Visualizing and Comparing Patterns of Action using *ThreadNet 3*[1]

Brian T. Pentland
Michigan State University
June 9, 2018

Overview 3
ThreadNet is a work in progress 3
ThreadNet2 versus *ThreadNet3* 4
ThreadNet concepts and vocabulary 5
Occurrences, Events, Threads, and Processual Phenomena 5
Contextual factors 5
Point of view (POV) 5
Basic Algorithm: Weaving threads into networks 6
Zooming in and out on event networks 6
Clock time, relative time and event time 7
ThreadNet data format: .CSV or .XES 8
Required format: First column must contain time stamps or sequence numbers 8
Required format: Remaining columns contain "contextual factors" 8
Example: Electronic Medical Records 8
Some general advice about the user interface 10
Notice the notifications in the lower right corner 10
Interacting with the graphs can be annoying 10
Be patient 10
If you get an error, you can often keep going 10
To report a problem: ThreadNetHelp@gmail.com 10
READ DATA 11
CHOOSE POV 12
Define Threads 12
Define Events 12
Select Subset 13
Save POV 13
VISUALIZE 14
Thread Trajectories 14
Whole Sequences 15
Contextual Factors 16

[1] This material is based upon work supported by the National Science Foundation under Grant No. SES-1734237. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

N-grams 16
Event Network (circle) 17
Event Network (force) 18
Other Networks 19
Role Maps 19
Data Table 20
COMPARISONS 21
Compare POV (synchronic) 21
Compare POV (diachronic) 21
MOVING WINDOWS 22
CLUSTERS AND CHUNKS (experimental) 23
Contextual Chunks 23
Cluster for Zooming 24
Find/Replace Patterns 25
Input your Pattern 25
Select subset of POV 26
Export and delete POV maps 26
Acknowledgements 27
References 27

Overview

ThreadNet is a software application for visualizing and comparing patterns of action (Feldman, 2016a,b). It provides a convenient way to construct what Pentland and Feldman (2007) refer to as a *narrative network*. *ThreadNet* does this by converting *threads* of sequentially related events into *networks.* It implements what Pentland and Liu (2017) call the "bottom-up" method for tracing associations between actions to create the network. It is meant to help qualitative researchers visualize and explore action patterns and hopefully ask better questions using their data.

There are many excellent tools for visualizing and analyzing networks (e.g., UCINet, Pajek, Gephi, NodeXL), for process mining (e.g., Pro/M, Fluxicon Disco, BupaR) and for sequence analysis (e.g., TraMineR). These tools offer a wide range of sophisticated capabilities. *ThreadNet* is aimed at a narrow gap in this landscape: creating, visualizing and comparing narrative networks from different points of view (Pentland and Feldman, 2007; Pentland, Recker and Wyner, 2016).

Figure 1 shows an overview of the steps for using *ThreadNet*. After you read the data and choose a point of view (POV), the rest of the features for visualization and comparison become available. You can choose and compare multiple points of view. You can also read in a different data file to visualize and compare. In the following sections, we provide some pointers about each screen to get you started.

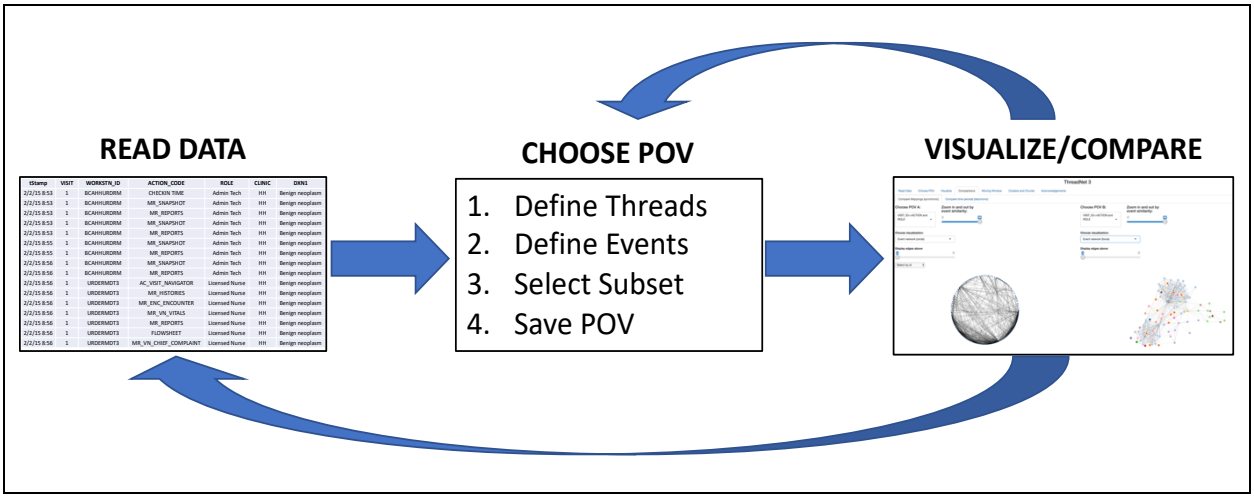


Figure 1: Overview of ThreadNet

ThreadNet is a work in progress

If you are expecting a slick commercial product, you will be disappointed by *ThreadNet*. If you have sequence data and you are looking for new ways to visualize it, *ThreadNet* may give you some ideas. Table 1 should help set your expectations. The main thing is that there are tremendous opportunities for improvement. The goal is to help researchers ask better questions.

Table 1: What ThreadNet3 is and what it is not

It is…	It is not…	Implications for users
A prototype	A finished product	Please suggest ideas for features
Open source	A commercial product	Feel free to develop new features yourself: http://Github.com
Buggy	Thoroughly tested	If you see something, say something: ThreadNetHelp@gmail.com
Exploratory	Confirmatory	Only visualization, no statistical tests
Sequence analysis	Text analysis	Data must be coded. It cannot analyze free text (e.g., uncoded tweets or emails)
Small data	Big data	If your data has more than ~50,000 rows, don't hold your breath
Suggestive	Exhaustive	At best, it scratches the surface

ThreadNet2 versus *ThreadNet3*

There are two versions of *ThreadNet* available for use. Table 2 summarizes the differences.

Table 2: Differences between ThreadNet2 and ThreadNet3

	***ThreadNet* 2**	***ThreadNet* 3**
Language	MatLab	Shiny R
Uses sequence info	X	X
Uses timestamp info		X
Threads → Networks	X	X
Occurrences → Events		X
Metrics (e.g., complexity)	X	
Basic visualization	X	X
Multiple Points of View		X
Interactive visualization		X
Convenient comparison		X
Clustering, Chunking, Zooming		X

This document is focused on the features of *ThreadNet3*. More details about *ThreadNet2* are available in a separate document ("How to Construct Narrative Networks using *ThreadNet", Nov 28, 2016, http://routines.broad.msu.edu/ThreadNet).*

ThreadNet concepts and vocabulary

Occurrences, Events, Threads, and Processual Phenomena

ThreadNet3 makes a distinction between four different levels or layers of time, as shown in Table 3. Loosely speaking, these can be thought of as levels of granularity in time. The distinction between occurrences and events is inspired by historical sociology (Griffin, 1992).

Table 3: Temporal levels of granularity in ThreadNet

Occurrence	Event	Thread	Processual phenomenon
Enter Check-in Time (seconds)	Check-in with receptionist (minutes)	One visit to a dermatology clinic (hours)	EMR Record keeping (on-going and constantly changing)

- **Occurrences** are the finest grained level. Each row in your data is one occurrence. Occurrences are assumed to be instantaneous (no duration). Occurrences are described by a set of contextual factors (the columns in your data).
- **Events** consist of one or more occurrences as seen from a particular point of view. Events are marked by changes in contextual factors.
- **Threads** are temporally ordered sequences of events. Threads are like fragments of stories.
- **Processual phenomena** are a constantly changing fabric of threaded events, wouldn't you agree? That's why it makes sense to visualize them with narrative networks.

Contextual factors

In practical terms, a "contextual factor" is simply a column in the spreadsheet that holds your data. Conceptually, the term "contextual factor" refers to any categorical label or descriptor that you might use to describe a situated occurrence or event. The classics contextual factors include who, what, when, where and why. You can add as many as you need to describe your processual phenomenon.

Point of view (POV)

You can view a process from different points of view. For example, how does it look from the point of view of the actors? Or the computer system? You choose a POV by selecting one or more contextual factors in your data to define the threads and events.

For example, in a dermatology clinic, you might define threads in terms of patient visits. This POV would trace the sequence of events within each patient visit. Alternatively, you could define threads in terms of roles and workstations: When a {nurse, physician, technician, …} works at a particular workstation {A, B, C, …}, what sequences of events do they perform? These threads would contain little fragments of activity. Each unique combination of the factors you choose will be one distinct thread. Within each thread, events are sorted by timestamp.

Within each thread, you choose contextual factors that define events. The most obvious choice would be actions, but you can add contextual depth to your analysis by defining events using any number of additional contextual factors. Events are marked by changes in contextual factors.

Basic Algorithm: Weaving threads into networks

ThreadNet traces along threads to create networks. Conceptually, it does this in two passes. First, it finds the complete set of events as defined by your point of view. Each kind of event becomes one node in the network. Each kind of event appears only once in the network, no matter how many times it appears in the threads. Then, *ThreadNet* traces the sequential relationship between events within the threads. Each sequential relationship between events is an edge in the network. Figure 2 shows the general idea.

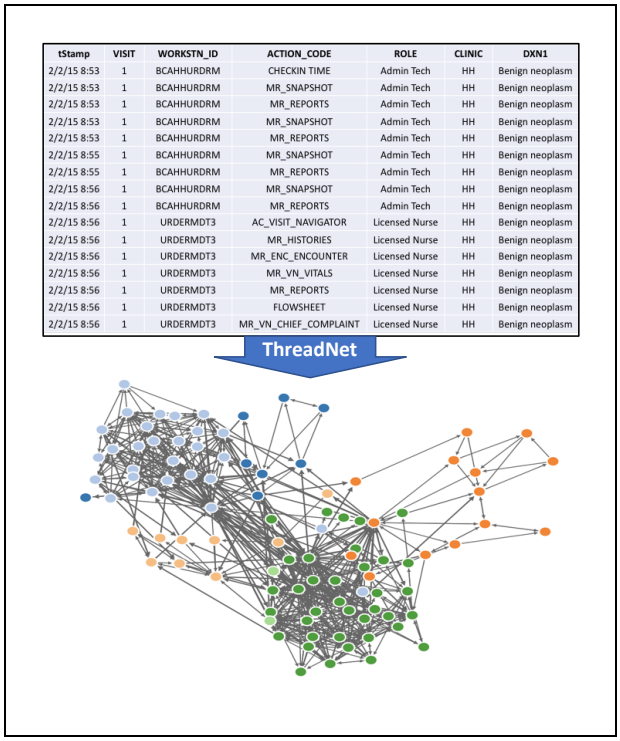


Figure 2: The general idea of ThreadNet

Zooming in and out on event networks

Events can be clustered according to their similarity. For example, does two events contain similar contextual factors (contextual similarity)? Do they contain similar sequences of occurrences (sequential similarity)? Are they close to each other in the network (network similarity)? *TheadNet* offers these three ways to measure distance between events, and uses Ward agglomerative clustering to group similar events into categories.

When events are clustered, we can label them according to their category, from very specific to

very general. *ThreadNet3* provides a slider that allows you "zoom" up and down these cluster levels, as shown in Figure 3. This is different than the zooming envisioned by Nicolini (2009). Zooming in this way does not change the number of events in the threads, but it changes the way they are labeled: *it changes the level of abstraction in the description of the events.* As you will see, the same set of threads looks very different when viewed at different levels of abstraction. Zooming is not applicable to all visualizations; if the slider doesn't work, that's because it would not make sense.

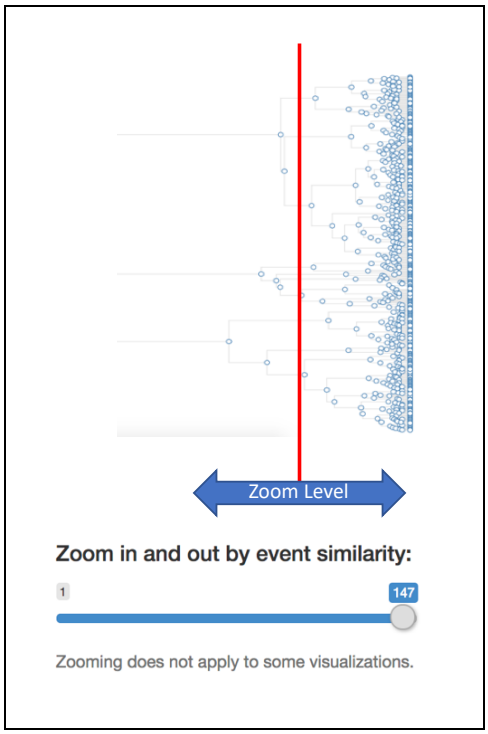


Figure 3: What zooming is really doing

Clock time, relative time and event time

Sequence is essential to the analysis of process. For better or worse, *ThreadNet3* uses timestamps to find the sequential relationship between occurrences and events in your data. If at all possible, try to include time stamps. If you cannot include time stamps, you need to include sequence numbers. ***Clock time*** is simply your timestamp data. For an event with multiple occurrences, it is the timestampe of the first occurrence. ***Relative time*** is defined within each thread as the elapsed clock time since the first event in the thread. ***Event time*** is the sequential number of the events in the thread; it just shows sequence, not "time."

ThreadNet data format: .CSV or .XES

ThreadNet works equally well with data collected from observation, interview, computerized event logs, or any source that can provide a sequence of occurrences. *ThreadNet3* reads in a simple spreadsheet (.CSV) as input (see Table 3 below for an example) or an .XES file (an IEEE standard format for computerized event logs). Each row in the spreadsheet represents one occurrence. Each column in the spreadsheet represents an attribute of the occurrences (action, actor, location, etc). A *thread* is formed by any set of sequentially related rows, and a single spreadsheet can contain thousands of threads. The specific format is explained in more detail below.

Required format: First column must contain time stamps or sequence numbers

Sequence is essential to forming threads. You can provide sequence information in one of two ways: (a) time stamp or (b) sequence numbers.

If your data has time stamps, the time stamps are used to sort the data.

- The time stamp data must be in the first column.
- Allowed formats: "mm/dd/yy hh:mm:ss", "mm/dd/yy hh:mm", "mm-dd-yy hh:mm:ss"
- If you use Excel to edit your data, set the Cell format for the tStamp column accordingly.
- The first column must be labeled "**tStamp**".

If your data does not have time stamps, use integers. The integers are used to define sequence.

- The integers must be in the first column.
- Simultaneous occurrences can have the same integer.
- The first column must be labeled "**sequence**".

Required format: Remaining columns contain "contextual factors"

After the first column, any number of additional columns can be included. In *ThreadNet*, these are referred to as "contextual factors." These columns contain text-based codes that describe the changing context of the occurrences.

- Each column should describe one contextual factor (e.g., who, what, where, etc.).
- Free text (e.g., the text of a tweet) cannot be analyzed directly; it needs to be coded.
- Every row for every column must be filled in.
- Column labels cannot contain spaces.
- Spaces within the coded data will be removed.

Data must be carefully coded. The main challenge in using *ThreadNet* is in capturing and coding the threads: sequences of related occurrences, all described by a set of common attributes. All of the threads go into a single spreadsheet (occurrences are the rows, attributes are the columns, and thread are blocks of adjacent rows). Coding actual processes and stories into a clean set of rows and columns requires editorial judgment and nerve. Categories need to be simplified; details need to be omitted. This is just "coding the data," but it involves a lot of decisions and interpretations.

Example: Electronic Medical Records

In this document, we use data from dermatology clinics at the University of Rochester Medical Center (Ryan, et al, 2016). For the purposes of this illustration, we present a snapshot of the record keeping process at one clinic on one day (February 2, 2015). This data was collected as part of a

larger research project investigating the antecedents of complexity in healthcare routines (NSF SES-1734237).

Table 4 shows the first five minutes of one patient visit, as captured in the EMR audit trail. When patients arrive at the clinic, they check in with a receptionist whose formal role in the system is "Admin_Tech." The computer workstation at the reception desk is labeled "BCAHHURDRM". To complete the check-in, the Admin_Tech visits several screens in the in the EMR system (e.g., "MR_SNAPSHOT"). After the patient is checked in, a Licensed Nurse gets the patient history, enters vital signs, and the chief complaint. Every patient visit begins with checking in at reception and ends with printing a visit summary at checkout. Throughout each visit, the situational and sequential context is constantly changing.

Table 4: Example data

tStamp	VISIT	WORKSTN_ID	ACTION_CODE	ROLE	CLINIC	DXN1
2/2/15 8:53	1	BCAHHURDRM	CHECKIN TIME	Admin Tech	HH	Benign neoplasm
2/2/15 8:53	1	BCAHHURDRM	MR_SNAPSHOT	Admin Tech	HH	Benign neoplasm
2/2/15 8:55	1	BCAHHURDRM	MR_REPORTS	Admin Tech	HH	Benign neoplasm
2/2/15 8:56	1	BCAHHURDRM	MR_SNAPSHOT	Admin Tech	HH	Benign neoplasm
2/2/15 8:56	1	BCAHHURDRM	MR_REPORTS	Admin Tech	HH	Benign neoplasm
2/2/15 8:56	1	URDERMDT3	AC_VISIT_NAVIGATOR	Licensed Nurse	HH	Benign neoplasm
2/2/15 8:56	1	URDERMDT3	MR_HISTORIES	Licensed Nurse	HH	Benign neoplasm
2/2/15 8:56	1	URDERMDT3	MR_ENC_ENCOUNTER	Licensed Nurse	HH	Benign neoplasm
2/2/15 8:56	1	URDERMDT3	MR_VN_VITALS	Licensed Nurse	HH	Benign neoplasm
2/2/15 8:56	1	URDERMDT3	MR_REPORTS	Licensed Nurse	HH	Benign neoplasm
2/2/15 8:56	1	URDERMDT3	FLOWSHEET	Licensed Nurse	HH	Benign neoplasm
2/2/15 8:56	1	URDERMDT3	MR_VN_CHIEF_COMPLAINT	Licensed Nurse	HH	Benign neoplasm
2/2/15 8:56	1	URDERMDT3	MR_REPORTS	Licensed Nurse	HH	Benign neoplasm
2/2/15 8:57	1	BCAHHURDRM	MR_REPORTS	Admin Tech	HH	Benign neoplasm
2/2/15 8:57	1	BCAHHURDRM	MR_SNAPSHOT	Admin Tech	HH	Benign neoplasm
2/2/15 8:58	1	URDERMXRM1	MR_REPORTS	Licensed Nurse	HH	Benign neoplasm
2/2/15 8:58	1	URDERMXRM1	AC_VISIT_NAVIGATOR	Licensed Nurse	HH	Benign neoplasm
2/2/15 8:58	1	URDERMXRM1	MR_ENC_ENCOUNTER	Licensed Nurse	HH	Benign neoplasm
2/2/15 8:58	1	URDERMXRM1	MR_HISTORIES	Licensed Nurse	HH	Benign neoplasm
2/2/15 8:58	1	URDERMXRM1	MR_REPORTS	Licensed Nurse	HH	Benign neoplasm
2/2/15 8:58	1	URDERMXRM1	MR_VN_VITALS	Licensed Nurse	HH	Benign neoplasm
...	...	...	...	...	...	...

Some general advice about the user interface

Notice the notifications in the lower right corner

For example, when you create a new POV, there is a notification in the lower right corner. If you have used the same POV name, it will warn you to choose another.

Interacting with the graphs can be annoying

ThreadNet uses two packages to render graphs: visNetwork (for the circle layout) and networkD3 (for the force layout). These packages have their own ideas about how to change the size and location of the image on the screen. Just try clicking, double clicking, dragging, etc. Eventually you will find a way.

Be patient

Many operations are almost instantaneous. However, converting occurrences to events to create a new POV can take minutes if you have a lot of data (more than 10K occurrences) or a slow computer. Some visualizations also take several seconds to appear. Unless the whole screen suddenly goes grey, *ThreadNet* is still working. If you click away from an operation while *ThreadNet* is still working (e.g., before a graph has displayed), *ThreadNet* may punish your impatience with passive aggressive silence.

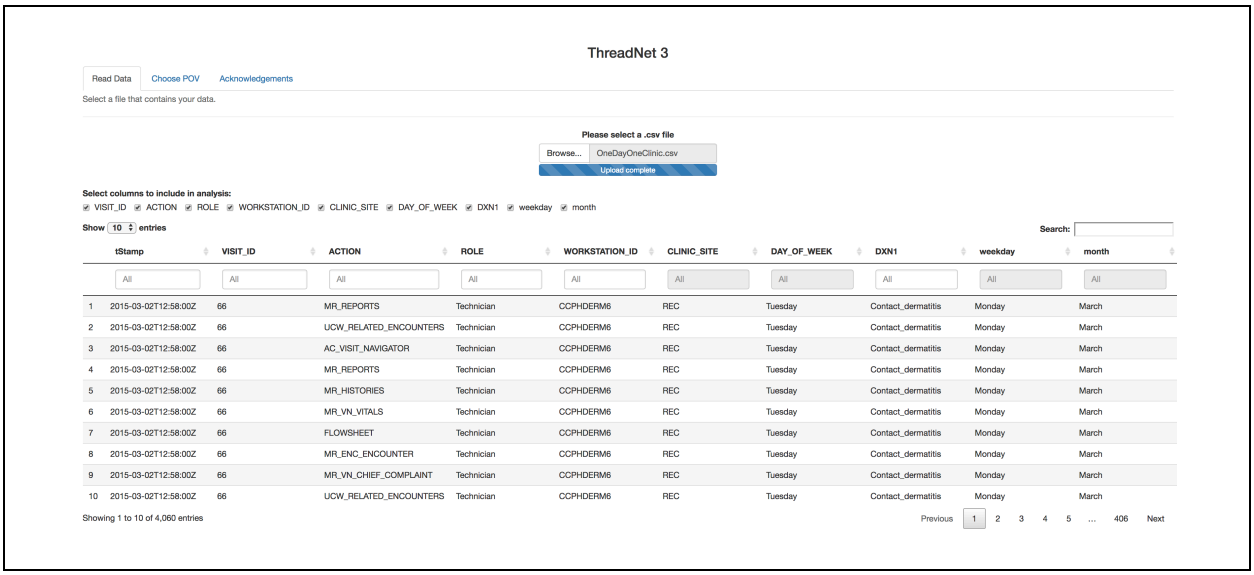
If you get an error, you can often keep going

If you get a cryptic red error message in the UI, but the screen is not greyed out, you can probably keep working. Often, these messages are caused by issues in your data that *ThreadNet* was not written carefully enough to handle. Sorry about that.

To report a problem: ThreadNetHelp@gmail.com

I make no promises about turnaround time, but if a problem is not reported, there is nothing I can do. At a minimum, please send a screen shot of the user interface plus a screen shot of the RStudio console window (if applicable), along with your description of what you think is wrong. Without the screenshots, there is no way to proceed. To solve problems that depend on data, you would also need to send the data.

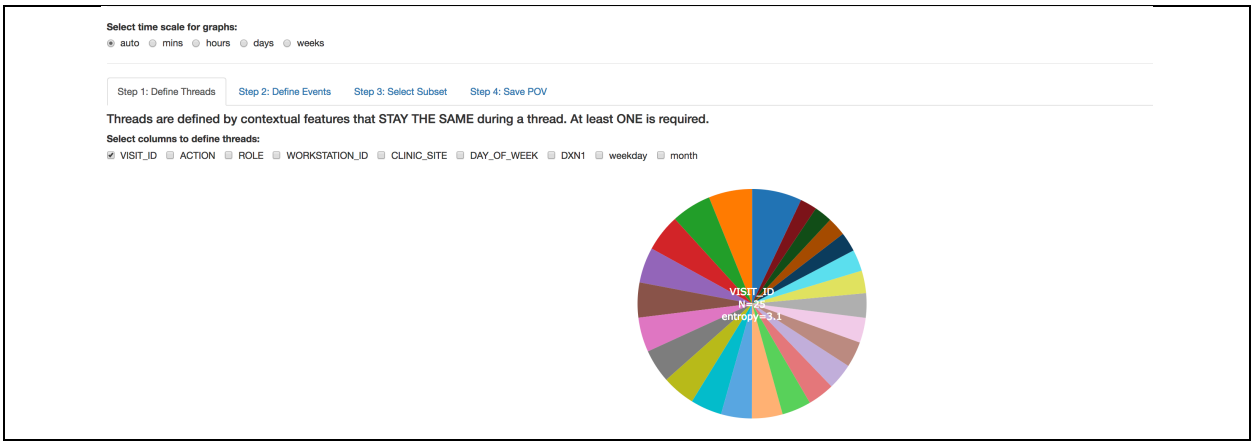
READ DATA



1. You can check/uncheck boxes for columns you do not want to include in the analysis. If you have a column that includes free text, leave it out. In general, it will work faster if you leave out columns you are not planning to use.
2. You can select a subset of your data. *ThreadNet* can read large files (100K rows is no problem), but some of the operations get bogged down with that much data. Pick a small subset to try it out and speed things up.
3. If your time stamps are not in the first column, it simply will not work (see above).
4. If your time stamps come out blank, try reformatting them (see above).
5. If you get an error like “cannot open the connection”, there is something wrong with your data file (see above).

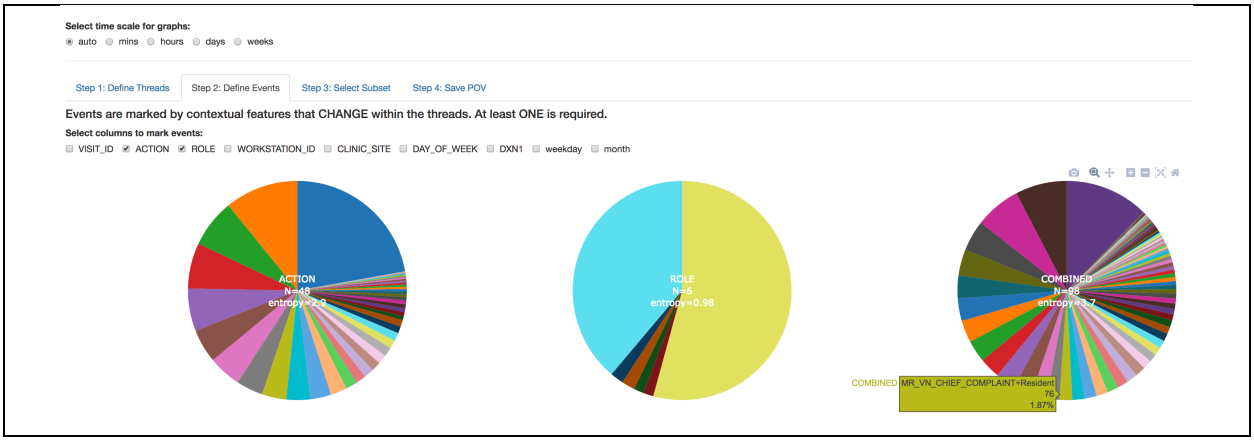
CHOOSE POV

Define Threads



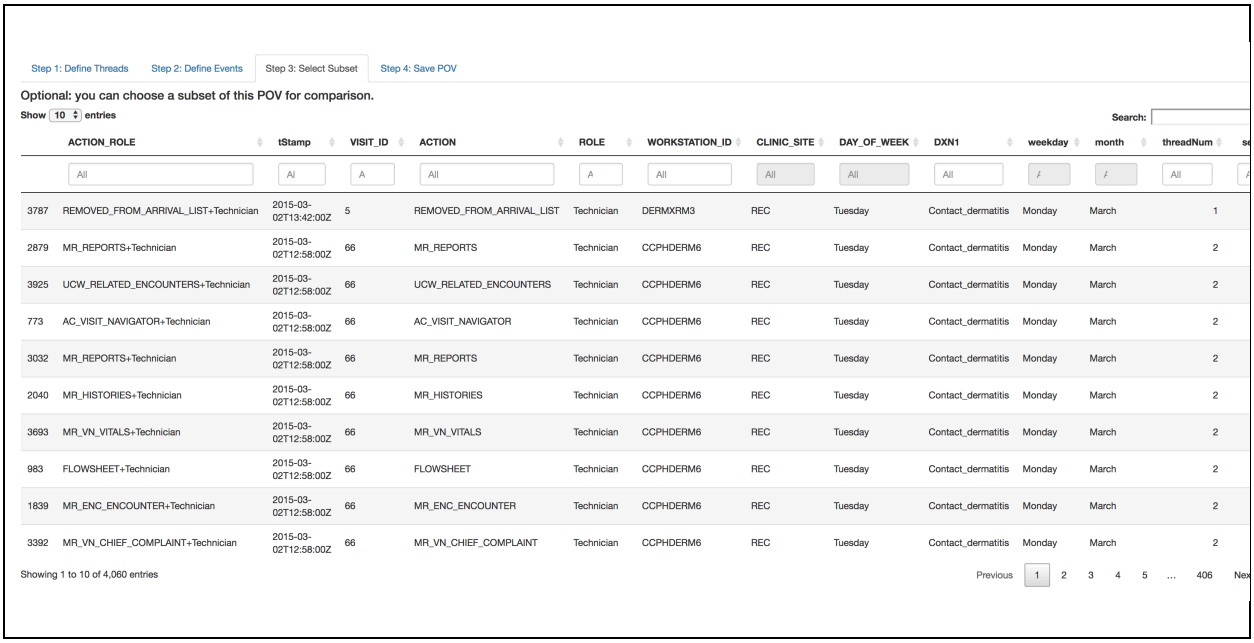
1. Pick one or more contextual factors to define what counts as a thread. For the EMR data, I chose VISIT_ID. I could also have chosen the WORKSTATION_ID to view the data from the point of view of the workstations.
2. The pie charts summarize the total number of rows of data in each category.
3. You can also choose a timescale (mins, hours, days, weeks) or let R choose this automatically. (Spoiler alert: Your selection may not flow through everywhere. Sorry.)

Define Events



1. Pick one or more contextual factors to define events. These can be any aspect of context that changes within a thread.
2. The set of combined contextual factors is shown instantly. Note that it includes only the combinations that actually appear in your data.
3. The entropy is the Shannon entropy for that particular pie chart. A larger number means that it provides more information.

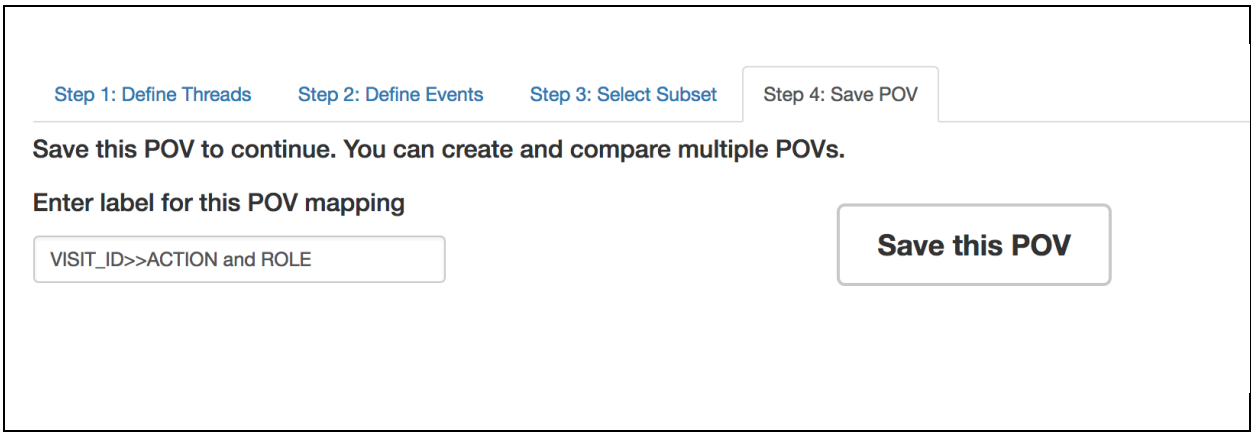
Select Subset



	ACTION_ROLE	tStamp	VISIT_ID	ACTION	ROLE	WORKSTATION_ID	CLINIC_SITE	DAY_OF_WEEK	DXN1	weekday	month	threadNum
3787	REMOVED_FROM_ARRIVAL_LIST+Technician	2015-03-02T13:42:00Z	5	REMOVED_FROM_ARRIVAL_LIST	Technician	DERMXRM3	REC	Tuesday	Contact_dermatitis	Monday	March	1
2879	MR_REPORTS+Technician	2015-03-02T12:58:00Z	66	MR_REPORTS	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2
3925	UCW_RELATED_ENCOUNTERS+Technician	2015-03-02T12:58:00Z	66	UCW_RELATED_ENCOUNTERS	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2
773	AC_VISIT_NAVIGATOR+Technician	2015-03-02T12:58:00Z	66	AC_VISIT_NAVIGATOR	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2
3032	MR_REPORTS+Technician	2015-03-02T12:58:00Z	66	MR_REPORTS	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2
2040	MR_HISTORIES+Technician	2015-03-02T12:58:00Z	66	MR_HISTORIES	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2
3693	MR_VN_VITALS+Technician	2015-03-02T12:58:00Z	66	MR_VN_VITALS	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2
983	FLOWSHEET+Technician	2015-03-02T12:58:00Z	66	FLOWSHEET	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2
1839	MR_ENC_ENCOUNTER+Technician	2015-03-02T12:58:00Z	66	MR_ENC_ENCOUNTER	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2
3392	MR_VN_CHIEF_COMPLAINT+Technician	2015-03-02T12:58:00Z	66	MR_VN_CHIEF_COMPLAINT	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2

1. Here is another opportunity to select a sub-set for comparison. This is optional.
2. Occasionally this conversion may fail due a bug in iGraph. Known issue. Sorry.

Save POV



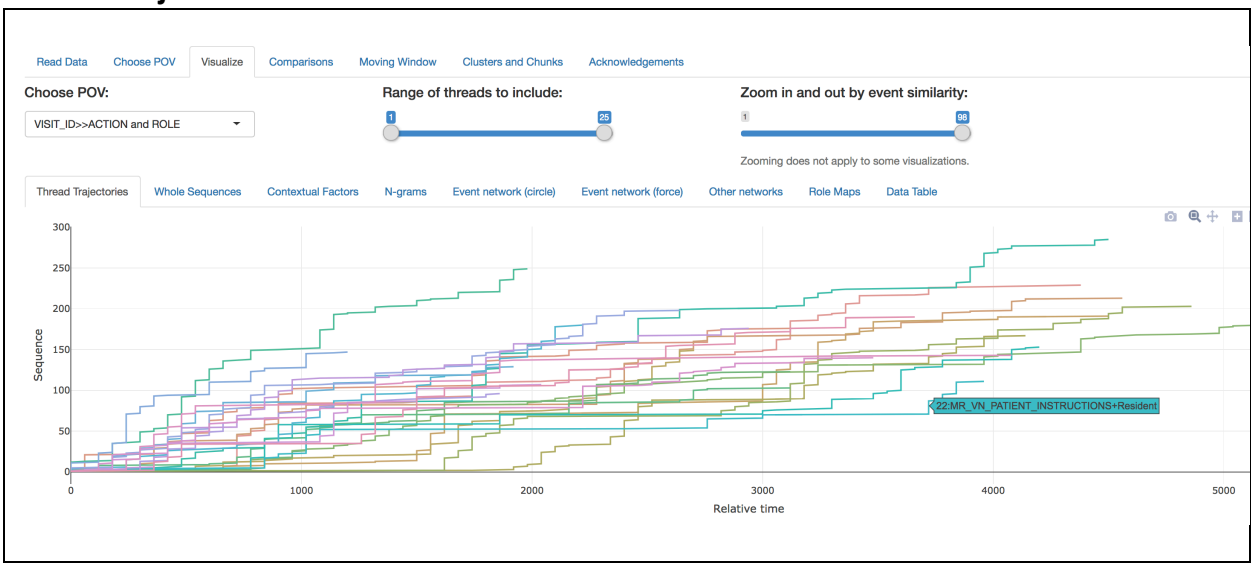
3. *ThreadNet* suggests a name based on the POV, but you can name it anything you want.
4. You must SAVE the point of view before you can move on.
5. You can revisit these steps to choose and save another POV.

VISUALIZE

Here are some general comments and suggestions.

1. This is the fun part.
2. Visualize any POV that you have saved. You can switch between POVs anytime.
3. Within a POV, you can focus on a range of threads.
4. For many visualizations you can zoom in and out.
5. Zooming does not apply to all visualizations and some visualizations are incompatible with some Points of View. It's a long story. What you see is what you get.
6. In general, colors in *ThreadNet* cannot be matched up in the way you would like.

Thread Trajectories

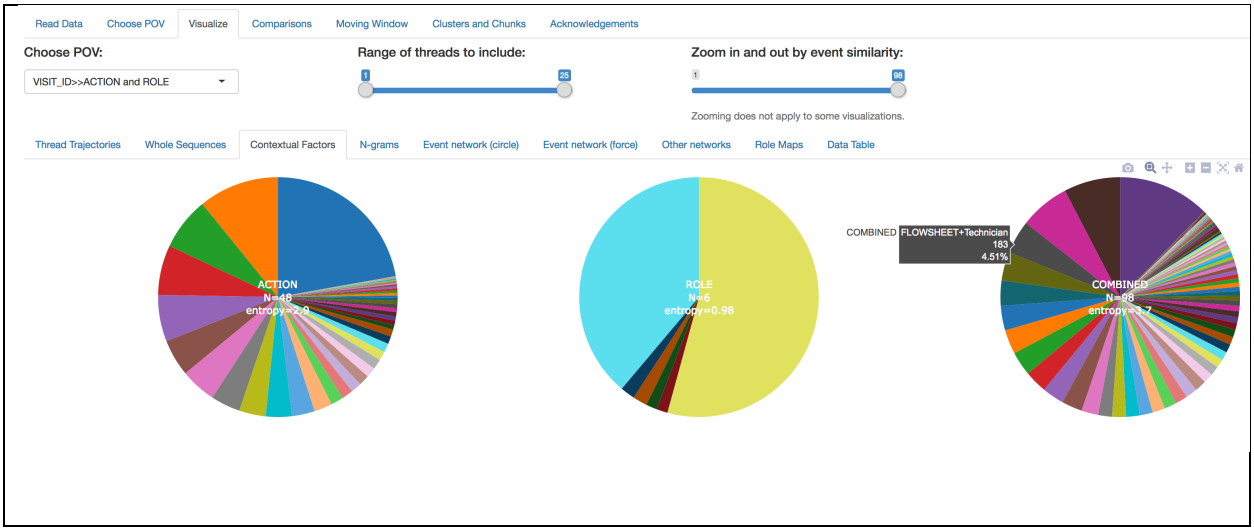


1. This graph was inspired by the work of Daniel Geiger and Anja Danner-Schröder (2016) "It's all about timing", EGOS Naples. It shows sequence (or event time) on the Y axis and relative time (clock time) on the x-axis.When the trajectory is horizontal, it means time is passing, but nothing is getting done.
2. When the trajectory is vertical, it means that things are getting done quickly.
3. You can mouse over the graph to see what is happening in each trajectory.
4. This graph scales to fill the entire width of the screen. If there is a huge time gap in any single trajectory, the rest of the traces can be compressed (so the image looks blank). Try moving the range of threads to see what happens.
5. The a-axis (relative time) should be in the units selected in Choose POV. But since the graph scales to fill the screen the units are irrelevant…

Whole Sequences

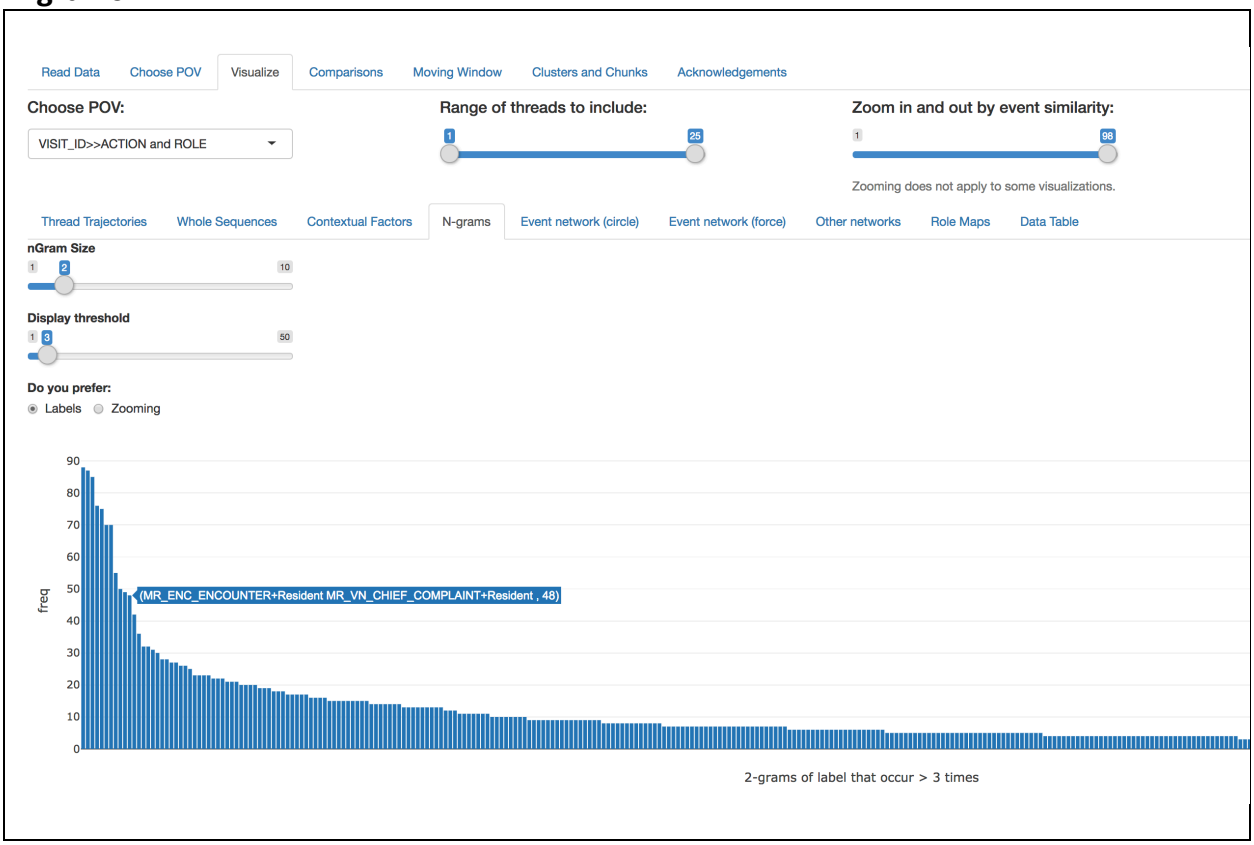
1. Whole sequences can be displayed in event time, clock time, or relative time.
2. When more than one event happens at the "same time", the dots overlap. If you want to see the whole sequence, you must use event time.
3. Zooming changes the granularity of the categories used to describe the events (it puts them more/fewer categories), but it does not change the number of events.
4. As with trajectories, the timescale should reflect what you selected earlier.

Contextual Factors



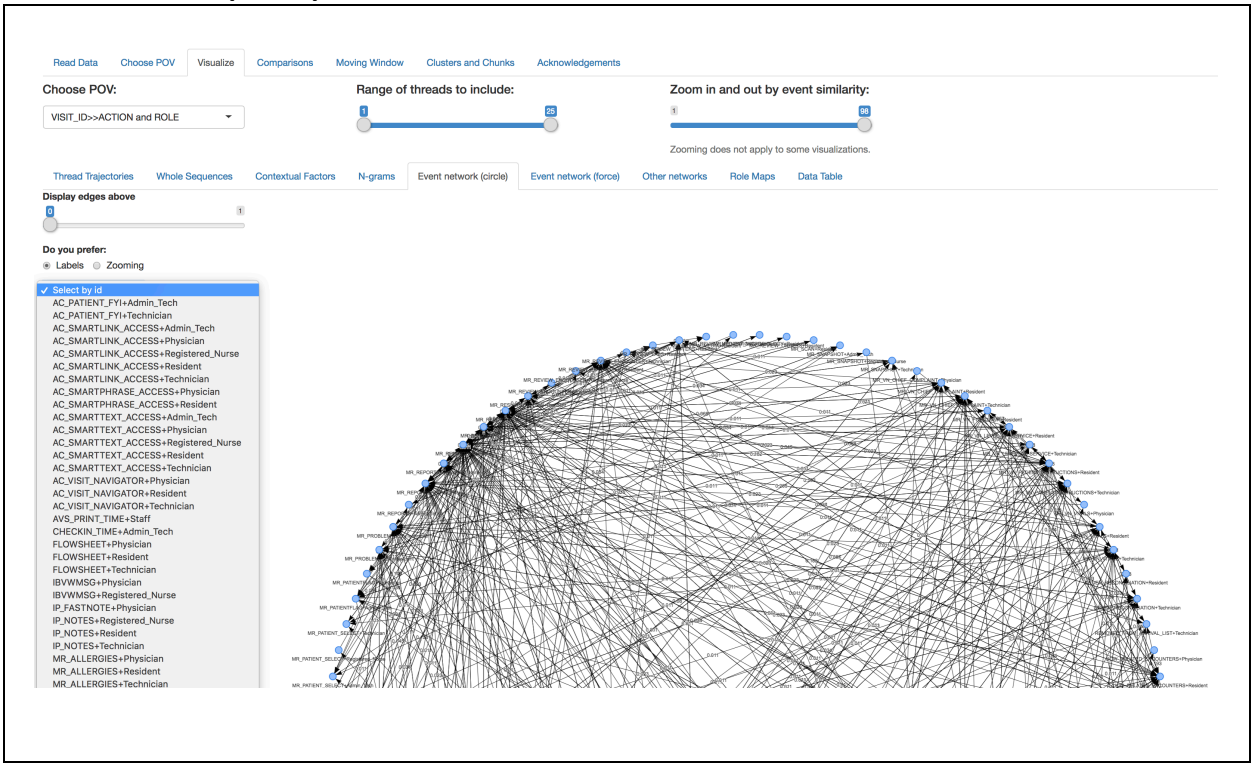
1. This is like the pie charts for contextual factors on Choose POV, but it is display only.

N-grams

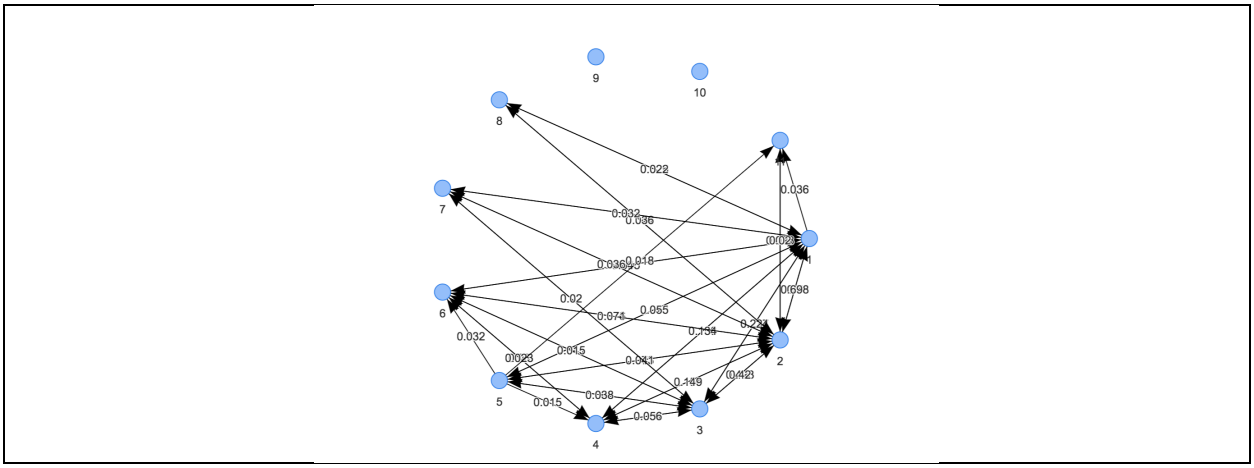


1. N-grams describe sequential relationships. Here, they are sorted by frequency.
2. If you prefer labels, it gives you the event labels for this POV, but you cannot zoom.
3. If you prefer zooming, the labels are no longer meaningful.

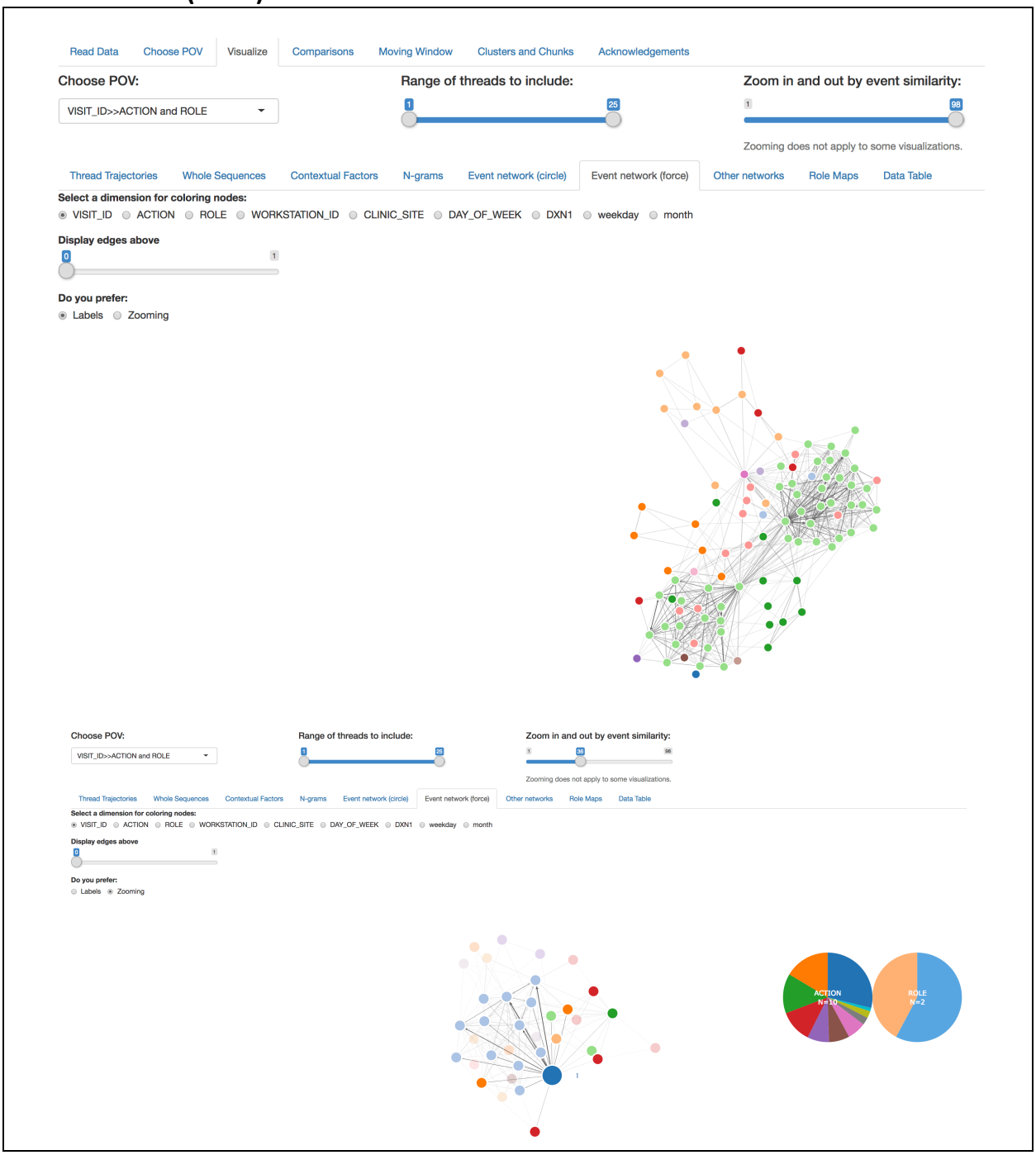
Event Network (circle)



1. This is a weighted, directed graph. The circle layout has a variety of interactive features.
2. You can filter out the infrequent edges to see the strong edges more clearly.
3. If you prefer labels, it gives you the event labels, but you cannot zoom.
4. If you prefer zooming, the labels are no longer meaningful.
5. Zooming can reveal structure in the graph that persists regardless of the granularity used to describe events.
6. These graphs are easier to understand when there are fewer nodes (see below).

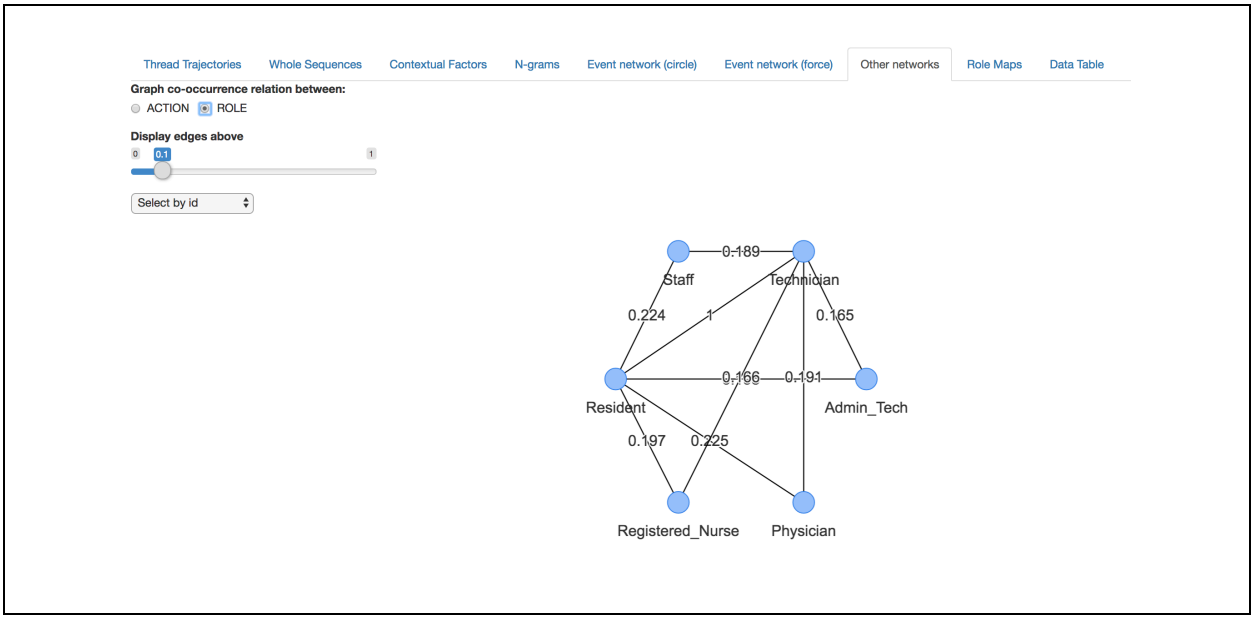


Event Network (force)



1. The force layout (courtesy of networkD3) is the most fun. Depending on our data, the colors of the nodes in the graph may or may not make sense.
2. If you prefer labels, you get labels.
3. If you prefer zooming, you can click on the nodes to see what's inside. As you zoom out, notice that nodes will include combinations of occurrences. The colors in the graph do not match the colors in the pie charts.

Other Networks

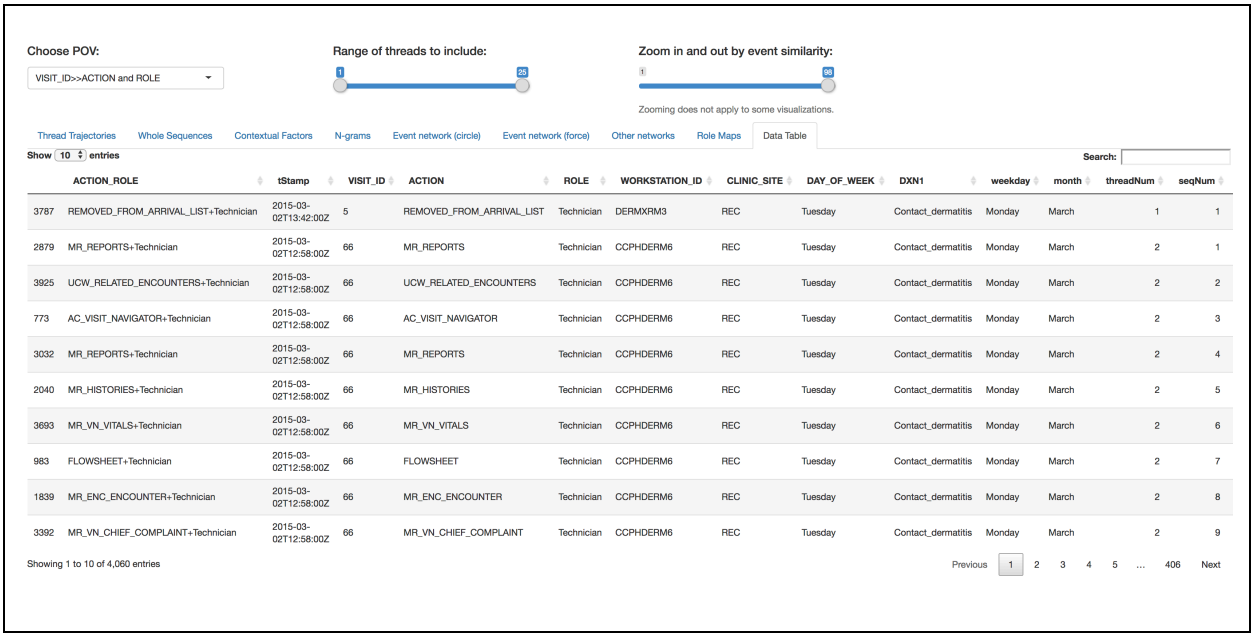


1. You can also graph the co-occurrence of contextual factors within a particular POV. In this example, We can see that resident and technical co-occur in 100% of the patient visits (the weight of that edge is 1). The other roles co-occur at proportionately lower rates.

Role Maps

2. You can also produce a kind of "heat map" for any pair of roles. This shows that the resident and the technician performed most of the actions, and which actions they performed more frequently.

Data Table



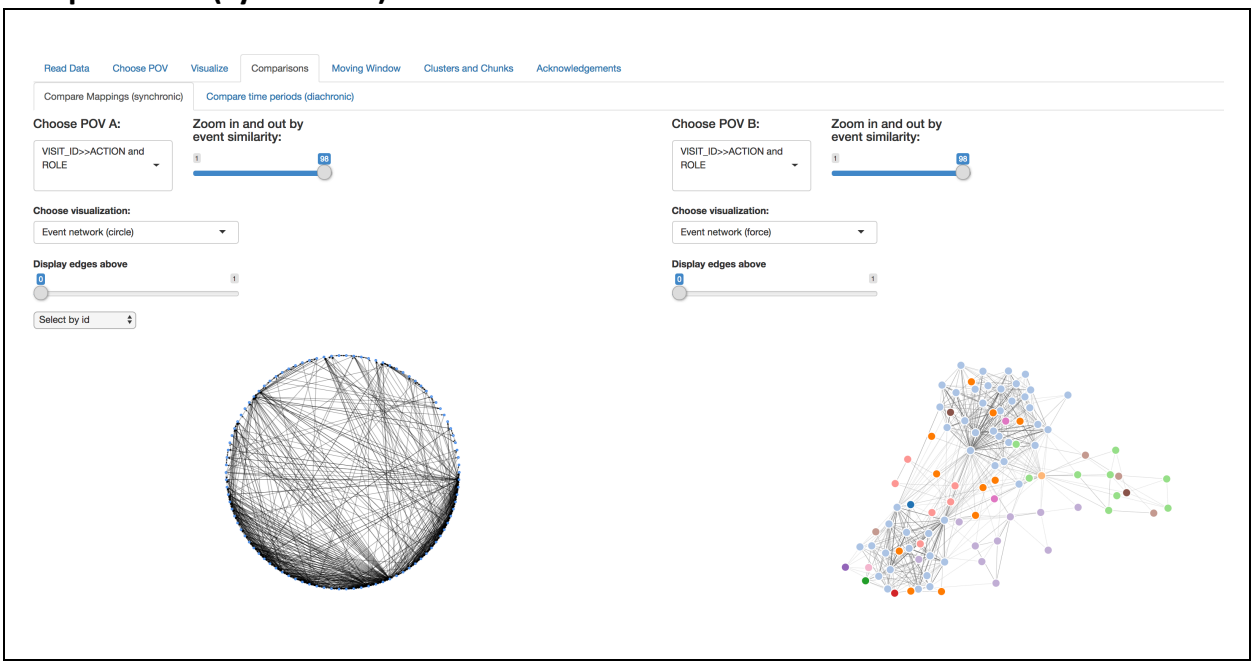
	ACTION_ROLE	tStamp	VISIT_ID	ACTION	ROLE	WORKSTATION_ID	CLINIC_SITE	DAY_OF_WEEK	DXN1	weekday	month	threadNum	seqNum
3787	REMOVED_FROM_ARRIVAL_LIST+Technician	2015-03-02T13:42:00Z	5	REMOVED_FROM_ARRIVAL_LIST	Technician	DERMXRM3	REC	Tuesday	Contact_dermatitis	Monday	March	1	1
2879	MR_REPORTS+Technician	2015-03-02T12:58:00Z	66	MR_REPORTS	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2	1
3925	UCW_RELATED_ENCOUNTERS+Technician	2015-03-02T12:58:00Z	66	UCW_RELATED_ENCOUNTERS	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2	2
773	AC_VISIT_NAVIGATOR+Technician	2015-03-02T12:58:00Z	66	AC_VISIT_NAVIGATOR	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2	3
3032	MR_REPORTS+Technician	2015-03-02T12:58:00Z	66	MR_REPORTS	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2	4
2040	MR_HISTORIES+Technician	2015-03-02T12:58:00Z	66	MR_HISTORIES	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2	5
3693	MR_VN_VITALS+Technician	2015-03-02T12:58:00Z	66	MR_VN_VITALS	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2	6
983	FLOWSHEET+Technician	2015-03-02T12:58:00Z	66	FLOWSHEET	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2	7
1839	MR_ENC_ENCOUNTER+Technician	2015-03-02T12:58:00Z	66	MR_ENC_ENCOUNTER	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2	8
3392	MR_VN_CHIEF_COMPLAINT+Technician	2015-03-02T12:58:00Z	66	MR_VN_CHIEF_COMPLAINT	Technician	CCPHDERM6	REC	Tuesday	Contact_dermatitis	Monday	March	2	9

1. Sometimes it is useful to look at the data being used to construct the graphs.
2. This is display only.

COMPARISONS

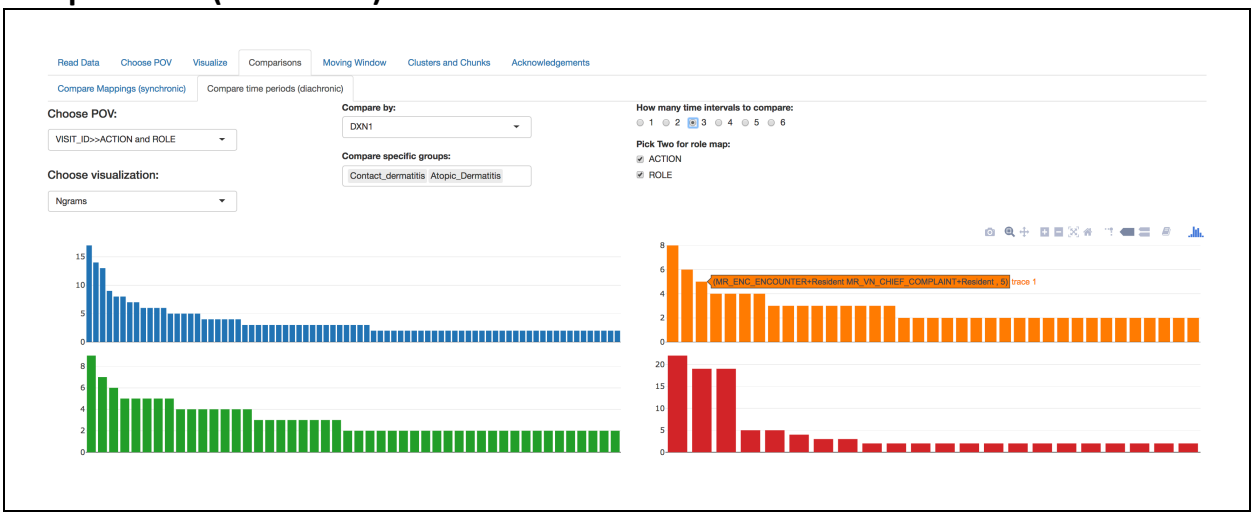
Insight comes from comparison.

Compare POV (synchronic)



1. Pick any two POV and visualize them side by side.
2. Nearly all of the visualizations from the prior section can be used.

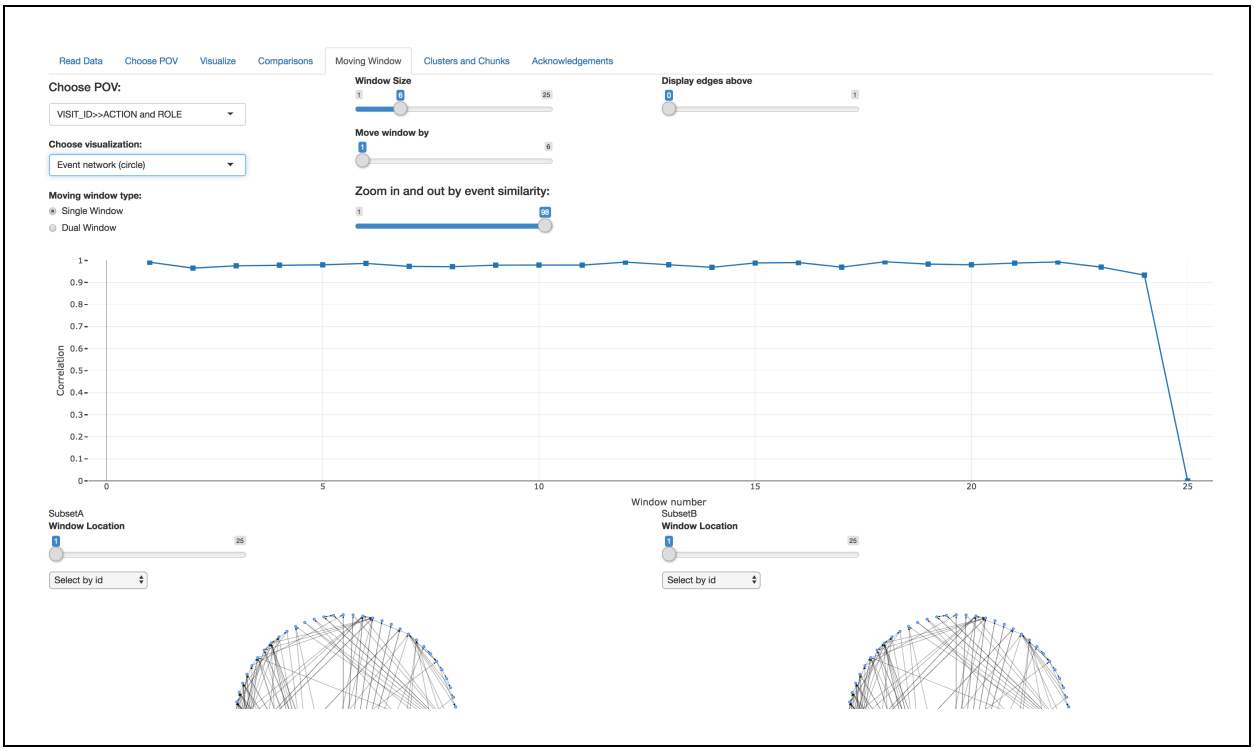
Compare POV (diachronic)



1. Pick any particular POV and break it down over time. The “time intervals” simply divide the total number of threads into equal buckets.
2. Only the visualizations based on Plotly are available here.

MOVING WINDOWS

Processual phenomena are constantly changing. Slide the windows around to visualize change.

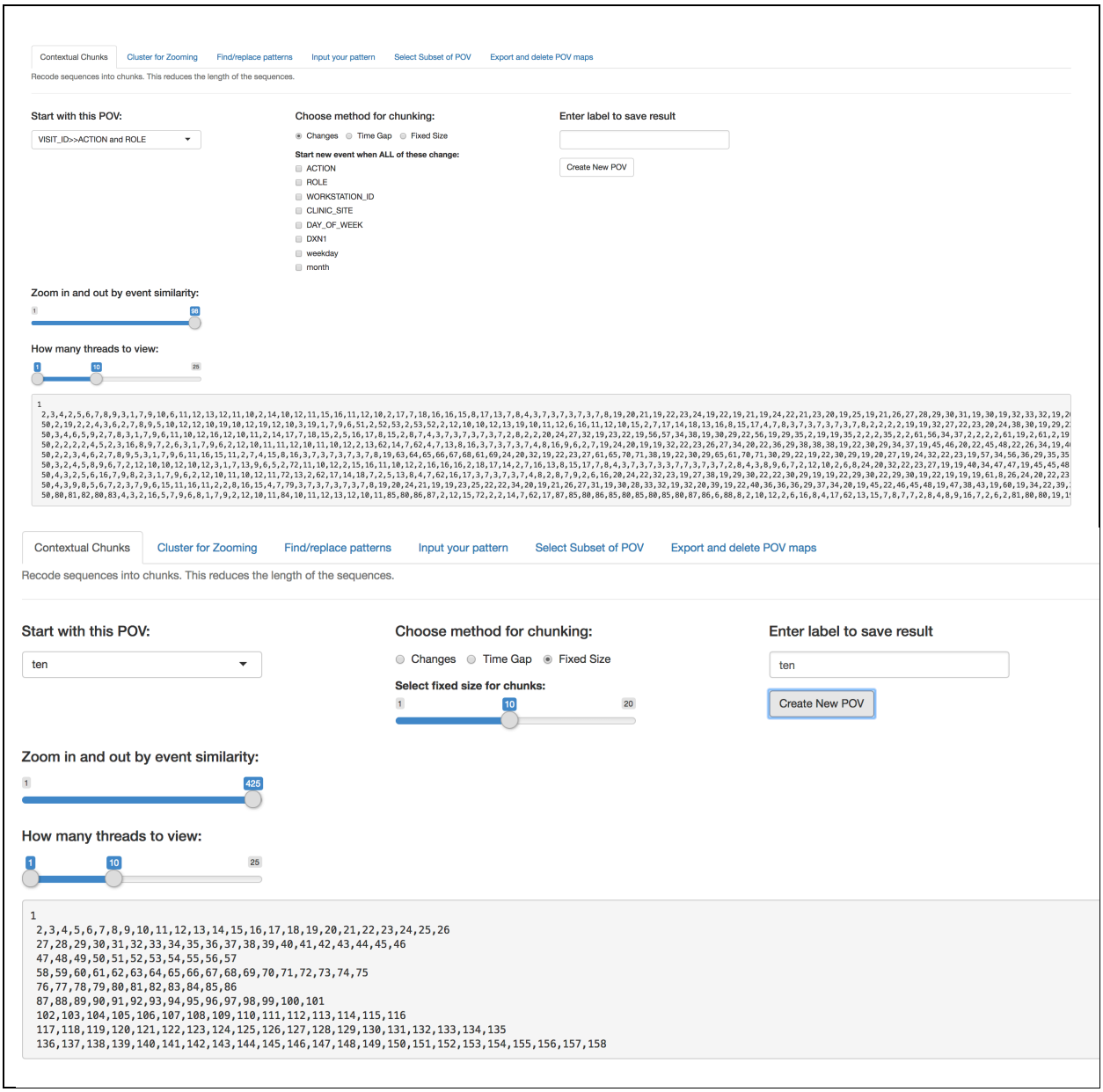


1. See how the threads within one POV are changing over time.
2. Pick one POV and one visualization.
3. The line graph shows how closely the process correlates with itself over time (using the network edge list to compare). The parameters can be set to define how the correleation is computed.
4. The two windows at the bottom can be located independently to show differences.
5. Any visualization can be used.

CLUSTERS AND CHUNKS (experimental)

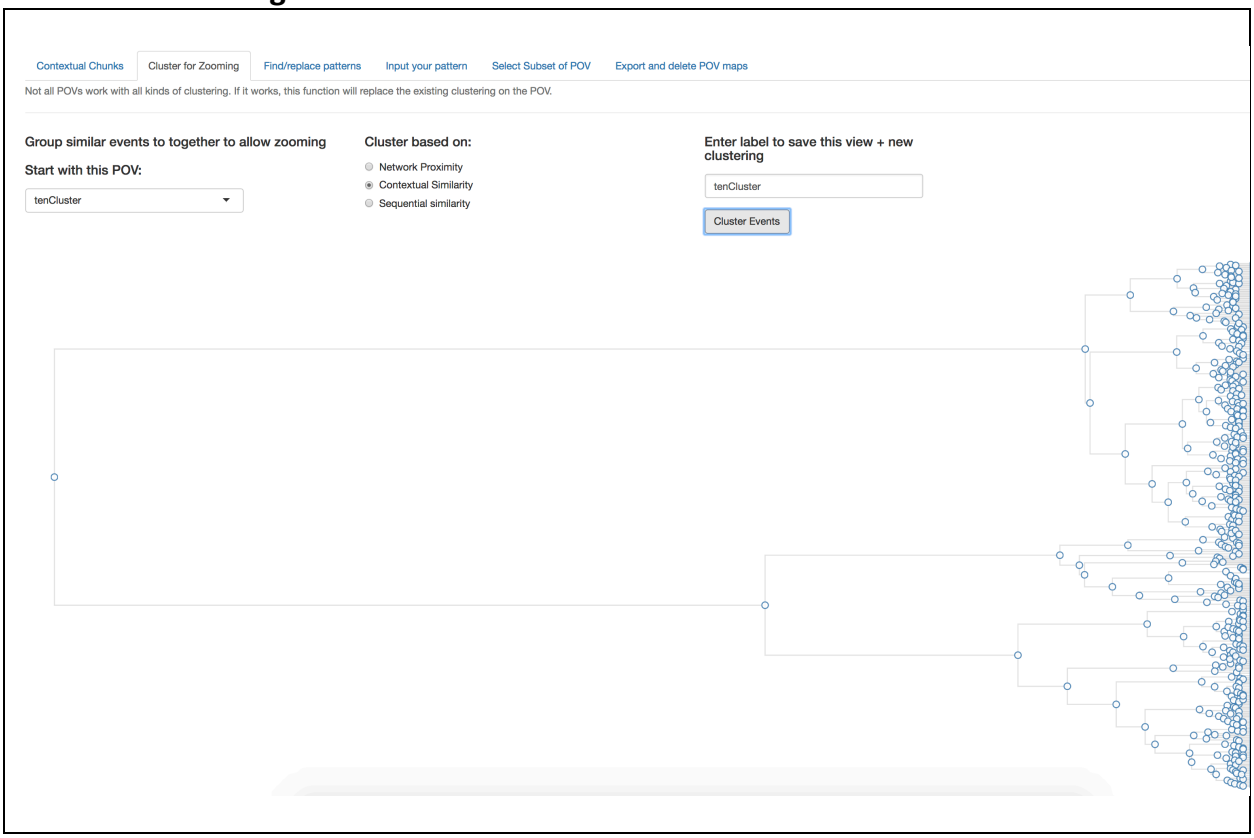
The goal here is use n-grams, regular expressions and clustering to find patterns within the raw data. Those patterns are used to re-code the data into new POVs that can be visualized and compared.

Contextual Chunks



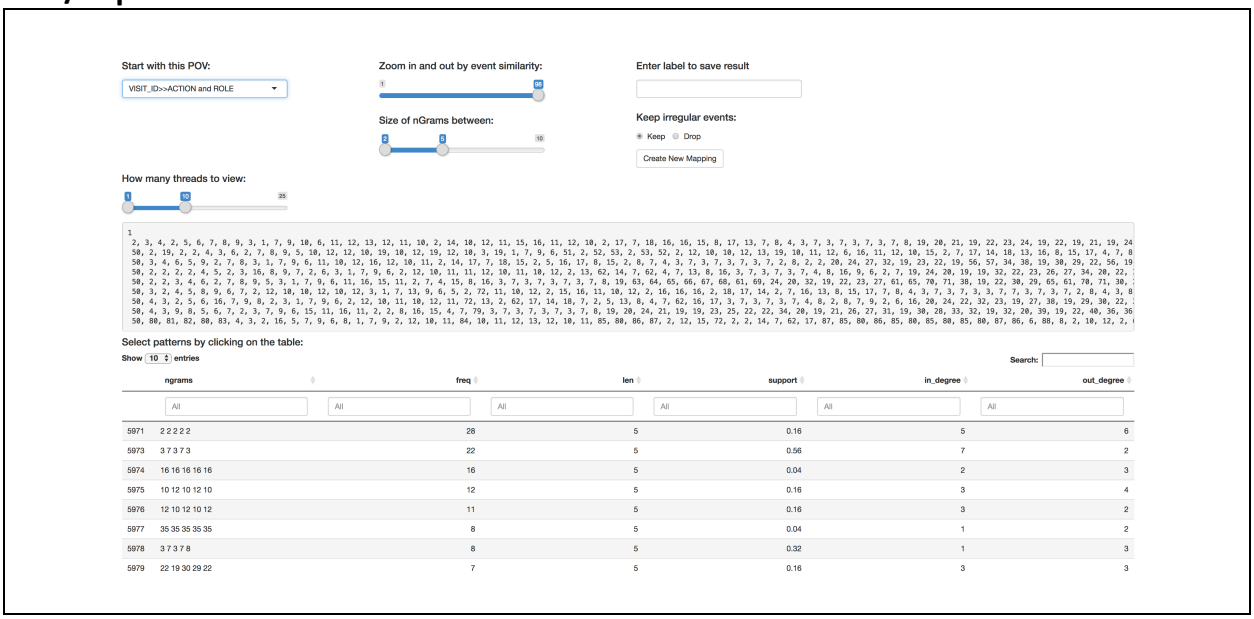
1. For convenience and speed, threads are displayed as sequences of integers.
2. There are three ways of constructing chunks from contiguous subsequences: by changes, by time gaps and by fixed length.

Cluster for Zooming



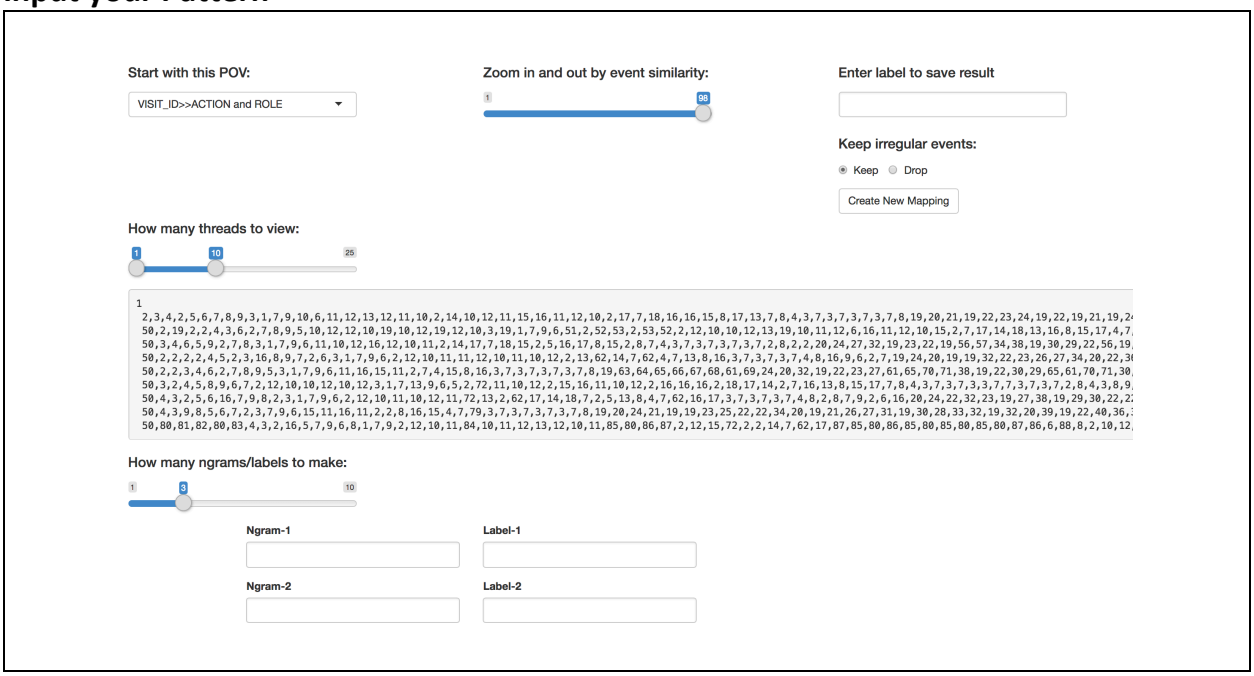
1. Clustering allows zooming, which is basically just moving up and down the cluster solution (from right to left in the above figure).
2. There are currently three methods: my network proximity, by contextual simulatity and sequential similarity.
3. Not all methods work on all POV. For example, sequential similarity only works on a POV that has sequences (not on raw data).

Find/Replace Patterns



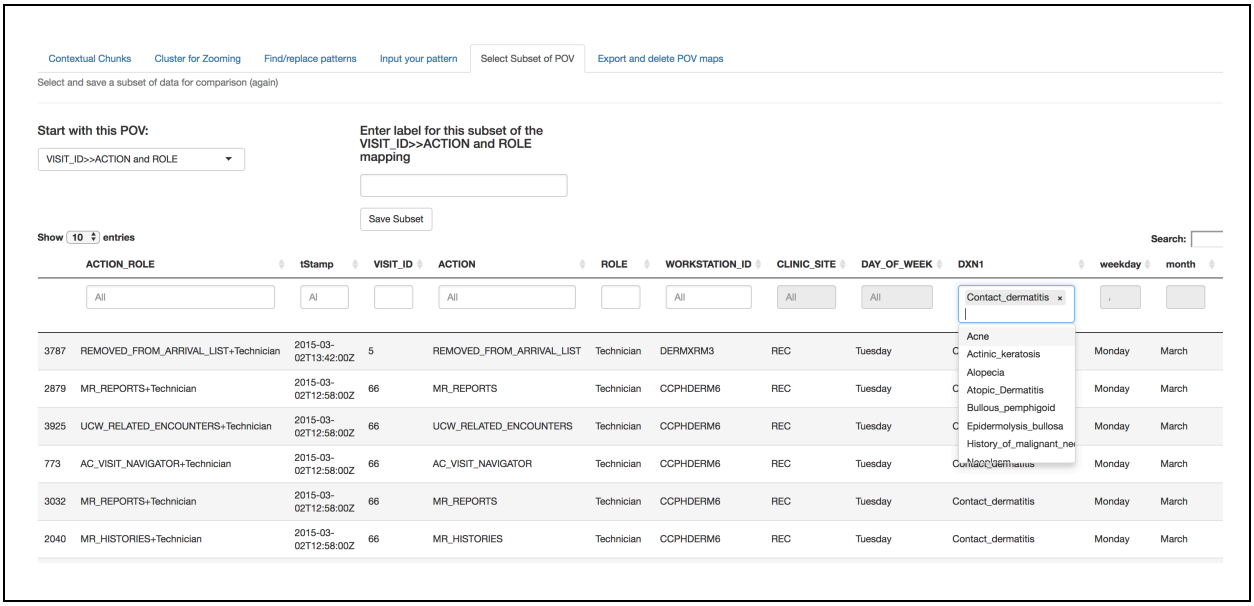
1. This method find n-grams in the data.
2. You can choose which n-grams you want to use to recode the data.

Input your Pattern



1. This method allows you to enter your own regular expressions.
2. You can give expression a new label to automatically recode the data.

Select subset of POV



1. This is just another opportunity to create a subset from any existing POV.

Export and delete POV maps

1. This allows you to export data in CSV or RData format.
2. The RData export also includes a version of the thread data for this POV in TraMineR format.
3. You can also delete any POV here, if you find you have too many.

Acknowledgements

The initial prototype of *ThreadNet* was implemented by Yu Lucy Han, a student in the MS-Business Analytics program at Michigan State University. She designed the clever and efficient method of defining nodes with multiple attributes that is at the core of this software. That part has worked well right from the start and it translated easily from MatLab to R. All the subsequent errors and bugs are my own. Many thanks to Ezra Brooks, Danielle Barnes, Douglas Krum and Morgan Patterson for their expertise in Shiny R; without their help, there would be more bugs and fewer features. And thanks to my colleagues and co-conspirators (Martha Feldman, Thorvald Hærem, Jan Recker, George Wyner, Julie Ryan, Steve Barley, Paul Leonardi, Kent Miller, Mathias Hansson, Peng Liu, Derek Hillison, Waldemar Kremser, Kenneth Goh, Katharina Dittrich, Christian Mahringer, Inkyu Kim and many others) who have been helping figure out what is interesting about action patterns...

References

Feldman, M.S. (2016a) Routines as process: Past, present and future. In Claus Rerup and Jennifer Howard-Grenville (eds.). *Organizational routines and process organization studies.* Perspectives on Process Organization Studies Series, Oxford University Press.

Feldman, M. S. (2016b). Making process visible: Alternatives to boxes and arrows. In Langley and Tsoukas (Eds), *The SAGE handbook of process organizational studies. London, UK: Sage Publications*.

Griffin, L. J. (1992) "Temporality, Events, and Explanation in Historical Sociology: An Introduction", *Sociological Methods & Research* 20(4), pp. 403-427

Nicolini, D. (2009) "Zooming in and Out: Studying Practices by Switching Theoretical Lenses and Trailing Connections", *Organization Studies* 30(12), pp. 1391-1418

Pentland, B. T. and Feldman, M. S. (2007) Narrative Networks: Patterns of technology and organization, *Organization Science,* 18(5), 781-795.

Pentland, B.T., Recker, J. and Wyner, G. (2016) Rediscovering Handoffs, *Academy of Management Discoveries*, *doi:10.5465/amd.2016.0018*

Pentland, B.T. & Liu, P. (2017) Network Models of Organizational Routines: Tracing Associations Between Actions, in S. Jain and R. Mir (Eds), *Routledge Companion to Qualitative Research in Organization Studies***.**

Ryan, J. L., Pinto, D., Pentland, A. P., & Pentland, B. T. (2016). Complexity thermometer: Unraveling complexity of clinic process. *Journal of Investigative Dermatology*, *136*(5), S26.