Jump ARCHES Data Visualization User Manual

Vincent Lam and Alexander Morain,
University of Illinois Urbana-Champaign

Published: 30 Jan 2022
Last Revision: 18 Mar 2022

jump | ARCHES

The Grainger College of Engineering
Health Care Engineering Systems Center

THE UNIVERSITY OF ILLINOIS
COLLEGE OF MEDICINE
PEORIA CHICAGO ROCKFORD URBANA

Table of Contents

Introduction

This data visualization tool is intended to provide an interactive view of Jump Applied Research in Community Health through Engineering and Simulation (ARCHES) investigators and projects. We built this visualization using Mike Bostock's D3.js, a JavaScript library for producing dynamic, interactive data visualizations in web browsers. We've used the d3-force module from this library to produce a force-directed graph layout between investigators, who have been linked to other investigators on a per-project basis to form a network of collaboration. This physics-based simulation allows the user to search for investigators, filter projects by various categories and tags, and interact with node forces. A combination of color, size, and distance allows the user to identify the most productive collaborators and a lack of projects in a particular research area. The visualization is completely rendered on the client side.

Hosting

The visualization can currently be found at https://amoslab.github.io/arches-data-visualization. The webpage and all related files are hosted on the Github repository of Amos Research Lab. The source for the github page is the gh-pages branch, which receives stable updates, in order to avoid unstable updates to the master branch.

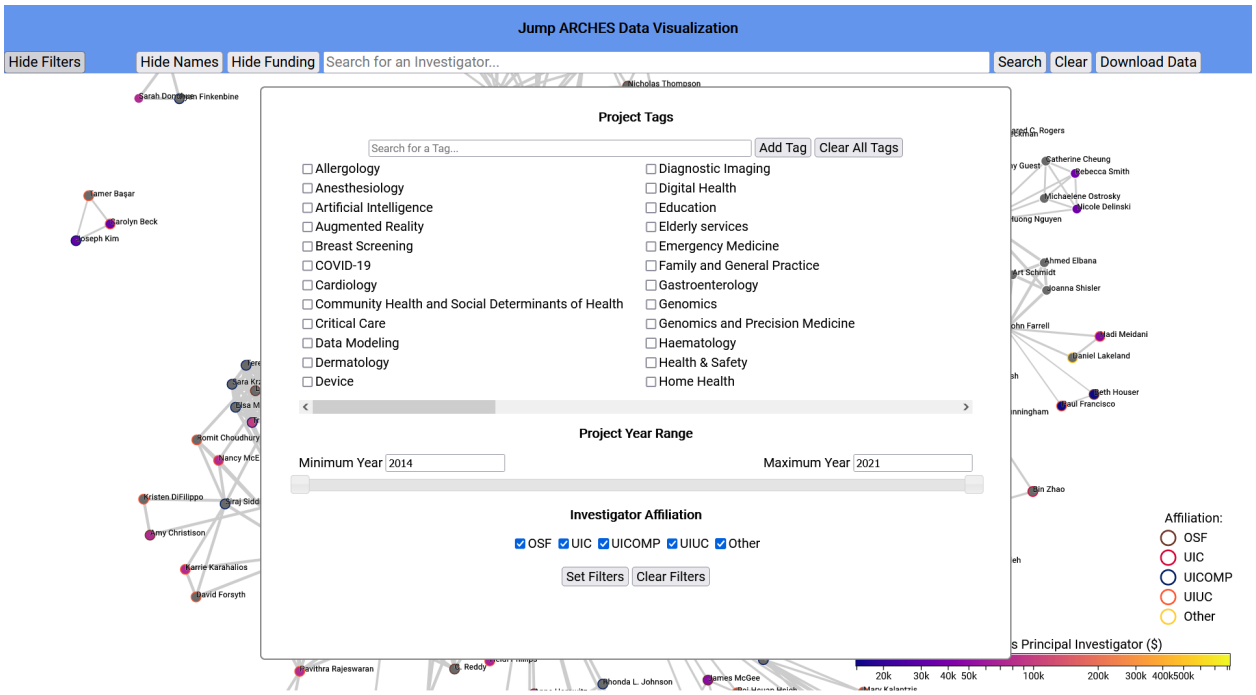
Using the Visualization Tool

Menu

Filters

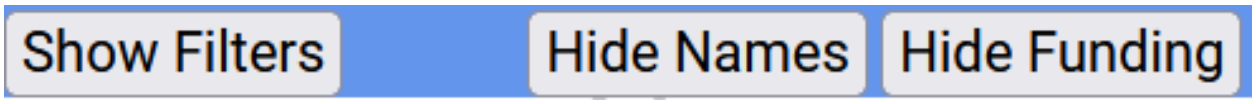
Clicking on the "Add Filters" button in the top left opens a window of tags and years to filter projects by and organizations to filter investigators by. Within this window there is a search bar and a menu of checkboxes for project tags. You can either search for a specific tag in the search bar using the autocomplete UI and click on "Add Tag", or manually select tags yourself. Clicking on "Clear All Tags" deselects all tag checkboxes and the network will show all project

links unless filtered by investigator affiliation and project year. Below the menu of project tags is a menu for investigator affiliation. If no organization checkboxes are selected, all investigator nodes will be displayed on the network. Below the menu of investigator affiliation is a range slider for project years. Only projects with funding years that are within the selected year range will be displayed on the network. Upon clicking “Set Filters,” the network will automatically reload to only display projects and investigators with the selected filters. If an investigator node has no associated projects displayed after filtering, the node is removed from the network. Clicking “Clear Filters” reloads the network to its default state with no filters selected. Clicking on the “Hide Filters” button hides the filter window.

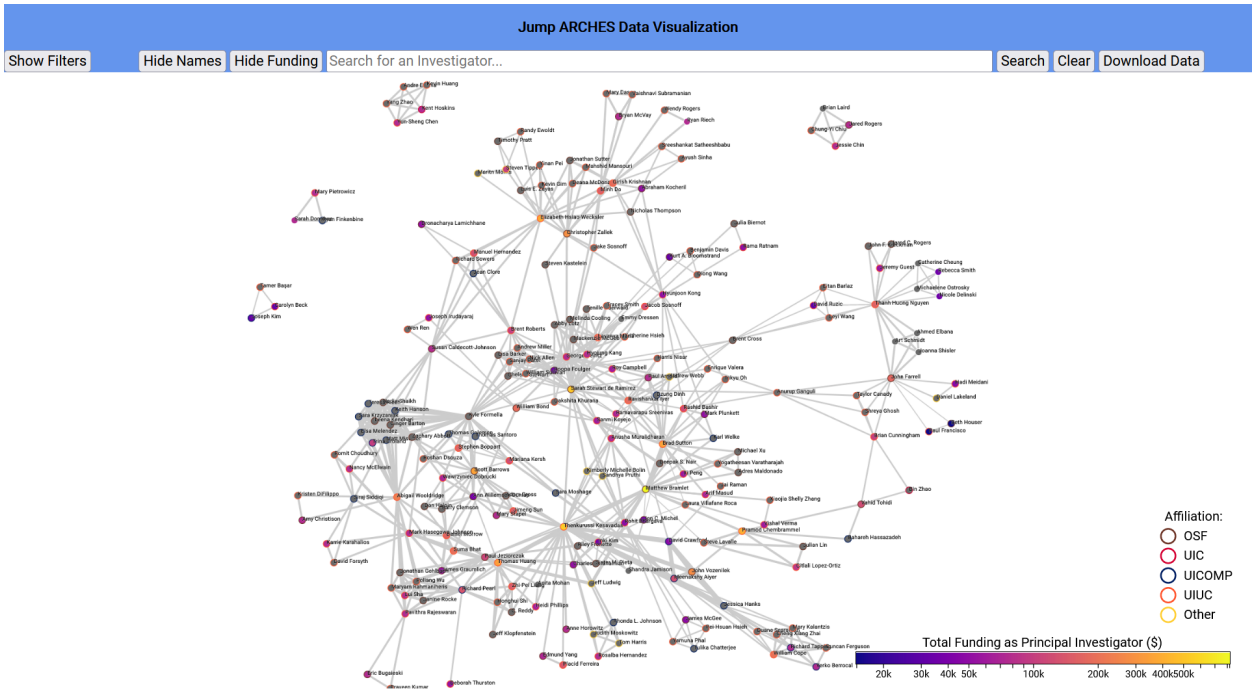


Toggle Investigator Names

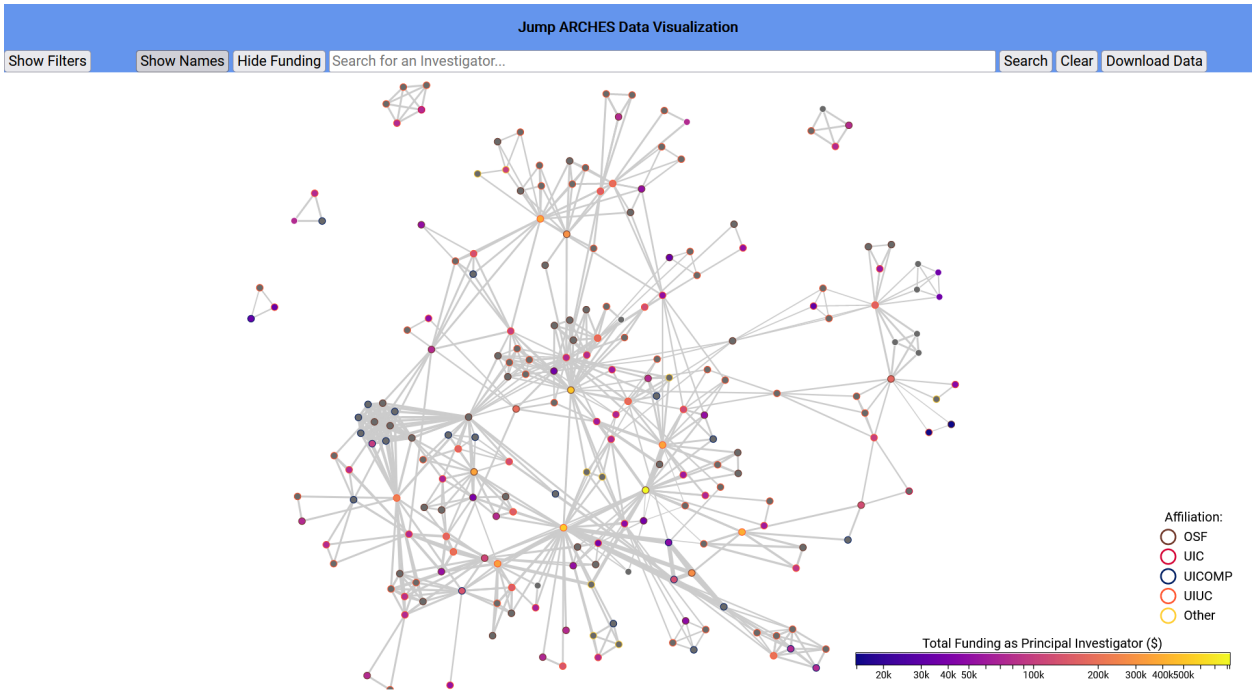
Clicking on the “Hide Names” button sets the opacity of the investigator name labels to 0, rendering them transparent and decluttering the visualization network. Clicking on the “Show Names” button resets the name label opacity to full.



The “Hide Names” button can be found in the page header



Investigator names are visible

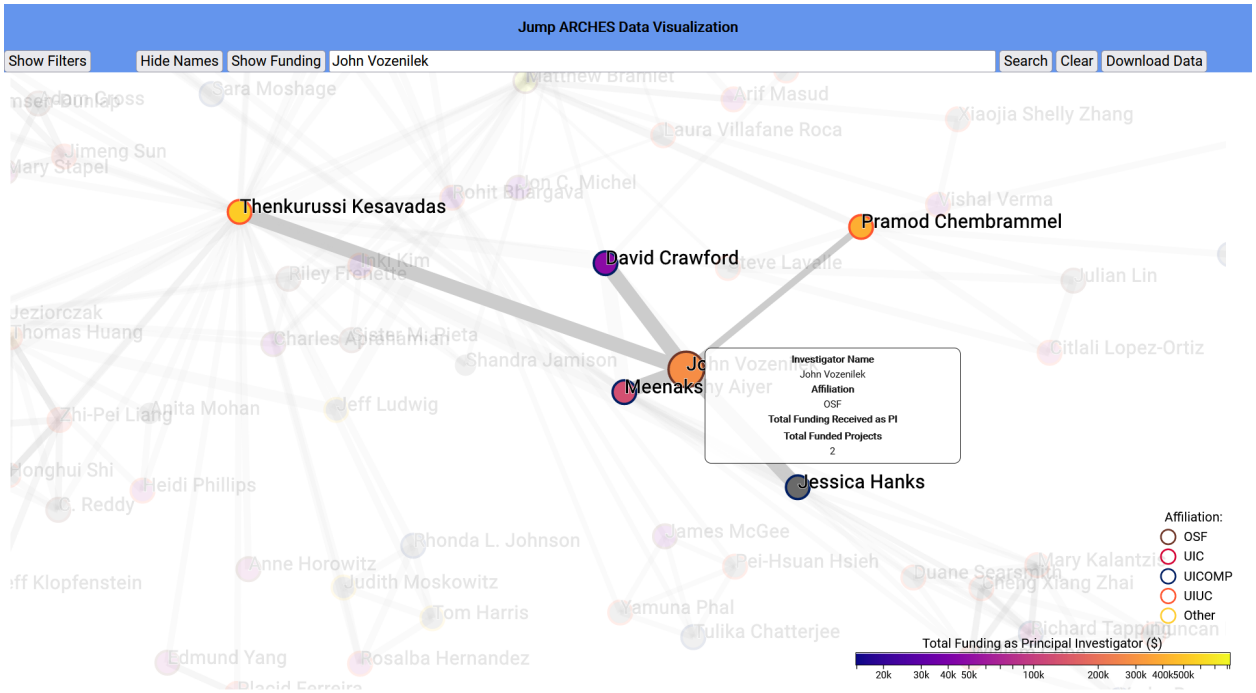


Upon clicking the “Hide Names” button, investigator names are hidden

Toggle Investigator Funding

Clicking on the “Hide Funding” button hides the total funding of an investigator, for all projects that they were principal investigator of, in their tooltip window and information panel. More

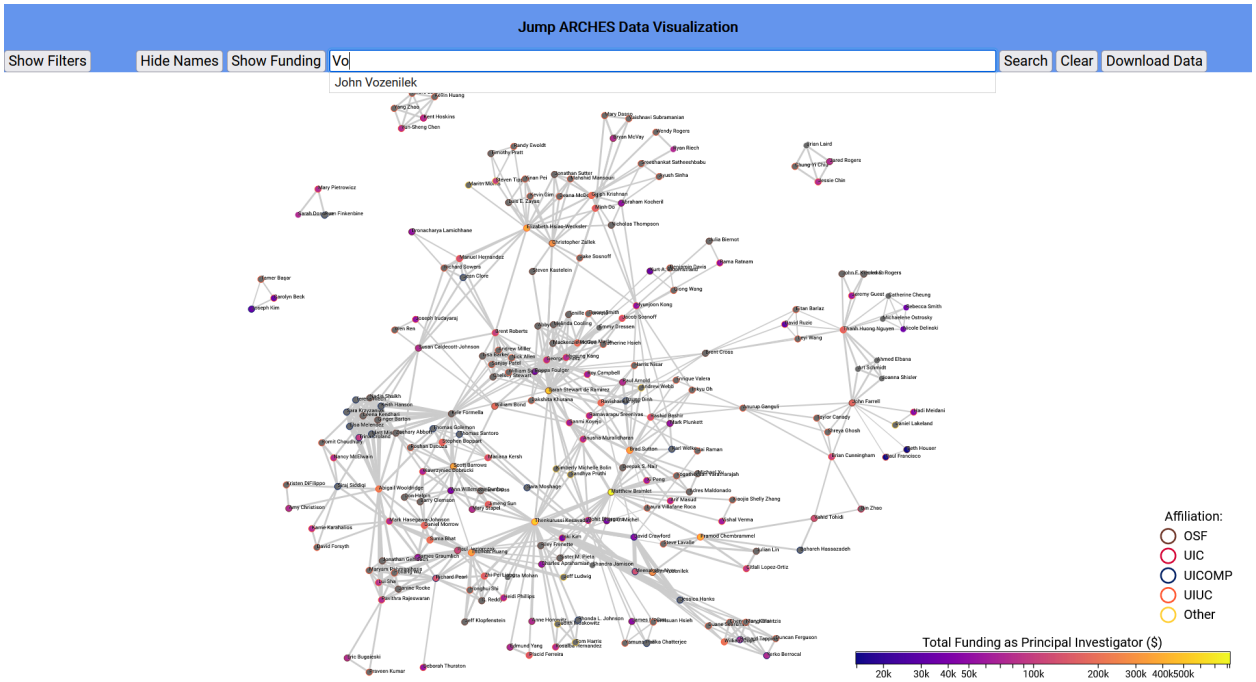
information about those sections are below. Clicking on the “Show Funding” button resets the investigator funding visibility.



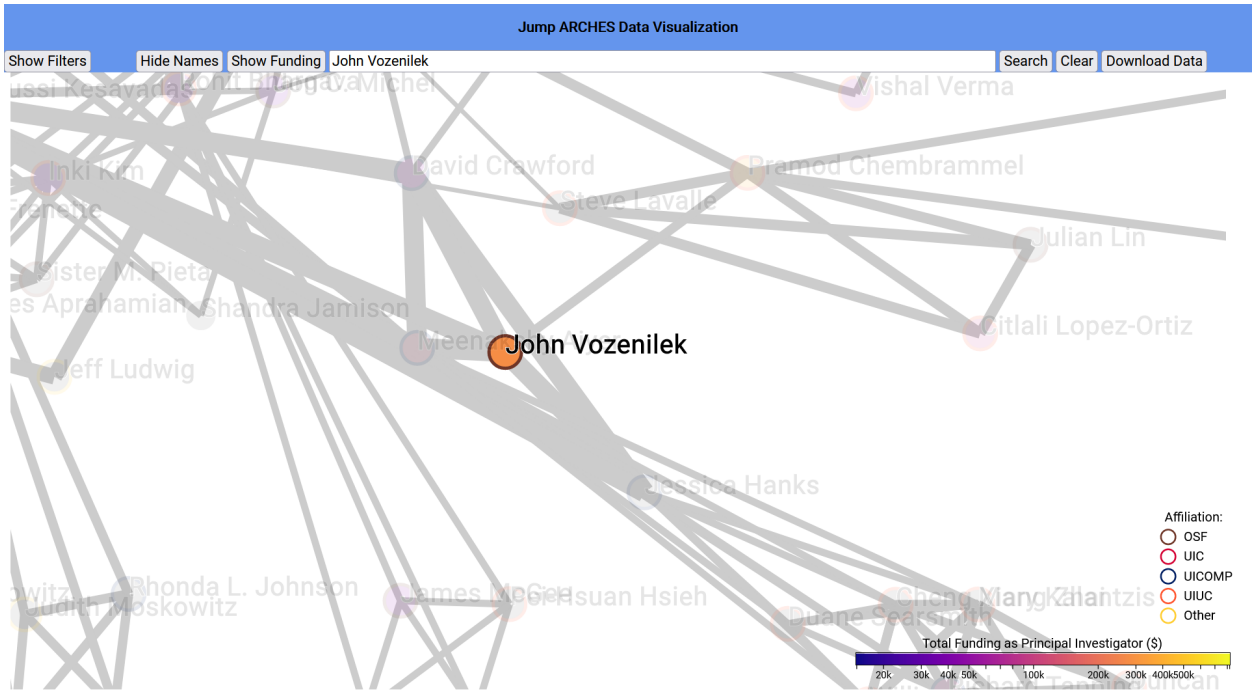
Upon clicking the “Hide Funding” button, investigator funding values are hidden

Searching for an Investigator

The search bar allows the user to easily find an investigator of interest. Typing part of name into the search bar opens an autocomplete UI with name suggestions. Once an investigator is chosen, clicking the “Search” button decreases the opacity of every node and link in the network except for those related to the investigator of interest.



Investigator search bar has an autocomplete feature

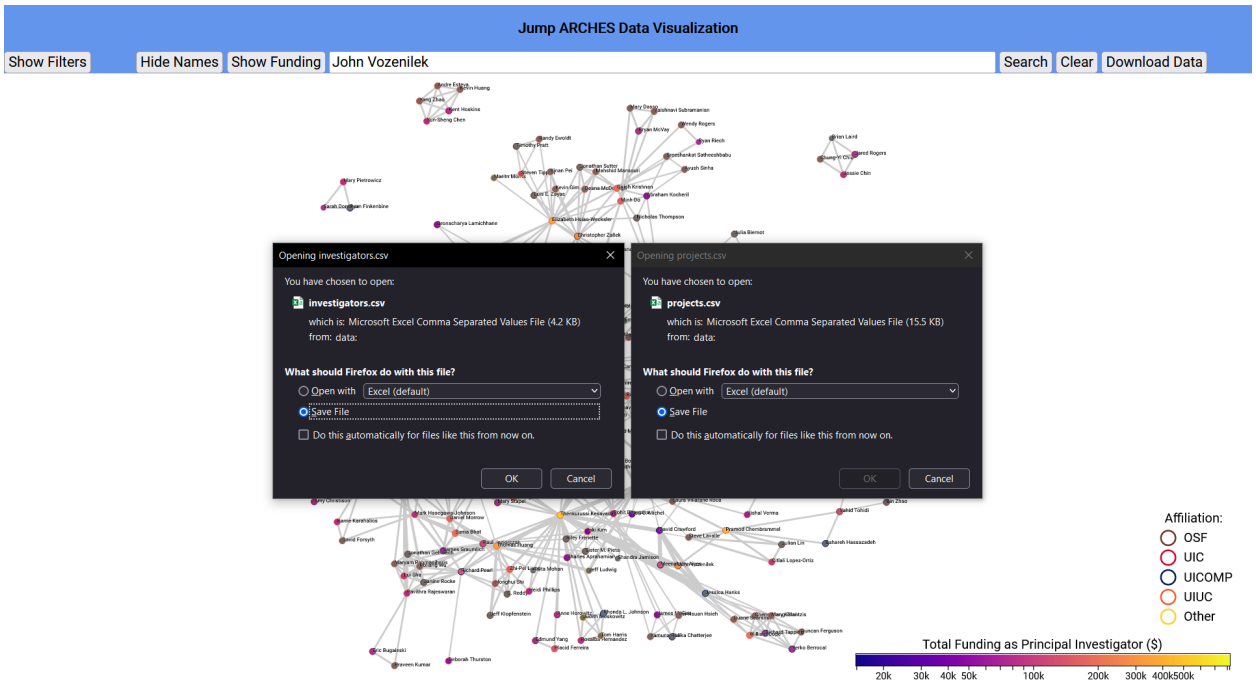


Searching for an investigator

Clicking on the “Clear” button resets the network visibility back to normal.

Downloading Data

Clicking on the “Download Data” button prompts two download dialogue boxes of CSV files containing project and investigator information.



Note: download dialogue boxes may overlap completely

Network

Drag anywhere on the background (whitespace) of the network to navigate around. Use the scroll wheel to zoom in and out of the network. Nodes can be dragged around the screen but will return to their equilibrium position once released.

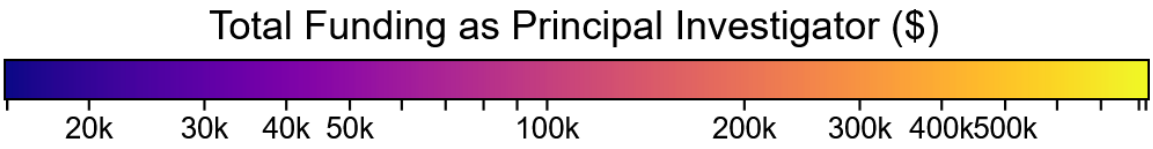
Investigators

Investigators appear as nodes in the network. Investigator name labels are bound to their respective nodes but can overlap each other. The strength of repulsion between investigator nodes decreases with distance.

Fill

The color of the fill of the node indicates the investigator’s total funding for all of the projects that they served as principal investigator for. The legend for this color scale can be found in the bottom right corner of the visualization tool.

Note: Investigators that have not been a principal investigator for a project have a node color fill of gray.



Color scale legend for total funding

Border

The color of the border of the node indicates the investigator's affiliation. The color legend for these organizations are found in the bottom right of the visualization tool above the funding scale.

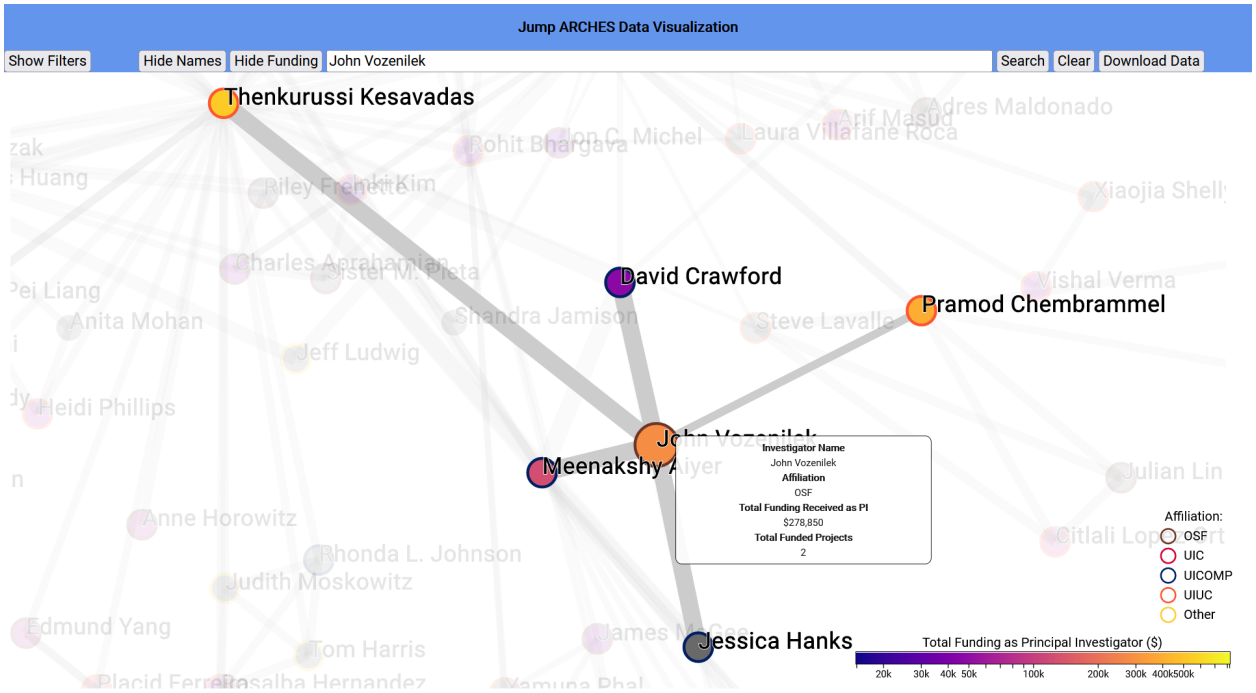
Note: Investigators not assigned to an organization or other in the "All names.xlsx" file have a node border color of white

Affiliation:
- OSF
- UIC
- UICOMP
- UIUC
- Other

Color legend for organization affiliation

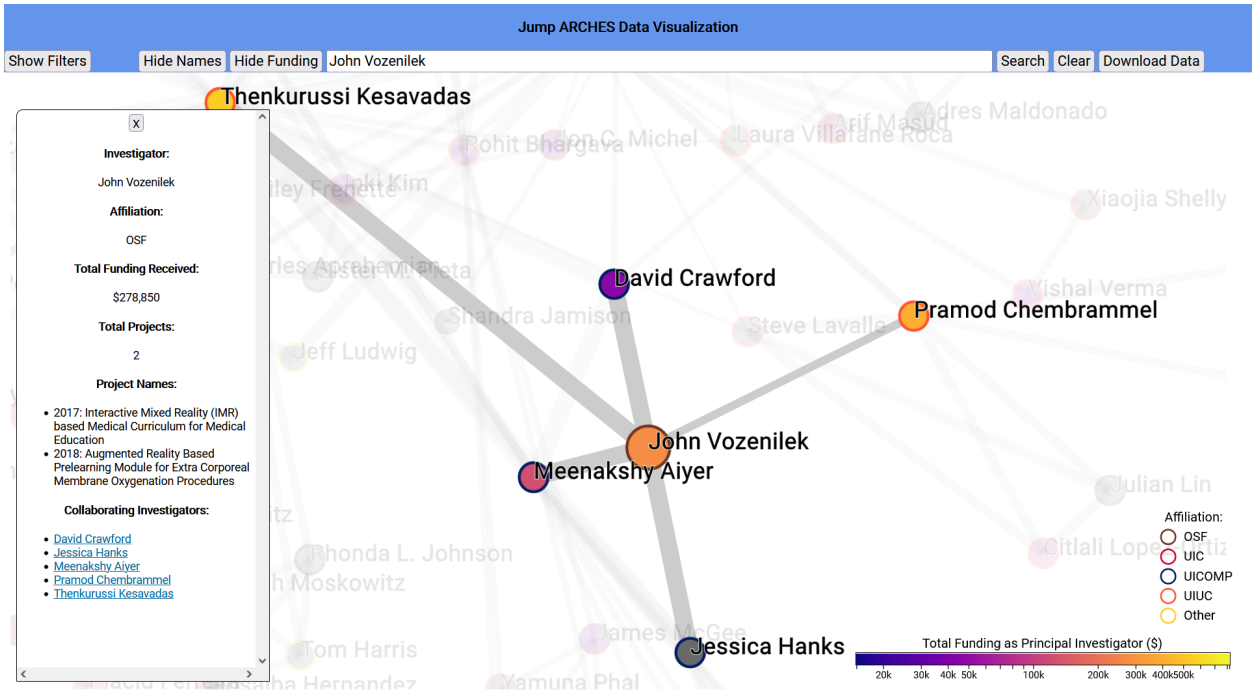
Tooltip

Hovering the cursor over an investigator's node opens a window with some brief information containing the investigator's information. It also increases the size of the selected node and decreases the opacity of links and nodes that aren't connected to the selected node.

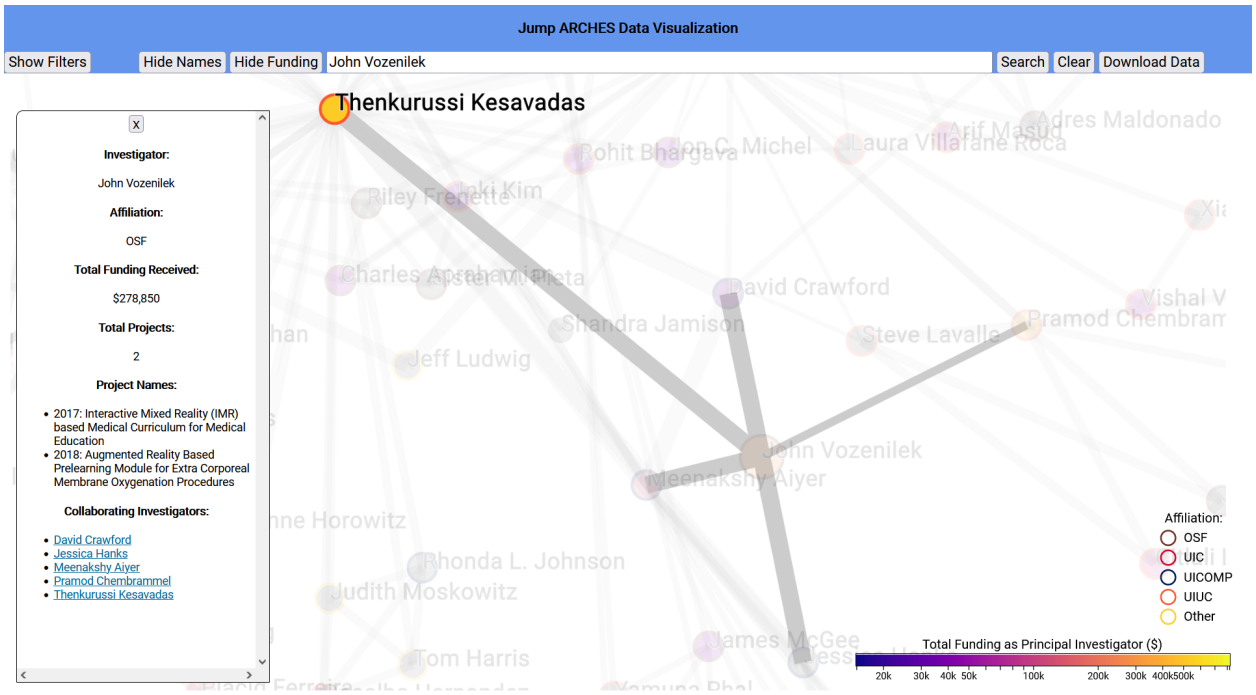


Information Panel

Clicking on an investigator's node opens a panel on the left side of the visualization tool containing all of the investigator's information. Clicking on the same investigator's node or the X button will close the panel.



The panel includes a list of collaborating investigators. If an investigator on that list is visible on the network after filtering, their name will appear as a button. Clicking on their name will then highlight their node.



Clicking on the button for "Thenkurussi Kesavadas" in the Collaborating Investigators list highlights that investigator's node

Projects

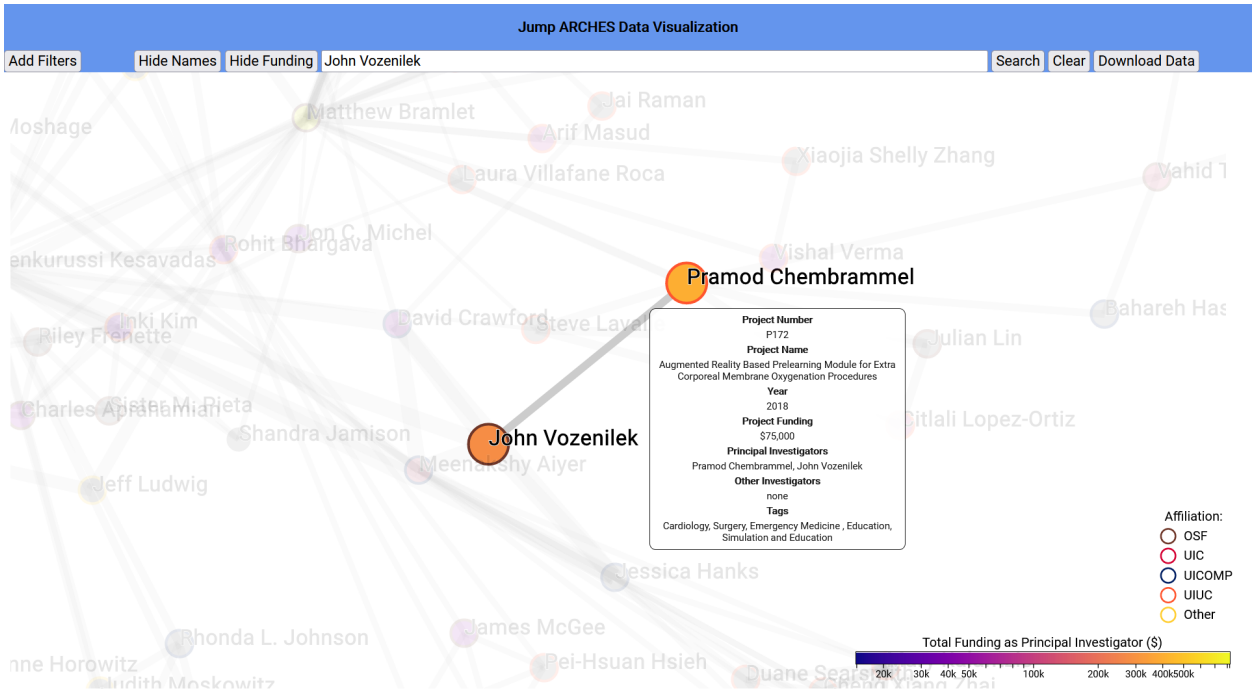
Projects appear as lines in the network to link investigator nodes. Projects are a force of attraction between investigators, allowing the network to come to equilibrium as greater collaboration between investigators causes clustering.

Line Width

The width of each project line represents the total funding that project received. This allows for rapid comparison of funding between projects.

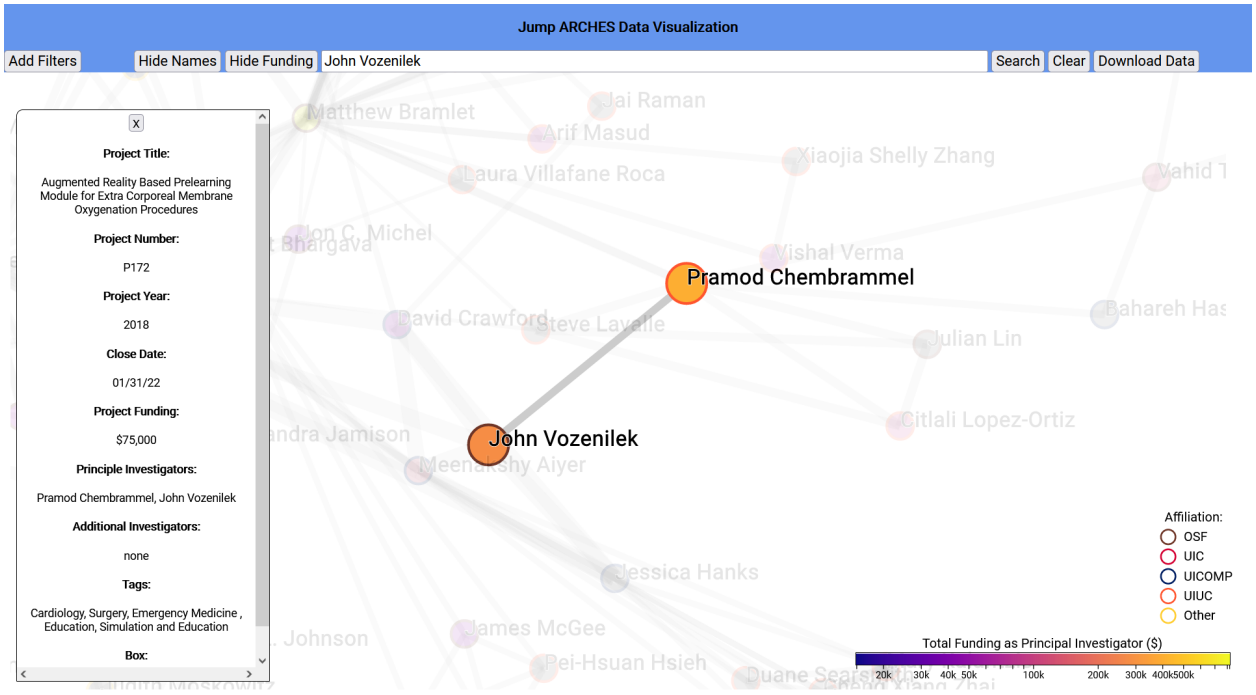
Tooltip

Hovering the cursor over a link opens a window with some brief information containing the project's information. It also increases the size of the connect nodes and decreases the opacity of links and nodes that aren't connected to the selected link



Information Panel

Clicking on a project's link opens a panel on the left side of the visualization tool containing all of the project's information. Clicking on the X button will close the panel.



Updating Data

Excel

Our visualization data tool uses two excel files: “ALL Funded ARCHES Projects.xlsx” and “All names.xlsx.”

ALL Funded ARCHES Projects

The first sheet of this excel file contains information about all funded projects, including year, award number, round, project number, principal investigators, project title, funding amount, additional investigators, categories, and tag numbers.

The second sheet is a key for tag numbers and names.

The third sheet contains basic information about miscellaneous projects.

The fourth sheet contains information about unfunded projects.

All names

This Excel file contains investigator names, email addresses, and organization affiliation.

MATLAB

The MATLAB LiveScript file “ARCHES_connectWriter.mlx” parses through the two Excel files to generate a map-like JSON file called “ARCHES_connections.json”. The LiveScript compiles project information from “ALL Funded ARCHES Projects.xlsx”, sums principal investigator funding, scales this funding value to [0,1] linearly and on a log scale, assigns organization affiliation from “All names.xlsx” to each investigator, and creates five arrays to be stored in the JSON:

1. The “nodes” array contains investigator information.
2. The “links” array contains project information for every investigator-investigator connection.
3. The “values” array contains funding and year values necessary to generate the PI funding legend and the year filter slider.
4. The “tagNames” array contains tag names.
5. The “orgNames” array contains organization names.

Note: Our script does not check the Excel sheets for misspelled names, so any misspelled names will become a new node separate from the correctly spelled names. Misspellings must be corrected in the Excel sheet.

JSON

“ARCHES_connections.json” is the heart of the data visualization tool. It contains all of the information for investigators, projects, and tags in a map-like format. Using a map-like JSON allows functions to access elements in the arrays using either an index or a key.

Modifying Tool Features

Investigator and project interactions are driven by forces implemented by the d3-force module in the D3.js library. Our tool is inspired by Mike Bostock’s Force-Directed Graph. We use D3v5 since there is more documentation available for the modules we implemented. D3v6 also changes and deprecates some modules we implemented. d3-selection, which we use to capture mouse events, has a new event manager and d3-collection, which we use to map data structures for elements keyed by string, is deprecated.

The continuous color legend for principal investigator total funding was inspired by Kai’s Block. Our implementation uses the “Plasma” color spectrum by van der Walt and Smith from the d3-scale-chromatic module in the D3.js library. We chose this perceptually-uniform color scheme due to increased distinction between nodes without potential overlapping color ranges. Tick marks for the scale were generated using the d3-axis module in the D3.js library.

Plasma colormap

We used Susie Lu’s d3 Legend library to create an ordinal legend for the organization affiliations. This library allows legend icons, labels, and stylization to be easily modified with a few functions, rather than having to draw each SVG shape individually. The version of this library implemented in this visualization is compatible with D3v4 and D3v5. Lu has not released an updated version compatible with D3v6.

Function Documentation

Functions in our code were split into various javascripts in our /scripts/ folder in order to more thoroughly organize the code. Some functions, however, were required to be kept in the main javascript file titled ARCHESgraph.js, as they required manipulation of graph variables that would be inaccessible through imported functions. In this section, we detail the documentation for the functions in each of the JavaScript files in our code base, including the parameters they use and their functionality.

ARCHESgraph.js

Read and Store Data

Function	Parameters/Variables Used	Purpose
d3.json(graphFile) .then(function(g))	graphFile – JSON file containing project information g – data from graphFile graph – data object for all nodes and links that are currently visible in the network store – data object for all nodes and links storeAdjlist – A dictionary containing all of the adjacent nodes in the data store	Overall function that loads data for network visualization by parsing through JSON file and storing them in both data graph and data store

Container Formatting

Function	Parameters/Variables Used	Purpose
svg.call(d3.zoom() .scaleExtent([.1, 4]) .on("zoom", function()	None	Detects event when mouse scroll wheel is used and then scales the svg container up or down

Graph Simulation

Function	Parameters/Variables Used	Purpose
d3.forceSimulation()	graphLayout – D3 force-directed graph	Sets force of repulsion between nodes (force is inversely proportional to the square of the separation distance). Sets force of attraction between linked nodes (force is constant). Sets a small force of

		attraction for all nodes to the center of the page.
addFilters()	tagIDs – array of project tag names to include in filtered network orgFilterList – array of organization names to include in filtered network g – object containing data loaded from graphFile JSON filterYears – array of project years to include in filtered network	Checks values of each filter checkbox in the tags menu and overwrites tagIDs. Checks values of each filter checkbox in the organization affiliation menu and overwrites orgFilterList. Checks values of each year slider handle and overwrites filterYears.
clearFilters()	tagIDs – array of project tag names to include in filtered network orgFilterList – array of organization names to include in filtered network "#slider" – JQuery UI slider object filterYears – array of project years to include in filtered network	Clears tagIDs, orgFilterList, and filterYears arrays. Resets handles of “#slider” to min and max years.
update()	graphLayout – D3 force-directed graph graph – data object for all nodes and links that are currently visible in the network adjlist – A dictionary containing all of the adjacent nodes in the graph	Pauses simulation, implements autocomplete for investigator search bar, merges new nodes and links from data graph into graphLayout, updates adjlist, and restarts simulation of graphLayout.
dragstarted(d)	d – selected node	Detects when the cursor drag event starts and begins overriding node position with cursor position.

dragged(d)	d – selected node	Sets cursor position as node position instead of allowing forces to determine node equilibrium. This allows nodes to be dragged around the visualization by the user.
dragended(d)	d – selected node	Detects when the cursor drag event ends and stops overriding node position with cursor position.
ticked()	node - node in the graph	Updates node and link positions per tick
fixna(x)	x – double value, x or y coordinate of a node or link endpoint	Returns – x if x is a finite value or 0 is x is not a finite value.
updateNode(node)	node – every node in the data graph	Returns - CSS styling to translate node SVG element to new node position per tick. Duration of 100 used to reduce twitchiness.
updateLink(link)	link – every link in the data graph	Returns - CSS styling to translate link SVG element to new link endpoint positions per tick. Duration of 100 used to reduce twitchiness.
focus(d)	d – selected node	Hovering over a node performs focusing and creates a popup window with some relevant investigator information
focusLink(l)	l – selected link	Hovering over a link performs focusing and creates a popup window with some relevant project information
unfocus()	node – node in the graph link – link in the graph	Resets opacity of all nodes and links to full once selected node or link is deselected.
clickNode(d)	d – selected node	Brings up information panel about selected investigator
clickLink(l)	l – selected link	Brings up information panel about selected project

exitPanel()	nodeClicked – boolean if node info panel is open. linkClicked – boolean if link info panel is open.	Fades out info panel and resets values.

Filtering

Function	Parameters/Variables Used	Purpose
filter()	orgFilterList – array of organization names to include in filtered network tagIDs – array of project tag names to include in filtered network filterYears – array of project years to include in filtered network graph – data object for all nodes and links that are currently visible in the network store – data object for all nodes and links	If no project tags or investigator organizations are selected and the filter year range is set to min and max years, reset the network by loading node and link data from the store. Otherwise, use orgFilterList to find matching nodes in the store and add them to the graph, and remove non-matching nodes from the graph. Then use tagIDs and filterYears to find matching links in store and add them to the graph, and remove non-matching links from the graph.

Filtering Helper Functions

Function	Parameters/Variables Used	Purpose
reformatTagName(tagName)	tagName – string of tag used for projects in JSON	Returns – String of a tag name reformatted to be compatible as HTML id attributes. Spaces are removed and special characters are replaced with a hyphen.
numberWithCommas(x)	x – double	Reformats numbers with commas.
$('#filterPanel').on('click',	#filterPanel – button on menu	Detects event when Filter

function()	bar #filterBar – filter window	Panel button is clicked, either opens the filter window if currently closed or vice versa, by changing css display property.

Investigator Funding Toggle

Function	Parameters/Variables Used	Purpose
toggleFundingVis(event)	event – Event detected when user clicks on “#toggleFunding” button	If investigator fundings are visible in the info popup and info panel, sets opacity to 0 and changes button text to “Show Funding.” If fundings are not visible, sets opacity to 1 and changes button text to “Hide Funding.”

Investigator Label Toggle

Function	Parameters/Variables Used	Purpose
toggleNameOpacity(event)	event – Event detected when user clicks on “#toggleNames” button	If investigator names are visible, sets opacity to 0 and changes button text to “Show Names.” If names are not visible, sets opacity to 1 and changes button text to “Hide Names.”

Investigator Search Bar Function

Function	Parameters/Variables Used	Purpose
searchNode()	selectedVal – String currently in investigator search bar searchedName – stored value of investigator of interest	Stores user input text for an investigator of interest for highlighting nodes. All other investigators in the network decrease node opacity to 0.1.
clearFocus()	searchedName – stored value of investigator of	Resets opacities of all investigator nodes to 1 and

	interest	clears searchedName so it won't interfere with highlighting new nodes.

Download Data

Function	Parameters/Variables Used	Purpose
exportCSV(graph)	graph – D3 object produced from JSON file that contains investigator and project information.	Exports visualization information and prompts download of two files, investigators.csv and projects.csv

Document Size

Function	Parameters/Variables Used	Purpose
getWidth()	None	Returns – width of page based on maximum of body and document elements
getHeight()	None	Returns – height of page based on maximum of body and document elements

/scripts/addTagList.js

Function	Parameters/Variables Used	Purpose
createTagList()	None	Uses sorted list of tag names to generate HTML div elements, each containing a checkbox and a label for each tag
getObjectValuesAlphabetical(dict)	dict – list of JSON objects	Returns – list of sorted values
searchTags()	tagName – string from text user input box for searching tags	Changes properties of checkbox elements in tagListWindow. If the label of

		the checkbox matches the tagName that is being searched and is currently unchecked, then it is set to checked. If the label matches and the checkbox is checked, then it is set to unchecked.
clearTags()	“#tagListWindow” – list of HTML div elements for tags	Unchecks every checkbox in “#tagListWindow”.
reformatTagName(tagName)	tagName – string of tag used for projects in JSON	Converts tag names to be compatible as HTML id attributes.

/scripts/contLegend.js

Function	Parameters	Purpose
continuous(selector_id, colorscale, axisScale)	selector_id – HTML id of color legend element colorscale – d3 linear scale mapping defined domain to values of color spectrum axisScale – d3 log scale with defined domain	Creates a canvas element in HTML, uses image manipulation to assign values of the color spectrum to the rectangle, and adds a log scale with tick marks below the rectangle.
formatPower(x)	x – numbering scheme with scientific notation “E”	Returns – reformatted scientific notation numbering scheme using powers of 10

/scripts/graphIndexers.js

Function	Parameters	Purpose
getConnections(d, store)	d – The current node in the graph store – The full store object from ARCHESgraph.js	Returns – Number of connections that the node of interest has in the data store.

getProjects(d, store)	d – The current node in the graph store – The full store object from ARCHESgraph.js	Returns – The names of all of the projects that the node of interest (researcher) has worked on in the data store. If a project is represented in multiple links, it is only included once in the list of projects
getResearchers(node, graph, adjlist)	node – The current node in the graph graph – The full graph object from ARCHESgraph.js adjlist – A dictionary containing all of the adjacent nodes in the graph store – The full store object from ARCHESgraph.js storeAdjlist – A dictionary containing all of the adjacent nodes in the data store	Returns – the total researchers an investigator had connections with, as well as an HTML output of the researchers' names in a bulleted list. Connected researchers visible on the graph have buttons to change the node focus. Researchers that aren't visible remain as text.
neigh(node1, node2, adjlist)	node1 – The first node to check node2 – The second node to check adjlist (or storeAdjlist) – A dictionary containing all of the adjacent nodes in the graph (or store)	Returns – True if the two nodes are neighbors in the graph (for adjlist) or store (for storeAdjlist), and false otherwise

/scripts/orgLegend.js

Function	Parameters	Purpose
function(d)	colorScale – D3 ordinal scale object containing organization names as domain and color hex codes as range	Recalling D3 circle paths to change style of stroke color to color domain and removing fill after using d3-legend to create ordinal legend.

/scripts/yearSlider.js

Function	Parameters/Variables Used	Purpose
slide: function(event, ui)	event – Event that is triggered when the HTML document is ready ui – JQuery UI object that contains handle HTML element, handleIndex number, values array for the multihandle slider	Upon loading the HTML document, the multihandle slider is created with specific default min and max values for both handles.
$("input.sliderValue").change(function()	ui – JQuery UI object that contains handle HTML element, handleIndex number, values array for the multihandle slider	Updates specific indices of slider of slider object to match ui values after dragging a handle.

Known Issues

exportCSV() produces CSV files with incorrectly separated values due to the presence of embedded commas in “ALL Funded ARCHES Projects.xlsx” that are preserved in “ARCHES_connections.json”. A solution to this would be to double quote the fields that contain embedded commas when generating the CSV file.