Select Database Tables

235

236 A total of 749,136 reports were extracted from December 2011 to December 2021 (Figure 4),
 237 corresponding to 272,714 patients, assessed across 13 diagnostic subspecialties from 79
 238 pathologists. Of these cases, 46,846 required intradepartmental consult (extraction of “qacc”
 239 from report), leading to a total of 2,764 consensus conferences (extraction of “conf”) and
 240 discussion amongst pathologists were reported for 3,330 reports (“dw_doc” reporting signature).
 241 Synoptic reporting was identified in 9,700 reports, though the availability of this information
 242 continues to grow. Addendums were found for 26,026 reports. Staining results were reported
 243 across 165,140 specimen blocks, and reports were linked to a total of 388,679 whole slide
 244 images. Clinical history, nuanced molecular findings (e.g., MLH1 hypermethylation), and flow
 245 cytometry findings were reported for 494,122, 162,059, and 3,065 reports respectively.

User Tests

246

247

248

249

250

251

252

253

254

255

256

As an initial test of Dendrite’s functionality, we asked collaborating pathologists to perform a series of example queries. Future works will explore the impact this database system has on facilitating timely review by assessing widespread usage across the department but was outside of the scope of this technical note.

Urine Specimen Atypia: A cytopathologist conducted a search to locate voided urine cytology specimens to assess instances of specimen atypia for the assessment of high-grade urothelial carcinoma. First, cases were filtered using the non-gynecological specimens table (*ng_safe*), searching for voided urine in the specimen *source* field (Figure 5). While we were able to identify a number of specimens using this search functionality, ultimately we opted to use the

257 diagnostic text (*diagnoses_safe*), searching for voided specimens, as we felt the free text was
 258 less restrictive (**Figure 6**). By specifying a search for voided specimens, we were able to
 259 remove washes and upper tract specimens. This search yielded approximately 4,178 reports,
 260 close to the number of voided urine specimens reported over the data collection period. We
 261 queried the diagnostic tables and merged these with a table pertaining to additional pathology
 262 reporting information, including sign-out time. This table was exported to a CSV format and
 263 further processed using a custom R script (using the “grep” function) to yield the number of
 264 negative, atypical, suspicious and positive cases over time. We plotted the incidence of these
 265 diagnostic categories over time along with a barplot depicting the overall categorization in
 266 **Figure 7**, demonstrating that in less than a minute, we could conduct a longitudinal study of
 267 urine specimen atypia to inform rapid bladder cancer screening. For instance, by plotting the
 268 incidence of different diagnoses, we found that the number of negative findings increased from
 269 2016 to 2018, while the number of positive findings decreased from 2016 to 2018. As our
 270 department implemented The Paris System for Reporting Urine Cytology in 2018¹⁴, these
 271 findings corroborate with previous research indicating early adoption of these reporting
 272 guidelines. Further analysis was conducted using hierarchical Bayesian regression modeling
 273 (utilizing the *brms* and *emmeans* R packages)^{15–17}. In this model, we treated the level of
 274 implementation—pre-publication (before 2016), publication of the guidelines prior to its actual
 275 implementation (2016-2016), and post-implementation (after 2018)—as an ordinal variable with
 276 monotonic effects. Incorporating pathologist-level random intercepts, we identified a reduction in
 277 specimens deemed atypical across the implementation stages, as detailed in **Table 1**.
 278
 279

Dendrite Demo

Instructions Database Description

Add Filter

Filter 1
Remove

- cy_ng_safe ×
- source ×
- Contain ×
- OR ×
- urine|

Filters info list:
Filter 1: ['cy_ng_safe', 'source', 'Contain', 'OR', 'urine ']

Filter Logic: Only one logic inside a pair of parentheses, use a pair of parentheses if only use one filter

(1)

Select Table(s) ▾

Select Page Size ▾

Final Query Table Rows Summary

Export

A: Urine, Voided B: Specimen is submitted to Mayo Medical Laboratori

Ureter: left (cystoscopic urine)

Urine, Cystoscopic (right renal pelvic urine)

Urine, ileal conduit (from clean pouch)

Urine, ileal Conduit cathed

Urine, Cystoscopic, Left renal

Urine (Left Nephrostomy Tube)

Pelvic urine, right (washing)

Urine, Voided (voided kidney stone)

Urine, catheterized (Foley tube)

Urine (right renal wash)

<< < 1 > >>

280
 281 **Figure 5: Querying non-gynecological table for voided urine specimens**
 282

Dendrite Demo

Instructions Database Description

Add Filter

Filter 1
Remove

diagnoses_safe x

spc_from_dx x

Contain x

OR x

void

Filters info list:

Filter 1: ['diagnoses_safe', 'spc_from_dx', 'Contain', 'OR', 'void']

Filter Logic: Only one logic inside a pair of parentheses, use a pair of parentheses if only use one filter

(1)

x diagnoses_safe x

20 x

Final Query Table Rows Summary ~ 4178 rows, 4178 unique ids, 2740 unique patient ids -

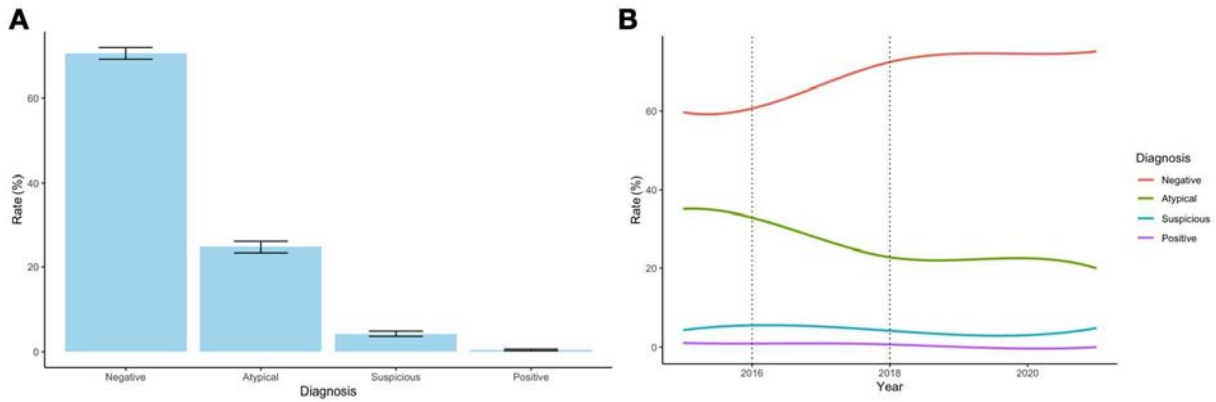
Export

id_safe	spc_from_dx	dx_txt_no_spc
000184e0-ee63-11ec-9e49-8c8caa4dd5eb	Urine, voided:	999 ** Negative for High Grade Urot
000314e3-ee63-11ec-9e49-8c8caa4dd5eb	Urine, voided:	999 ** Atypical Urothelial Cell
0003d966-ee63-11ec-9e49-8c8caa4dd5eb	Urine, Voided:	999 ** Negati
001722c5-ee63-11ec-9e49-8c8caa4dd5eb	Urine, voided:	999 ** Negative for High Grade Urot
00180611-ee63-11ec-9e49-8c8caa4dd5eb	Urine, voided:	999 ** Negative for High Grade Urot
001b12cb-ee63-11ec-9e49-8c8caa4dd5eb	Urine, voided:	999 ** Atypical Urothelial Cell
001eecd8-ee63-11ec-9e49-8c8caa4dd5eb	Urine, voided:	999 ** Negati
0023b27d-ee63-11ec-9e49-8c8caa4dd5eb	Urine, voided:	999 ** Negative for High Grade Urot
00291e8d-ee63-11ec-9e49-8c8caa4dd5eb	Urine, voided:	999 ** Negative for High Grade Urot
00363bc2-ee63-11ec-9e49-8c8caa4dd5eb	Urine, voided:	999
00417db0-ee63-11ec-9e49-8c8caa4dd5eb	Urine, voided:	999 ** Atypical Urothelial Cell
00462303-ee63-11ec-9e49-8c8caa4dd5eb	Urine, voided:	999 ** Negati

283

284

Figure 6: Querying diagnostic text for voided urine specimens



285

286

287

288

289

290

Figure 7: Breakdown of The Paris System diagnostic assignments: A) Aggregated across the entire database reporting period, and B) Yearly over time across the reporting period; vertical lines indicate publication of the official TPS guidelines (2016) followed by official implementation of the guidelines in 2018

Table 1: Regression Coefficients from Statistical Model Comparing Atypia Rates between Various Implementation Periods for The Paris System: Odds ratio above one indicates that atypia rates increased over time whereas below one indicates reductions in reported atypia compared to the other TPS categories

Parameter	Odds Ratio	2.5% CI	97.5% CI	p-value
Degree of Implementation: Post-Implementation > Post-Publication > Pre-Publication	0.90	0.82	0.97	0.01
Post-Implementation vs. Post-Publication	0.94	0.80	1.00	0.01

291

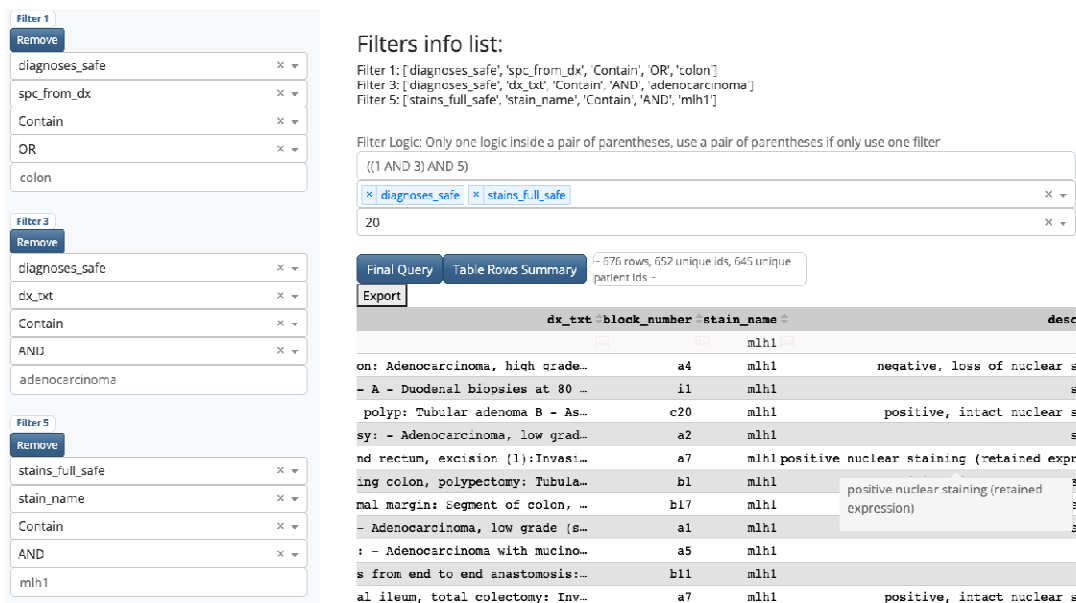
292

293

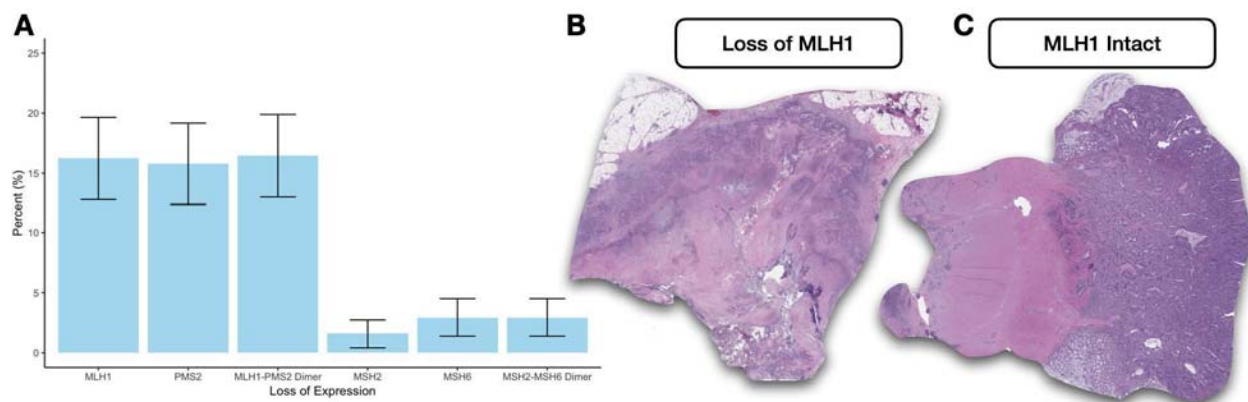
294

Post-Implementation vs. Pre-Publication	0.80	0.68	0.95	0.01
Post-Publication vs. Pre-Publication	0.87	0.72	0.99	0.01

295
 296 *Colon adenocarcinoma cases:* A GI pathologist attempted our second search. We sought to pull
 297 MLH1 staining results from colon adenocarcinoma cases to assess for mismatch repair
 298 deficiency to build a retrospective cohort that assesses the metastatic potential of this subgroup
 299 in addition to pulling corresponding whole slide images for a digital image analysis. To do this,
 300 three filters were constructed– the first two from the diagnostic text tables and the final one from
 301 the staining results tables. We first filtered by whether the text contained “colon” and
 302 “adenocarcinoma”. The final filter sought to query for cases where MLH1 staining had been
 303 done. Multiple stains were ordered for specific cases. The interactive filtering mechanisms of the
 304 final display table were used to subset stains by MLH1 status. For the final query, this table was
 305 merged with a whole slide image table, which specifies the file locations of all matching whole
 306 slide images for tissue slides stained with hematoxylin and eosin within our Aperio Image
 307 server. This search yielded approximately 676 pathology reports (**Figure 8**). We conducted
 308 additional filtering for other mismatch repair related genes (MLH1/PMS2 and MSH2/MSH6 form
 309 two separate heterodimers with different progression characteristics but indicative of the
 310 microsatellite instability pathway). Results were exported to a CSV file. The CSV file contained
 311 information on staining results and the file locations of whole slide images within our Aperio
 312 image server. Staining results were further processed using a custom R script to generate a bar
 313 plot of negative and positive staining findings, identifying significantly higher instances of loss of
 314 expression for the MLH1/PMS2 heterodimer as assessed through immunohistochemistry
 315 (**Figure 9A**). This information was also used to pull the corresponding whole slide images of
 316 H&E stained tissue (**Figure 9B**).
 317



318
 319 **Figure 8: Query results of diagnostic text and staining information for Colon**
 320 **Adenocarcinoma cases for potential mismatch repair deficiency**
 321



322
 323 **Figure 9: Further analysis of extracted Dendrite table for studying colon cancer**
 324 **mismatch repair deficiency: A)** Breakdown of loss of expression for mismatch repair genes as
 325 assessed through immunohistochemistry; **B)** Whole slide images extracted using Dendrite
 326 image file location information illustrating slides with and without MLH1 positive staining
 327

328 *Qualitative description of the pathologist experience using Dendrite:* From discussions with
 329 pathologists in our department, we found that compared to the search functionality from our
 330 EMR (Cerner), the pathologists found Dendrite to be faster, more responsive and flexible.
 331 Pathologists were able to finalize search criteria across hundreds of thousands of cases and
 332 merge information from disparate data sources in seconds. The secondary search / filtering
 333 capabilities offered using the interactive display table were found to be particularly effective in
 334 further refining the search results.

335 Discussion

336 By leveraging expert domain knowledge and natural language processing algorithms to parse
 337 free-text pathology reports into structured multi-table formats, Dendrite presents a viable
 338 database tool that pathologists can use to rapidly build study cohorts and leverage for quality
 339 improvement purposes to complement other emerging EMR systems.
 340

341 While we have implemented an initial prototype of the Dendrite web application and deployed
 342 this tool over AWS, there are many areas where this tool can be improved. First, we plan to
 343 develop and expand a user management system, which will allow for user logins and permit
 344 tracking of user data¹⁸. This will help us determine the broad utilization of this tool within the
 345 department and areas to improve this application for further widespread adoption. Users will be
 346 able to access their search history offline to help guide more informed searches as users gain
 347 more familiarity with the system. Leveraging the users' search history can also help personalize
 348 further searches based on their preferences (e.g., accurate suggestions and operations). For
 349 example, if some users are accustomed to querying data about colon cancer, then the keywords
 350 related to colon cancer can be suggested through the integration of knowledge and semantic
 351 databases which link biological entities. Natural language processing algorithms can continue to
 352 improve their suggestions and complex instructions through the accumulation of more data and
 353 user feedback.

354
355 Furthermore, we plan to further integrate NLP (natural language processing) and machine
356 learning technologies into this web application in order to mine free text data to extract
357 structured reporting information that may be more readily queryable. This will be accomplished
358 by configuring multiple machine learning algorithms to make predictions across multiple tasks,
359 including but not limited to 1) CPT code prediction to identify instances of underbilling¹⁹, 2)
360 named entity recognition tasks (e.g., extraction of staining information, histological findings)²⁰,
361 and 3) instances where an outside consult is needed, etc. Trained predictive algorithms can be
362 further integrated into future iterations of this web application. Large language models (LLMs)
363 offer a promising approach by capturing nuanced contexts and relationships. We acknowledge
364 that the reporting information is still incomplete and biased towards certain subspecialties, and
365 we are aware that more work is needed to extract information from less common or structured
366 specialties. This database features record linkages to imaging data. Further integration of our
367 text database into our genomics and imaging infrastructure can enhance its search capabilities
368 as well as motivate future multimodal analysis²¹.

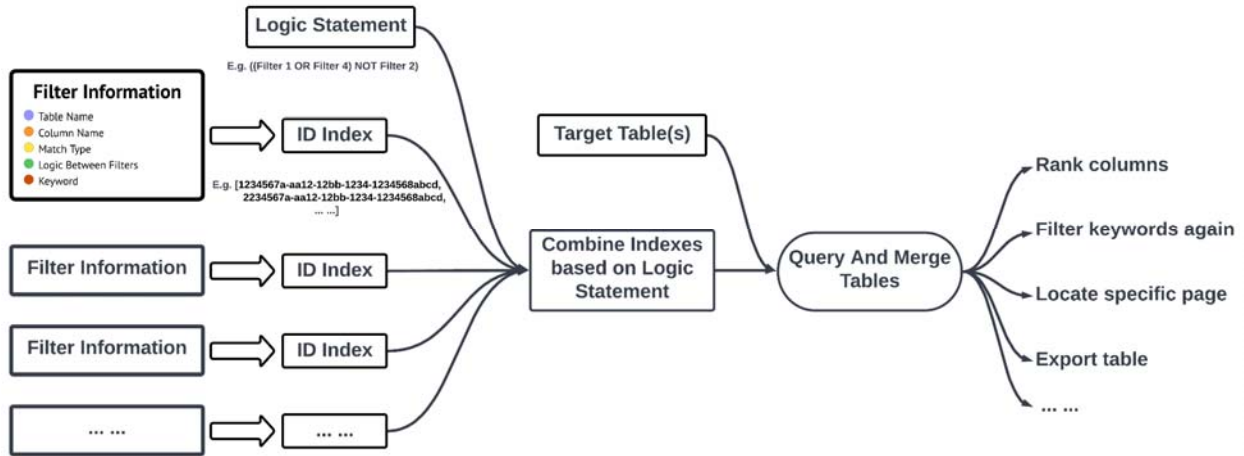
369 **Conclusion**

370 Dendrite, a web application developed using natural language processing and expert
371 knowledge, offers pathologists a flexible search capability for conducting large-scale
372 assessments of clinical pathology reports in the context of clinical research and quality
373 improvement. The application employs various features to enable efficient search functionality,
374 including filters that can be combined using conditional logic statements, comprehensive
375 merging of reporting tables associated with selected pathology reports, and dynamic sorting and
376 filtering operations on the display table. These features greatly facilitate the search process for
377 pathology data, making it easier and more convenient for pathologists in their clinical and
378 research endeavors.

379 **References**

- 380 1. Berho, M. & Bejarano, P. A. Judging pathological assessment in cancer specimens. *Journal*
381 *of Surgical Oncology* **110**, 543–550 (2014).
- 382 2. Nagtegaal, I. D., West, N. P., van Krieken, J. H. J. & Quirke, P. Pathology is a necessary
383 and informative tool in oncology clinical trials. *The Journal of Pathology* **232**, 185–189
384 (2014).
- 385 3. Hirschberg, J. & Manning, C. D. Advances in natural language processing. *Science* **349**,
386 261–266 (2015).
- 387 4. Daniel, C. *et al.* Standards to Support Information Systems Integration in Anatomic
388 Pathology. *Archives of Pathology & Laboratory Medicine* **133**, 1841–1849 (2009).
- 389 5. Krasowski, M. D. *et al.* Implementation of Epic Beaker Clinical Pathology at an academic
390 medical center. *Journal of Pathology Informatics* **7**, 7 (2016).
- 391 6. Allada, A. K. *et al.* Analysis of Language Embeddings for Classification of Unstructured
392 Pathology Reports. in *2021 43rd Annual International Conference of the IEEE Engineering*
393 *in Medicine & Biology Society (EMBC)* 2378–2381 (2021).
394 doi:10.1109/EMBC46164.2021.9630347.

- 395 7. Schadow, G. & McDonald, C. J. Extracting Structured Information from Free Text Pathology
396 Reports. *AMIA Annu Symp Proc* **2003**, 584–588 (2003).
- 397 8. Arvisais-Anhalt, S. *et al.* Searching Full-Text Anatomic Pathology Reports Using Business
398 Intelligence Software. *Journal of Pathology Informatics* **13**, 100014 (2022).
- 399 9. Green, D. *et al.* Clinical Implementation of a Robust Cloud-Based Architecture for the
400 Analysis of Somatic Whole Exome Sequencing Data. in *JOURNAL OF MOLECULAR*
401 *DIAGNOSTICS* vol. 24 S86–S86 (ELSEVIER SCIENCE INC STE 800, 230 PARK AVE,
402 NEW YORK, NY 10169 USA, 2022).
- 403 10. Wender, R. C., Brawley, O. W., Fedewa, S. A., Gansler, T. & Smith, R. A. A blueprint for
404 cancer screening and early detection: Advancing screening's contribution to cancer control.
405 *CA: A Cancer Journal for Clinicians* **69**, 50–79 (2019).
- 406 11. Greenburg, J. *et al.* Development of an interactive web dashboard to facilitate the
407 reexamination of pathology reports for instances of underbilling of CPT codes. *Journal of*
408 *Pathology Informatics* **14**, 100187 (2023).
- 409 12. Lantada, A. D. & Morgado, P. L. Rapid Prototyping for Biomedical Engineering: Current
410 Capabilities and Challenges. *Annual Review of Biomedical Engineering* **14**, 73–96 (2012).
- 411 13. Dabbas, E. *Interactive Dashboards and Data Apps with Plotly and Dash: Harness the power*
412 *of a fully fledged frontend web framework in Python–no JavaScript required.* (Packt
413 Publishing Ltd, 2021).
- 414 14. Levy, J. J. *et al.* Large-scale longitudinal comparison of urine cytological classification
415 systems reveals potential early adoption of The Paris System criteria. *J Am Soc Cytopathol*
416 *S2213-2945(22)00241–1* (2022) doi:10.1016/j.jasc.2022.08.001.
- 417 15. Bürkner, P.-C. Advanced Bayesian Multilevel Modeling with the R Package brms. *The R*
418 *Journal* **10**, 395–411 (2018).
- 419 16. Bürkner, P.-C. & Charpentier, E. Modelling monotonic effects of ordinal predictors in
420 Bayesian regression models. *British Journal of Mathematical and Statistical Psychology*
421 **n/a**.
- 422 17. Lenth, R. V. *et al.* emmeans: Estimated Marginal Means, aka Least-Squares Means. (2023).
- 423 18. Kim, J. Y., Gudewicz, T. M., Dighe, A. S. & Gilbertson, J. R. The pathology informatics
424 curriculum wiki: Harnessing the power of user-generated content. *J Pathol Inform* **1**, 10
425 (2010).
- 426 19. Levy, J., Vattikonda, N., Haudenschild, C., Christensen, B. & Vaickus, L. Comparison of
427 machine-learning algorithms for the prediction of current procedural terminology (CPT)
428 codes from pathology reports. *Journal of Pathology Informatics* **13**, 3 (2022).
- 429 20. Oliwa, T. *et al.* Obtaining Knowledge in Pathology Reports Through a Natural Language
430 Processing Approach With Classification, Named-Entity Recognition, and Relation-
431 Extraction Heuristics. *JCO clinical cancer informatics* **3**, 1–8 (2019).
- 432 21. Lu, M. Y. *et al.* Towards a Visual-Language Foundation Model for Computational Pathology.
433 Preprint at <https://doi.org/10.48550/arXiv.2307.12914> (2023).
- 434
- 435



439 **Supplementary Figure 1: Illustration of table querying logic using Dendrite Web**

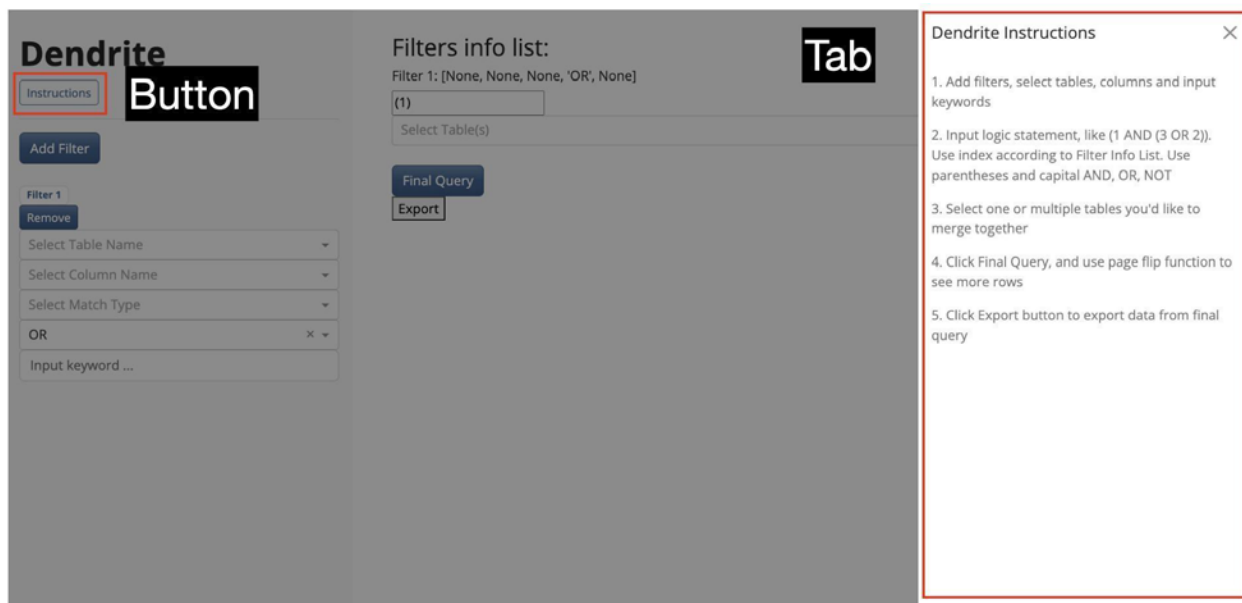
440 **Application:** Separate filters are specified by the user. Each filter returns a list of pathology
 441 report IDs. Indices are joined through intersection and union operations based on user-specified
 442 logic statements. Target tables are selected, each filtered based on the final set of report IDs.
 443 Additional table operations may be performed prior to export.

445 **Additional Information on Dendrite Web Application Text Search Workflow**

446 In the intermediate level of the Dendrite searching process, every action on the interface level
 447 will have a corresponding processing interaction in the underlying level. The first step is to
 448 collect the information from the filters, including table name, column name, match type, logic
 449 choice and keyword. It involves multiple callbacks. For example, the column name depends on
 450 the table. The column choices will only update after the table is chosen. Each filter will generate
 451 an index list from a specific table. All of the filters' information will be displayed on the right side
 452 of the screen, so that it remains visible to facilitate user experience. Filters are removable, and
 453 once removed, will be deleted from the display window with no information saved.

455 The next step is to verify the logic statement. By inputting logic choices in the filter, Dendrite will
 456 automatically generate a logic statement in the order of the filter indexes. Physicians can also
 457 edit the logic statement by adding, removing and reordering the filter indexes. Dendrite will then
 458 parse the statement and query the index in order. It will then combine the indexes based on the
 459 logic between them. For example, if the logic between two filters is "OR", Dendrite will keep all
 460 indexes. However if the logic is "AND", Dendrite will only keep the intersection of the indexes,
 461 and then form a final index list. After that Dendrite will extract the table, based on the target
 462 table(s) input by the physicians. It will retain the rows which have the same indexes as the final
 463 indexes list. And then merge these tables together to the final table. During the merging, if a
 464 missing value appears, it will be left blank. Clicking on Query will display the Table. Inside this
 465 table, physicians can further filter the keyword or perform operations. Dendrite will recognize the
 466 column content type as a string data or numeric data, and then rank it in alphabetical order or
 467 numerical order. And the table data will be saved. The generated table data will be exportable
 468 via the download option.

469
 470 Physicians can select multiple keyword-based search terms at the same time and combine
 471 these search terms in any way they want. For example, they can select any two conditions that
 472 are logically related to "AND" or "OR" or "NOT". This allow physicians to perform real-time
 473 interactive data manipulation across one or more target tables that have been merged together.
 474 Also, the application automatically recognizes the table data types, allowing it to perform
 475 different sorting operations for tables that contain character and numeric data. Physicians can
 476 filter the table further by adding filters. While taking into account the readability of the data, the
 477 data table provides page numbering and scrollable axis functionality. This allows for more
 478 precise targeting of rows and also extends the table's display range while maintaining
 479 readability.
 480



481
 482 **Supplementary Figure 2: Dendrite Instructions Tab** In case of unfamiliarity with operation of
 483 the web application, we developed an instruction tab that will appear by clicking the instruction
 484 button directly below the title. With readability in mind, we designed the instruction tab to appear
 485 by sliding from the right. This allows users to read the guide while simultaneously having access
 486 to the main dendrite content on the left, enabling immediate recognition on the interface of
 487 salient features described in the instructions.
 488

489 Additional Information on Structured Dendrite Table Schema

490
 491 **Supplementary Table 1:** Pathology database table names, column names, data types and
 492 metadata

Table Name	Column Name	Data Type	Note
ap_case	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession. Relationship to patient is a separate linking table.

	pathologist	varchar	Name of pathologist signing out case.
	qacc	tinyint	Dummy variable - Does "qacc" appear in report text? (i.e, Intradepartmental consultation sought?)
	dw_doc	tinyint	Dummy variable - Indicates whether the case describes discussion with another physician.
	conf	tinyint	Dummy variable - Does the case mention an intradepartmental or consensus conference?
	addend	tinyint	Dummy variable - Does the report contain any addenda?
	syn	tinyint	Dummy variable - Does the report have a synoptic section?
	frz	tinyint	Dummy variable - Does the report have a frozen section?
	steps	tinyint	Dummy variable - Does the report contain common ways of describing additional sections?
	decal	tinyint	Dummy variable - Does the report mention decalcification?
	edta	tinyint	Dummy variable - Does the report mention EDTA decalcification?
	ssc	varchar	Subspeciality code.
	svc	varchar	Service calculated from subspeciality code at Python script runtime.
	case_refs	int	Attempts to count the DH surgical accession number mentioned in the report text.
	indecision	int	Counts keywords related to indecision or uncertainty.
	lesser_indecision	int	Counts keywords related to lesser degrees of indecision or uncertainty.
	negation	int	Counts keywords related to negation.
	TAT	int	Calculated field - Turnaround time, measured as the difference between valid_int and sub_int.
	valid_int	int	Interval between a secret date and validation of the case. Was automatically converted back to validation date using secret key.
	sub_int	int	Interval between a secret date and submission of the case. Was automatically converted back to validation date using secret key.
case_parts : 1:1 to case	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
	case_parts	text	List of parts of the case, as described by clinical team
	clin_hxdx	text	Brief history and diagnosis as provided by clinical team.
diagnoses: 1:1 to case	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.

	spc_from_dx	text	Attempts to capture the part of diagnosis that describes the specimen.
	dx_txt_no_spc	text	Attempts to capture diagnosis with specimen removed.
	dx_txt	text	Contains the entire "diagnosis" field.
discussions: 1:1 to case	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
	ds_txt	text	Deidentified discussion text.
gross: many-to-one to case	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
	part	varchar	Part of the case to which the row belongs.
	subsection	varchar	Subsection of the gross to which the row belongs.
	field_name	text	Name of the reported field (e.g., tumor size, weight of specimen).
	field_val	text	Value of the field in text format, regardless of numeric nature.
h_and_e: many-to-one to case	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
	slide_no	varchar	Block letter and number to which the slide belongs.
	description	text	Prosector's description of tissue submitted for the slide.
pt_cases_safe	pt_id_safe	UUID in varchar	UUID in varchar; maps 1:1 to MRN and many-to-one to id_safe.
	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession and many-to-one to pt_id_safe.
scans: many-to-one to case	id_scans	int	Autoincrement; related to surgical accession.
	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
	slide_id	int	Aperio server slide id.
	block	varchar	Block (if available) stained.
	link	varchar	Calculated field - Readymade GET request to Aperio server.

stains_full: many-to-one to case	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
	block_number	varchar	Block number for the stain, if identifiable.
	stain_name	varchar	Name of the stain performed.
	result	varchar	Result of the stain performed, if identified.
stains_short: many-to-one to case	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
	stain	varchar	Name of the stain performed.
synoptics: many-to-one to case	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
	field_name	varchar	Name of synoptic report field.
	field_val	varchar	Name of synoptic report field.
	Note		Valid only for reports generated during later time periods; earlier synoptics are not as well-delineated.
flow:	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
	viability	int	Attempts to detect percent viability with regex.
	source	varchar	Specimen source.
	markers	varchar	Comma-separated list of markers tested by flow, if detected.
	flowdx	text	Classification of diagnosis section relating to flow.
	flowds	test	Classification of discussion section relating to flow.
	Note		Changes in flow cytometry reporting over the years; some aspects of cases with flow may be identified with lower precision.
cy_gyn:	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
	hpv_opt	varchar	HPV option chosen.

		ar	
	prep	varchar	Preparation type.
		ar	
	source	varchar	Cervical/endocervical/vaginal source.
		ar	
	hormones	varchar	Patient on hormone therapy?
		ar	
	hyst	varchar	History of hysterectomy?
		ar	
	post_partum	varchar	Post-partum pap smear?
		ar	
	preg	varchar	Patient pregnant during smear?
		ar	
	iud	varchar	Patient has an IUD?
		ar	
	pelvic_rads	varchar	Patient history of pelvic radiation?
		ar	
	prior_gy_tx	varchar	Prior gynecologic treatment?
		ar	
	hx_abnl_pap_bx	varchar	History of prior abnormal pap or cervix biopsy?
		ar	
	hpv_vacc	varchar	Patient received the HPV vaccine?
		ar	
	smoker	varchar	Patient smoke?
		ar	
	des	varchar	Patient history of in-utero exposure to diethylstilbestrol?
		ar	
	icd_dx	varchar	ICD10 diagnosis code (not present for many reports).
		ar	
	clin_hx_imp	text	Free text clinical history and impression.
	Note		Only cases which a pathologist has handled have matching MRNs. Many of these fields are Yes/No most of the time with occasional free-text description.
cy_hpv:	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
		ar	
	hpv_status	tinyint	HPV status: 0 negative, 1 positive.
		t	
	hpv_type	varchar	HPV type: 16, 18, or Other High Risk.
		ar	
	Note		Many-to-one to surgical accessions - patient can be positive for multiple HPV types
cy_ng: All non-GYN cases, including urines.	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
		ar	
	source	varchar	Source of specimen.

		ar	
	site	varchar	Another location description.
		ar	
	clin_hx_tx	varchar	Partially overlapping clinician's impression of case.
		ar	
	clin_hx_imp	varchar	Partially overlapping clinician's impression of case.
		ar	
	clin_imp	varchar	Partially overlapping clinician's impression of case.
		ar	
	rads_findings	varchar	Radiology findings per the clinician.
		ar	
	gross	varchar	Gross description of the specimen.
		ar	
	thy_size	varchar	Thyroid specific: Lesion size.
		ar	
	thy_echo	varchar	Thyroid specific: Echogenicity
		ar	
	thy_cyst	varchar	Thyroid specific: Cyst or not?
		ar	
	thy_calc	varchar	Thyroid specific: Calcified?
		ar	
	thy_vasc	varchar	Thyroid specific: Vascularity?
		ar	
	ia_byline	varchar	Immediate assessment physician and text.
		ar	
	Note		Prior to the creation of the FN prefix (about 20% of the dataset falls into this timeframe), also included FNAs.
cy_ia: Immediate assessment data from FNA and NG cases.	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
		ar	
	passes	int	Number of passes in immediate assessment.
	slides	int	Number of slides in immediate assessment.
	txt	text	Actual content of the immediate assessment episode.
	Note		Specific to pathologist. Each row is one assessment episode. See ia_byline in cy_ng table for the identity of the assessing pathologist, as it is extremely rare for two immediate assessments on the same case to be done by different people.
Molecular_ result:	id_safe	UUID in varchar	UUID in varchar; maps 1:1 to surgical accession.
		ar	
	mol_res_txt	text	Text of the molecular pathology result.