🌻 Filter Links tab

Do qualitative causal analyseis on the selected links by filtering or manipulating them.

🔍 **What you can do here:** Apply filters to focus your analysis on specific aspects of your data. You can trace causal pathways, group similar concepts, filter by themes or demographics, and much more. Think of this as your analysis toolkit - combine different filters to explore your data from different angles.

The Filter System: overview

Use filters to narrow down and/or transforme the links you want to study. Filters are applied in order, from top to bottom.

- **Default filter**: Factor Label Filter
- **Add Filter +** lets you insert filters at the start or between existing ones
- **Enable/Disable** toggles turn individual filters on or off
- **Remove ✕** deletes a filter
- **Collapse** hides a filter's controls to save space
- **Clear All ✕** resets to the Factor Label Filter

Hard vs Soft recoding

Most filters leave factor labels untouched, but these 'Transform filters' filters temporarily relabel factors:

- Zoom
- Collapse
- Remove Brackets
- Soft Recode Plus
- Auto Recode
- Soft Relabel
- Cluster

No filters actually change your original coding.

💡 Tip: If you want to permanently rename or "hard recode" your factors, there are several ways to do that:

- Search and replace factors
- Search and replace links

For example, after clustering (which may give labels like C11), click a factor on the map and rename it (e.g., "Wellbeing") to save the new name permanently.

Zoom Filter

🔍 **What this does:** Simplify complex factor labels by zooming to higher levels of a hierarchy. For example, turn "Health; Mental Health; Depression" into just "Health" (level 1) or "Health; Mental Health" (level 2). Perfect for getting a big-picture view of your data.

- **Radio buttons** for levels (None, 1-9). Combine with Collapse Filter for label cleanup.
- **Level 1**:
- "foo; bar; baz" becomes "foo"
- "foo; bar; baz" becomes "foo"
- **Level 2**:
- "foo; bar; baz" stays the same
- "foo; bar; baz" becomes "foo; bar"
- **None**: No transformation

Collapse Filter

🏷️ **What this does:** Merge similar factors under one common label. Type or select multiple similar terms like "money", "income", "salary" and they'll all be replaced with the first term. Great for cleaning up data where the same concept is described in different ways.
Widgets:

- **Selectize dropdown** with existing labels where you can select one or more existing factor labels, or type parts of existing labels.
- **Matching options**: Start / Anywhere / Exact
- **Separate** toggle for individual replacements. When off, this filter replaces all matches with first search term. When on, a separate factor is created for each of the search terms.

Remove Brackets Filter

🧹 **What this does:** Clean up your factor labels by removing text in brackets. For example, "Education (primary school)" becomes just "Education". You can remove round brackets ( ) and/or square brackets [ ].

- **Two switches**: Round () and Square [] (both off by default)
- Removes all text within the selected bracket type(s)

Factor Label Filter 🏷️

🎯 **What this does:** Show links connected to factors you care about (e.g. "Education"). Choose how many steps to look upstream (causes) and downstream (effects).
Widgets:

- **Factor selector** with existing labels. By default shows only labels from links currently visible at this stage of the filter pipeline. Use the **Show All** toggle to display all factor labels from the entire project instead.
- **Steps Up** (0-5): How many levels upstream to include
- **Steps Down** (0-5): How many levels downstream to include
- **Source tracing toggle**: Retain only links which are part of complete paths which all belong to the same source
- **Highlight toggle** (default: on): Show/hide custom highlighting (⭐ star and magenta border) for matching factors
- **Matching**: Start / Anywhere / Exact. Matching is case-insensitive.

How to use:

1. Select one or more factors.
2. Set Steps Up/Down to widen or narrow the neighbourhood.
3. (Optional) Turn on Source tracing to require paths from a single source.
4. (Optional) Turn off Highlight to hide the custom highlighting.
5. The map and tables update to show only links on those paths.

All the label and tag filters including exclude filters have three radio buttons below the selectize input called Match: Start (default), Anywhere or Exact to control how search terms match against labels/tags:

- **Start**: Match only at the beginning of text (default)
- **Anywhere**: Match anywhere within the text
- **Exact**: Match the entire text exactly

Multiple search terms are treated as OR not AND. preserve and highlight factors matching ANY of the search terms.

Focused factors show with colored borders in the map and have a star added for easy identification (when Highlight toggle is on).

Exclude Factor Label filter

🚫 **What this does:** Remove unwanted factors from your analysis. Type factors like "Unclear" or "Other" to hide them from your map and tables. Useful for cleaning up your data by removing vague or irrelevant categories.

- **Factor selector** for factors to exclude. By default shows only labels from links currently visible at this stage of the filter pipeline. Use the **Show All** toggle to display all factor labels from the entire project instead.
- **Matching options**: Start / Anywhere / Exact
- Multiple entries combined with AND logic
- If you want to exclude both/all of two or more entries, add another Exclude Factor Label filter.

Path Tracing Filter

🛤️ **What this does:** Find causal pathways between two specific points. Set a starting factor (like "Poverty") and an ending factor (like "Poor Health") to see all the causal chains that connect them. Great for understanding how problems and solutions are linked.

- **From selector** for starting factors. By default shows only labels from links currently visible at this stage of the filter pipeline. Use the **Show All** toggle to display all factor labels from the entire project instead.
- **To selector** for ending factors (results visible in Map and Links). Uses the same label source as the From selector (controlled by the **Show All** toggle).
- **Matching options**: Start / Anywhere / Exact
- **Steps** (1-5): Maximum path length
- **Thread tracing toggle**: Require only paths within same source
- **Highlight toggle** (default: on): Show/hide custom highlighting (⭐ star/magenta border for From factors, 🎯 target/dark yellow border for To factors)
- **Only indirect links** (default: off): Remove all direct links from From to To (only makes sense when both From and To are non-empty)

Exclude self-loops Filter

You can exclude self-loops from the maps, but that is more of a visual change. This is a real filter as part of the filter pipeline. For example, if you are using a filter like Link Frequency that might be retaining link bundles which are actually self-loops, so you might get unexpected results if you use the map setting to remove the self-loops. So this filter is a better way. It simply removes all links which are self-loops from the links table.

Link Tags Filter 🏷

🏷️ **What this does:** Filter your analysis by the tags you've added to links. Show only links tagged as "#important" or "#policy" to focus on specific themes or types of relationships you've identified.

- **Tag selector** with existing link tags from current project
- **Matching options**: Start / Anywhere / Exact

Combine Opposites filter ⇄

🔄 **What this does:** Unify opposite factor labels by matching tags. The simple version: `~good health` is the opposite of `good health`. It's another way to code `Poor health`. This filter rewrites `~good health` as `good health` to combine them under one label but adds `flipped_cause` and/or `flipped_effect` columns to the links table to track which causes and effects were flipped in this way so that this information can be used in maps and tables. There's an alternative, more flexible version: If you have pairs like `Foo [99]` and `Bar [~99]`, this filter rewrites `Bar [~99]` as `Foo [99]` to combine them under one label. That way you can even have multiple opposites: `[~27] Low Income` and `[~27] No Income` might be the opposites of `[27] High Income`. **Toggle** – Turn the filter on/off.

Opposites mode toggles (you can use either or both):

- **~ prefix (e.g. `~ foo`)** – Treat `foo` and `~ foo` as opposites (no brackets or numbers).
- Works with hierarchies too: we flip **every** `;`-separated component when comparing opposites. Example: `~Healthy habits; smoking` is the opposite of `Healthy habits; ~smoking`. **Important**: this “flip every hierarchy component” behavior applies **only** to the **~ prefix** mode (not to the numeric tag mode below).
- `~foo` is always rewritten to `foo` (even if `foo` does not appear elsewhere in the current factor list).
- **[~N] tags** – Treat numeric pairs like `Foo [99]` and `Bar [~99]` as opposites. (Square brackets are optional: `Foo 99` / `Bar ~99`.)

More details on [~N] tags

Labels can be written in pairs like:

- `Foo [99]`
- `Bar [~99]`

where `Bar` represents the opposite of `Foo`. The square brackets are optional - you can use `Foo 99` and `Bar ~99` - but brackets make it easier to remove tags later using the Replace Brackets filter.

If there are any such pairs, with matching integers, and the filter is switched on:

rewrite any Bar [~99] filters as Foo [99] and add new columns...

- `flipped_cause` column tracks which causes were flipped
- `flipped_effect` column tracks which effects were flipped

... to the current augmented links table, so that if the label has been flipped, the value is True and otherwise False.

Strip tags from labels (default: on) – When enabled, removes `[N]` and `[~N]` tag patterns from labels after combining opposites. This keeps labels clean while preserving the tracking information in the `flipped_cause` and `flipped_effect` columns.

Bundling strategy (Separate vs Together):

- **Separate** (default) – Treat each flipped/unflipped **variant as its own bundle** (so counts are plain numbers per bundle). This means that you may often see two or even more links between two factors.
- **Together** – Put all variants into a **single bundle** (one link between two factors). Optionally show an **embellished per-variant breakdown** inside the link label (see "Map legend" below).

Link labels (Simple vs Detailed):

- **Simple** – Show only the total source/citation count (Together mode only).
- **Detailed** – Show the per-variant breakdown using the unicode markers (Together mode only).

Map legend (what you'll see on the map) — 4 variants

Notes:

- The "Link labels" setting only matters when Bundling is **Together** (it is disabled/ignored in **Separate**).
- "Count" below means whatever your map is currently showing for link labels (e.g. **Sources** or **Citations**).
- **Colours**: Separate = per-end flipped status (tail=cause, head=effect; blue=no, red=yes). Together = **average** flipped share (tail=cause, head=effect; blue→red).

1) Separate + Simple

- In the map, you'll often see **multiple parallel links** between the same two factors (one per variant).

- Each link label is a **plain count** for that variant.
- Variants correspond to `flipped_cause` / `flipped_effect`:
- `foo >--> bar` (unflipped/unflipped)
- `foo >-F> bar` (unflipped/flipped)
- `foo >F-> bar` (flipped/unflipped)
- `foo >FF> bar` (flipped/flipped)

You'll see this special `>F->` notation in the Bundle column of the Links Table

2) Separate + Detailed

- Same as **Separate + Simple** (Link labels detail does not apply in Separate mode).

3) Together + Simple

- You’ll see **a single link** between the two factors.
- The link label is the **total count across all variants** (no breakdown).

4) Together + Detailed

- You’ll see **a single link** between the two factors.
- The link label shows a **per-variant breakdown** using these unicode markers:
- `‾n` = neither flipped (`--`)
- `\n` = effect flipped (`-F`)
- `/n` = cause flipped (`F-`)
- `_n` = both flipped (`FF`)
- Example: `‾5 \2 /1` means 5 `--`, 2 `-F`, 1 `F-` (zero variants are omitted).

Exclude Link Tag filter ✖

🚫 **What this does:** Remove specific types of links from your analysis. Exclude links tagged as "#uncertain" or "#duplicate" to focus on higher-quality data. Helpful for filtering out questionable or irrelevant causal claims.

- Same as Link Tag filter except *exclude* links containing these tags. Multiple entries are combined with AND, i.e. only exclude links where both entries match. (💡 Tip: if you want to exclude both/all of two or more entries, add another filter).

Exclude self-loops Filter ↺

You can exclude self-loops from the maps, but that is more of a visual change. This is a real filter as part of the filter pipeline. For example, if you are using a filter like Link Frequency that might be retaining link bundles which are actually self-loops, so you might get unexpected results if you use the map setting to remove the self-loops. So this filter is a better way. It simply removes all links which are self-loops from the links table.

Link Frequency Filter

📊 **What this does:** Focus on the most important causal relationships by filtering out rare ones. Choose "Top 10" to see only the most frequently mentioned connections, or set a minimum threshold like "at least 3 sources" to ensure reliability.

- **Slider** (1-100) for threshold
- **Type**: Top vs Minimum
- **Count by**: Sources vs Citations

Examples:

- **Minimum 6 Sources**: Only links mentioned by 6+ sources
- **Top 6**: Only the 6 most frequent link bundles

By default, setting the slider to 6 means we are selecting only links with at least 6 citations.

If you switch to “Sources”, we are selecting only links with at least 6 sources.

If you switch to “Top” we are selecting only the top 6 links by citation count, etc. The selection respects ties, so that if there are several links with the same count, either all of them or none of them will be selected.

Factor Frequency Filter

📈 **What this does:** Similar to Link Frequency, but focuses on the most important factors (causes and effects). Show only the most frequently mentioned themes or concepts to identify the key issues in your data.
Same controls as Link Frequency but applies to factors instead of links.

Source Groups filter

👥 **What this does:** Filter your analysis by participant demographics or document characteristics. For example, show only responses from "women aged 25-35" or interviews from "urban areas." Perfect for comparing how different groups see causal relationships. This is very similar to the [Source Groups widget](../source-groups-sub-panel/) in the Sources sub-panel, but having it here too means you can add multiple source filters to the pipeline.

- provides
- a prepopulated dropdown called Field with all the metadata fields plus title and projectname
- another multi-selectzie called Value. Multiple values work as OR: either/any count as a match

- a previous/next button pair to cycle through values of the selected group
- Example: Add two Source Groups filters in the pipeline to combine criteria (e.g., first filter Field = gender → Value = women, then another filter Field = region → Value = X) so you see links from women AND from region X.

Everything Filter ▼

👥 **What this does:** Filter your analysis by any characteristic of your links and their sources. Useful for anything not covered by the other filters, for example, - show only links with negative sentiment - Show only links from one source Also shows source separators and their values, often used for common sections within multiple sources texts.

- **Field dropdown** with all fields in the links table
- **Value selector** filtered by selected field
- **Navigation buttons** to cycle through values
- **Clear button** to reset

Soft Relabel Filter ▼

👥 **What this does:** Temporarily relabel factors.

- **Old factor labels** listed on the left
- **New factor labels** editable, listed on the right
- **Load labels button** when pressed, adds into the Old labels list any current factor labels (in links as currently filtered) which are not yet listed in the Old labels list and adds the same Old label to the New column as default.
- **Clear button** to clear the New fields
- **Clear ALL button** to clear all rows

Effect: all factors exactly matching any of the labels in the Old list are relabelled with the corresponding labels from the New list. factors not listed are not relabelled but preserved.

Many use cases:

- temporarily merge multiple factors into one
- you are using magnets and you can't really use the formulation you want because you want to maximise similarity with existing labels
- eg you are using "floods" as a magnet but you really want it as a hierarchical factor like "environmental problems; floods" but you can t use that as a magnet.

Keyboard shortcuts (Win/Linux ⇄ macOS):

- Tab / Shift+Tab: move focus down/up between NEW cells

- Arrow Up/Down: move focus up/down between NEW cells
- Alt+Arrow Up/Down (mac: Option+Arrow): move the current row up/down
- Ctrl+Arrow Up/Down (mac: Cmd+Arrow): move the current row up/down
- Delete current row:
- Shift+Delete (mac: Shift+Fn+Backspace) or
- Ctrl+Shift+K (mac: Cmd+Shift+K)

Potentially, one NEW label might have multiple OLD labels.

Soft Recode Plus filter 🖊

Requires AI subscription

🧲 **What this does:** Group messy factor labels under clearer names you choose (called "magnets"). Example magnets: `Improved health`, `Education programs`, `Income changes`. The filter finds the closest magnet for each label and replaces it.

Controls:

Create Suggestions for Magnets

(collapsed by default): Optional. Ask AI to propose clear names from your current labels. Insert adds them to your magnets box to review/edit.

- **Number of clusters** – Choose how many groups to find for AI suggestions.
- **Representatives per cluster** – How many example labels the AI sees for each cluster (8–20; default 8). Consider choosing more than 8 if you want to split clusters into positive/negative variants.
- **Labelling prompt** - With the usual buttons to save and recall previous prompts
- **Insert**

Main panel

- NEW: **Only unmatched** – A new toggle which appears right at the top, before the Create Suggestions subpanel. default off.
- NEW: **Apply to link tags** – When enabled, Soft Recode Plus processes **link tags** (instead of factor labels). Tags are split on commas, trimmed, and then matched to magnets just like labels. Reserved tags like `#plain_coding` are never recoded.
- **Magnets** – One magnet per line. Saved per project. Use Prev/Next to browse recent sets.
- **Similarity slider** – The raw labels are dropped if they are not at least this similar to at least one cluster.
- **Drop unmatched** – If on, remove links whose labels don't match any magnet. If off, keep them as they are. (When **Apply to link tags** is on, this drops **unmatched tags**, not links.)

- **Save** – Save magnets and apply the recode.
- **Remove hierarchy** – Strip any text before the final semicolon
- **Clear / Prev / Next** – Manage saved sets.
- **Recycle weakest magnets:** – A slider starting at 0 , default is 0. If the slider is n >0, then we look at the cluster assignenments which would have been returned and find the n clusters which we are going to recycle. Reassign them to their nearest cluster, providing the similarity is still better than the similarity cutoff. This way we don't lose factors / links which are otherwise assigned to smaller clusters which may get excluded later on in the filter pipeline. When it is on zero, it makes no difference and we just use the solution based only on the magnets, similarity, and remove_hierarchy. The maximum value changes to match the total number of magnets.

If you have say 50 magnets but later apply a filter which only shows 5 factors, in that kind of case it can be nice to apply this slider, to reassign the links for the 45 factors which do not show, to the ones that do show, if they fit. It is not automatic because it depends on you looking to see how many factors are left in the FINAL output e.g. maps. It should work whether you drop unmatched or keep them.

Recoded columns

When you use Soft Recode Plus, the Links and Factors tables show special columns that track which labels have been recoded:

- **Links table**: Shows `_recoded_cause` and `_recoded_effect` columns (✓ for recoded, ✗ for not recoded)
- **Links table (tags mode)**: Shows `_recoded_tags` column (✓ if any tag was recoded at current similarity cutoff)
- **Factors table**: Shows `_recoded` column (✓ if the factor appears at least once as recoded, ✗ otherwise)
- These columns only appear when Soft Recode Plus is active in your filter pipeline
- You can filter by these columns using the True/False dropdowns in the table headers

These columns track recoding from any filter that transforms labels: Soft Recode Plus, Zoom, Collapse, Remove Brackets, Soft Relabel, Cluster, Hierarchical Cluster, and Combine Opposites.

Process only unmatched NEW

the point of this is: what if I apply some (maybe standard) magnetisation and matches plenty of factors but there might be some important material left unmatched, not just noise. so i can use a PAIR of these filters. in the first one, I leave OFF its Discard Unmatched toggle and in the second filter switch ON its Only Unmatched filter. (if there is no preceding SRP filter with Discard Unmatched=OFF, this second SRP filter does nothing).

So now,

- the Create Suggestions (if used) optionally processes ONLY the UNMATCHED factor labels
- the magnetisation (if labels are non-empty) works only on the unmatched factor labels.
- the actual output of the second filter is now the union of both soft-recode processes, i.e. the original matches from the first and the new matches of the previously discarded material from the second.
- the Discard Unmatched on this second filter works as usual: if it is OFF, then we also return all the still-unmatched labels.

Meaning Space (2-D projection of embeddings)

Go to the map formatting and select Layout → Meaning Space to see a 2-D scatter plot of your factor-label embeddings (projected down to 2 dimensions).

- Magnets are shown with labels; raw factor labels are dots.
- Colour indicates the magnet group; magnet dot size represents group size.
- Meaning Space uses the **Similarity** and **Drop unmatched** settings from your most recent Soft Recode Plus filter:
- If **Drop unmatched = ON**: unmatched raw labels are not shown (dot count shrinks as you increase Similarity).
- If **Drop unmatched = OFF**: unmatched raw labels are still shown, but they render in **grey**.
- You can pan (drag) and zoom (mouse wheel and zoom controls).
- Double-click on an empty part of the map to zoom in at that point.
- Tooltips on dots show the original (raw) labels and the magnet label.

Motivation for Remove Hierarchy

"Remove hierarchy", default off. if on, strip any text before a final semi-colon, if no semi-colon, do not change the text.

```
something; another thing
```

is treated same as

```
another thing
```

.... but it continues to be treated as "something; another thing" in the rest of the filter pipeline.

Quick workflow:

1. (Optional) Open Create Suggestions for Magnets panel → set Number of clusters and use Insert to get AI suggestions.
2. Use these suggestions and/or edit them, paste or type your own magnets (one per line).
3. Click Save.

4. Clusters your current labels (factors as currently filtered), ranks typical examples, and asks AI to suggest clear names.

5. Returns suggested names into the magnets box; you can edit them before Save.

See tips on using the history to reuse both your labelling prompt and magnet sets.

Motivation for "recycle weakest magnets": suppose you create 20 magnets, and then apply more filters like say a link frequency filter so that you end up with say only 5 factors. If you then *remove* those factors from the magnets list which are *not* included in the final output, you will usually increase the coverage of your map (re-assigning raw labels which fit best with one of the "lost" labels but still fit well with one of the "surviving" labels). This is what the Recycle slider does: it recycles the specified number of smaller magnets and reassigns them to the larger magnets. So in the example, if you start off with 20 magnets but your final map only shows 5, try recycling say 10 or even 15 of the missing factors.

Note that Recycle Weakest Magnets is applied BEFORE Drop Unmatched.

Clustering filter

Requires AI subscription

🎯 **What this does:** Automatically discover groups of similar factors in your data using machine learning. The system finds natural clusters of related concepts and labels them with cluster numbers. Great for exploratory analysis when you're not sure what causal themes exist.

- **Enable toggle** (starts disabled)
- **Number of clusters** (1-9)
- **Server-side processing** using `cluster_factors_pgvector` database function
- Uses k-means clustering on factor embeddings
- Labels clusters with numeric IDs

Auto Recode filter

🪜 **What this does:** Quickly turns your current set of labels (after any previous filters like Zoom) into a simple tree you can "roll up" or "open out". Pick a small number of clusters for a big-picture view, then nudge the Balance and Similarity to tidy results. Designed for fast, practical exploration on real projects.

Motivation

Making sense of hundreds or thousands of factor labels is hard.

You might use something like soft Recode Plus, but often you'll ask for 20 clusters to cover a wide range of meanings. Then after filtering out insignificant data, you end up with only 7 clusters — losing coverage. Ideally you'd go back and recreate just 7 clusters, but that gives different results. Frustrating!

The point of this Auto Recode filter: have your cake and eat it. Ask for an foldable/unfoldable hierarchical solution. When you move the slider to 15, you get the best solution for 15 clusters. Slide it to 3, you instantly get the best solution for 3 clusters.

Controls:

- **Enable toggle** (starts disabled)
- **Balance (0..1)**: 0 = prefer more distinct clusters; 1 = prefer more even sizes. Changing this can be slow because the tree has to be rebuilt
- **Number of clusters (K)**: 2–50. Unfolds the returned tree locally to K. This is fast unless you increase beyond 20.
- **Similarity** ≥: prune locally by similarity to the centre of each cluster.

NEW: **AI labelling prompt** with history controls. Use this to suggest clearer names for each cluster:

- Saved in the prompts table as type `hierarchical_label` (shared across projects; history shows current first then others).
- A Save button stores your prompt; it also auto-saves on blur and after the first tree build.
- When you raise K (unfold deeper), we call AI in parallel only for the two new child clusters introduced by each applied split, using up to 8 representative labels per child as context. For K clusters this is K−1 requests. Folding to fewer clusters does not call AI; existing AI labels or medoid representatives are used.
- If the prompt is blank, we show the medoid representatives for each cluster.
- If earlier splits already have AI labels (K > 1), we include a reference list of those labels so new labels avoid overlapping meanings.

NEW: **Seed labels (optional)** with history and strength:

- Provide up to K seed labels (one per line). Seeds softly influence split formation but are not included in the final tree (not nodes, not representatives).
- Saved in the prompts table as type `hierarchical_seeds` with standard history controls (Prev/Next/Dropdown/Save).

- Seed strength (0..1) adjusts influence; 0 is a no-op (identical to no seeds). Changing strength or seeds triggers a single backend rebuild (like Balance). Changing K or Similarity does not re-call the backend.

How to use (quick):

- Add the filter and enable it. We build a quick draft tree from the labels you see now (respecting any filters above, like Zoom).
- Set **Balance** if you want more equal-sized groups; the first build may take a moment on large projects (one server call).
- Use **K** to choose how many clusters to show. Changing K is instant (no extra server calls).
- Use **Similarity** ≥ to drop weak matches. If either side of a link isn't matched, that link is hidden.

Notes:

- On very large projects, we automatically sample a representative set to build the tree, then assign the rest to the nearest cluster. This keeps things responsive while preserving the overall picture.
- 💡 Tip: changing the number of factors should be instant if they are less than 20. Setting more than 20 can be slow. If you are going to want more than 20, set this number initially to the maximum number you are likely to want. You can then easily reduce it. Gradually decreasing the number is fine, but *gradually increasing* it will be very slow.

A good prompt looks something like this:

> This is a list of many raw labels grouped into two different clusters, with their cluster IDs, together with a reference list of other labels. Return a list of two new labels, one for each cluster ID. Each label should capture the meaning of the whole cluster, using similar language to the original raw labels, but in such a way that the labels you create are distinct from one another in meaning. Try not to be too generic, try to be as concrete as you can. Do NOT provide labels which include causal ideas, like "X through Y" or "X leading to Y" or "X results in Y" or "X improves Y" etc. Equally, don't include conjunctions in the title like "X and Y". The meaning of the labels you give me should ideally not overlap in meaning with one another or with the labels in the reference list.

Optimized Cluster filter 🎯

⚠️ DEPRECATED Requires AI subscription

⚠️ **This filter is deprecated.** Its functionality has been merged into Soft Recode Plus. The filter still works for backward compatibility with existing bookmarks/URLs, but new instances cannot be created. Use Soft Recode Plus instead for optimal clustering and recoding.

🎯 **What this does:** Automatically finds the most optimal factor labels to use as centroids through genuine optimization. Unlike regular clustering that just groups similar items, this finds the best possible n≤N labels that maximize coverage with similarity ≥S. Perfect for discovering the most representative concepts in your data.

Controls:

- **Max Centroids (n)** - Maximum number of optimal centroids to find (2-50)
- **Similarity ≥** - Minimum similarity threshold for grouping labels (0-1)
- **Timeout (s)** - Optimization time limit in seconds (5-60)
- **Drop unmatched** - Remove labels that don't meet similarity threshold
- **Real-time status** - Shows optimization progress and results

How it works:

1. Extracts all unique labels from your current data (1K-30K labels supported)
2. Runs iterative optimization with multiple strategies (random, frequency-based, diverse selection)
3. Uses hill-climbing optimization to find the best possible centroids
4. Shows coverage percentage and timing information
5. Returns recoded links table with optimal centroid labels

Optimization Strategies:

- **Random selection** - Tests random starting points
- **Frequency-based** - Prioritizes most connected labels
- **Diverse selection** - Maximizes distance between centroids
- **Hybrid approach** - Combines best-so-far with random exploration

Performance Features:

- **Sampling strategy** for datasets >1000 labels (uses representative subset)
- **Early termination** when excellent coverage (≥95%) is achieved
- **Configurable timeout** prevents infinite optimization loops
- **Multiple iterations** with different starting strategies for robustness

- **Smart caching** - Embeddings cached separately from algorithm parameters for fast parameter changes
- **Quote-safe processing** - Handles labels with quotes, apostrophes, and special characters

Technical Implementation:

- Client-side optimization using cosine similarity on embeddings
- Hill-climbing algorithm with local search improvements
- Genuine optimization problem solving (not just k-means clustering)
- Real-time UI feedback showing progress and final results
- Handles massive datasets efficiently through smart sampling
- **Original label preservation** - Stores original labels in `_recoded` metadata for map display
- **Chain compatibility** - Works seamlessly with zoom filter and other transformations

Soft Recode Integration:

- Optimized cluster results available as magnet source in Soft Recode filter
- AI can generate meaningful labels for optimal centroids
- Seamless workflow from optimization to AI-powered naming

This filter implements the optimization challenge described in the technical documentation: finding optimal centroids that maximize label coverage within similarity constraints.

Tribes filter 👥

Requires AI subscription

🏛️ **What this does:** Group your sources (participants/documents) by how similarly they describe causal relationships. This reveals different "tribes" or perspectives in your data - for example, optimists vs. pessimists, or urban vs. rural viewpoints.
Controls:

- **Number of clusters** - Radio buttons: Off, 1-9
- **Similarity cutoff** - Slider: 0-1
- **Drop unmatched** - Toggle
- **Min cluster %** - Slider: 0-20% (prevents "1 big + many singletons" pattern)
- **Counts (Report)** - For the Tribe Report tables: count by **Sources** (unique participants/documents) or **Citations** (links). Default: Sources.

.

It returns:

- `tribeId` (cluster ID)
- similarity to the centroid
- similarity rank These are joined to the links table by source ID and appear as additional columns. If Drop unmatched is ON, links with similarity below the cutoff are removed.

We can then show maps for each tribe and/or for the most typical source in each tribe. we could also then create a typical story centred around the current factors, i.e. told in terms of our concepts.

—->

Custom Links Label 🏷

🏷️ **What this does:** Configure how link labels appear on your map based on any field in your data. Choose what information to display (like tribe memberships or custom attributes) and how to show it (counts, percentages, or statistical significance).
Controls:

- **Field** - Dropdown of available fields from your filtered data (typically shows custom fields like tribe ID)
- **Counts** - Choose whether to count **Sources** (unique participants/documents) or **Citations** (links)
- **Display mode** - Choose how to show the data:
- **Tally** - Show counts for each value (e.g., "T1:4 T2:3")
- **Percentage** - Show what % of each value's total links appear in this bundle (e.g., "T1:34% T2:22%")
- **Chi-square** - Show bundle size, then which values are significantly over-represented (⬆) or under-represented (⬇) (e.g., "45 (T1⬆ T3⬇)")
- **Chi-square (with counts)** - Also show the observed count for each significant value (e.g., "45 (T1 4⬆, T3 3⬇)")
- **Chi-square (with counts/totals)** - Also show observed/total for each significant value (e.g., "45 (T1 4/5⬆, T3 3/6⬇)")
- **Ordinal correction (numeric groups)** - When ON (and group values are numeric-like, including strings like `"12 foobar"`), Chi-square modes use an ordinal trend test and show only the overall total plus **(⬆)** or **(⬇)** (no per-group totals).
- **Sig level** - Significance threshold for Chi-square / Ordinal trend (default p < .05).

⬆ / ⬇ meaning (only in Chi-square modes):

- **Normal Chi-square**: ⬆ over-represented vs expected; ⬇ under-represented.
- **Ordinal correction ON**: ⬆ increasing over numeric groups; ⬇ decreasing.

To use:

1. Add the Custom Links Label filter to your pipeline
2. Select a field (e.g., `custom_tribeId` after running the Tribes filter)
3. Choose a display mode
4. In Map Formatting, set Link Labels to "Custom Links label"

Example use cases:

- **After Tribes filter:** Show which tribes contribute to each connection (T1:5 T2:2 T3:1)
- **Significance testing:** Identify connections where certain tribes are surprisingly over/under-represented (T1↑ T3↓)
- **Custom attributes:** Display any custom field you've added to your data

<!—-

Technical details:

This is a **non-filtering** filter - it doesn't change which links appear, only configures how they're labeled on the map.

Display modes: All modes use the **Counts** toggle:

- **Citations**: each link counts as 1 observation
- **Sources**: each unique `source_id` counts as 1 observation (per value and per bundle)

- **Tally:** For each value, show its count within the bundle (based on chosen Counts unit)

- **Percentage:** For each value, calculates: (count in this bundle) / (total count of this value across all filtered links) × 100 (based on chosen Counts unit)
- **Chi-square (no counts):** For each value, tests whether observed differs from expected (based on chosen Counts unit):
- Expected = (bundle size) × (value total) / (grand total)
- Chi-square contribution = (observed - expected)^2 / expected
- Critical value for $p < 0.05$ with df=1 is 3.84
- Format: `bundleSize (value⬆, value⬇)` (only significant values shown)
- **Chi-square (with counts):** Same test, but format includes observed count: `bundleSize (value observed⬆, value observed⬇)`
- **Chi-square (with counts/totals):** Same test, but format includes observed/total: `bundleSize (value observed/total⬆, value observed/total⬇)`

The filter populates its field dropdown from `currentFilteredLinks` (the output of the filter pipeline), so it sees all fields added by previous filters.