

Proposal: Judicial Case Law History Timeline  
viewer  
CPSC 547

## Overview

I am working with the local startup Knomos to provide a solution to visualizing the citations (cross-references) between case decisions (written judgements) related to the treatment of a particular legal concepts over time. The goal of this visualization is to aid the goal that Knomos inc. has to create a visual platform for legal research and collaboration - essentially creating a Google Maps for Canadian Law. The data is sourced from publically available legal databases such as Canlii and the BC Laws Database. The Knomos API's backend has parsed through that data and stored it on a Neo4j database server. For the purpose of this project, that data has been made available through a Knomos web API returning structured JSON data. For this project I will restrict the database to the cases stored in the BC Laws database (around 16000 cases).

### Domain Situation:

Canadian Law

### Task:

Visualizing citations

### Dataset:

BC Laws Database

## Personal Expertise

I have no expertise in the research domain, and no expertise in any of the technologies being used. The project will be coded in D3 which is a JavaScript library for information visualization. The technologies used:

D3, JavaScript, JQuery, HTML, CSS, and JSON are all new to me, so there will be an initial steep learning curve.

## Proposed Solution

### Data Munging

The first task is to get the data in the format that is needed. The visualization the Knomos team used as an inspiration for this project, the “Events in Game of Thrones” visualization, used statically stored .csv’s which are quick and easy to work with and have all the complex queries pre-processed. For this project Knomos is using a dynamic data source so it will need to be retrieved from a remote server. So far this has proven to be a tricky task. First I was pointed to the BC Laws Database API and told that I should extract the data from there. This would have required a fair amount of data mining to extract all the necessary data from the provided .XML output. Since this is a Viz course and time is limited, this would have been out of the scope of this project/class and would have taken too much time. Subsequently I was informed that this data has already been parsed and stored in their Neo4j database. I was then given a link to an API returning JSON data to individual cases with the citations parsed into JSON tags, rather than direct access to the database. So far I’ve spent a day trying to access this data through various JavaScript JQuery calls, only to receive the error: No 'Access-Control-Allow-Origin' which apparently has something to do with preventing unauthorized server code to run on the client. I have been in contact with the Knomos server developer and he informed that he will change the server response to allow the client to request the data by tomorrow. So as of today I am still having troubles accessing the data. However, having direct contact with their developers will make a world of difference if there are any future problems.

### InfoVis Solution

The original proposal by Jesse and Adam was to do something similar to *Events in the Game of Thrones* Visualization by Jerome Cukier. They have adapted a mockup (figure 1) to fit the context of Judicial Law.

### Problems with the GoT Viz

In the original, each line between one bubble node to another indicates a kill. The visualization is a very creative way to depict the interactions between Game of Thrones Characters, but it falls short as being useable as production level software. For example, there is no indication of who killed who (ie. which direction) between nodes connected by lines. The viz is also crowded with no labels on the bubbles, you must hover over a node to determine the character’s name and then hover over the node that it links to, to determine who that character killed or was killed by.

## Proposed Adapted GoT Viz

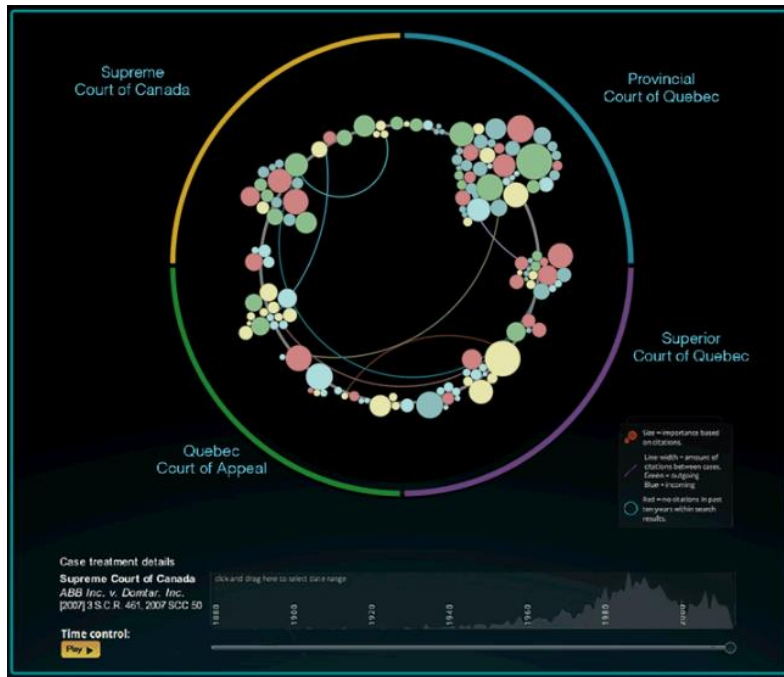


Figure 1: Knomos Inc's proposed visualization based on the Events in Game of Thrones Visualization

1. Each node represents a case decision, clustered by jurisdiction (eg. Supreme Court of Canada)
2. The size of the node represents the frequency of the case's citations throughout other cases decisions
3. The colour of each node signals the relative treatment of that case decision in other cases (eg. Green = favourable, yellow = distinguished, red = overturned)
4. The lines between nodes represent interactions (ie. The citations) between case decisions
5. The time control scrollbar can be shifted to adjust the viz, and stopped at a specific point in time
6. Content scope can be filtered by including or restricting the years or jurisdictions selected beneath the scrollbar.

Currently we are limiting the scope to only the the BC Laws database. This will make the project more feasible for the short time line we have.

## Problems with Adapting the GoT Style Viz

In the original GoT viz the nodes were grouped by families and kills mostly happened between one family to another. In the Judicial cases this is likely not going to be the case. It is highly likely that all

cases could all pertain within one jurisdiction so grouping the cases this way with lines (citations) looping back to the same group would be difficult to distinguish. Additionally, for this project we are limiting our data to one jurisdiction so the depicted clustering will not work.

With the case of kills, a character is *usually* killed only once, meaning there is only line coming into it (though possibly multiple lines coming out?). In the case of law citations, there is likely to be a many to many relationship. With the large clusters in the GoT viz it is hard to see where the lines are coming in and out.

## The Proposed Viz

Given all these issues and limitations my proposal will revolve around d3.js Chord Layout visualization.



Figure 2 d3.js Chord Layout Visualization

In particular, I would like to take inspiration from the Dependency Wheel adaptation of the chord layout:

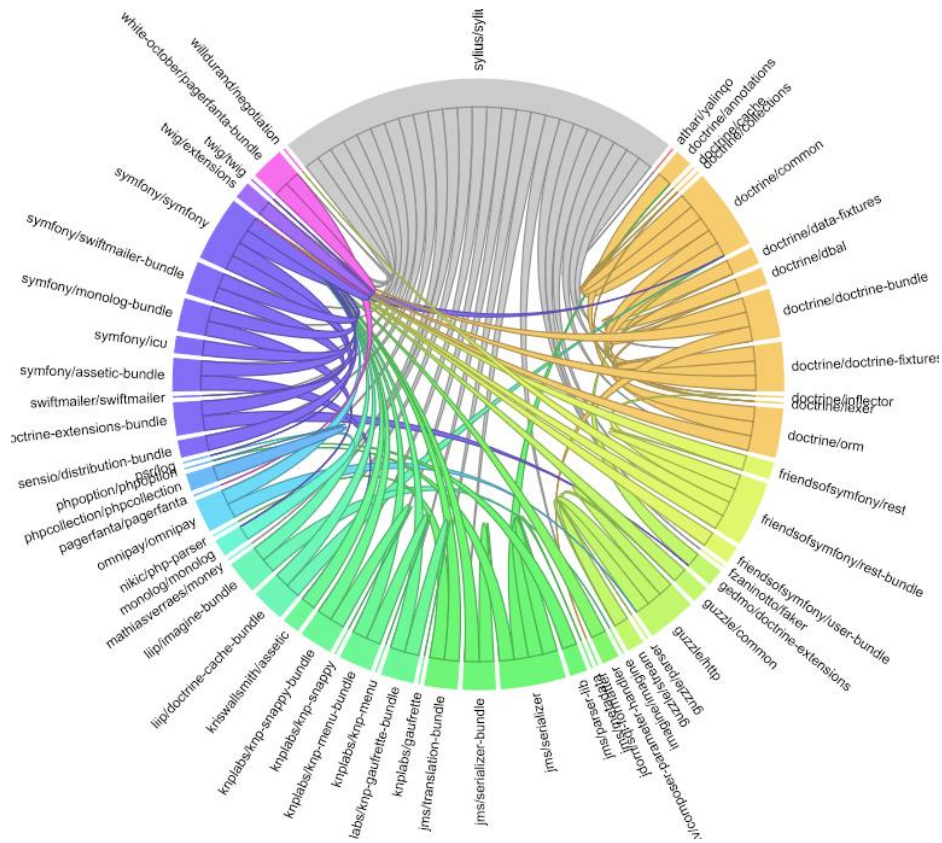


Figure 3: d3 Dependency Wheel

<http://www.redotheweb.com/DependencyWheel/>

This layout has a much cleaner look than the GoT viz, in particular all chords are labelled. In the overall view there still exists some occlusion of the links between the nodes. This is alleviated by hovering over one of the chords to single out its connections.

With a chord highlighted:





and/or pre-process the data. I do not currently have server access so hopefully this will not be necessary.

The next step would be to associate each case with a timestamp and allow for timeline animation. The GoT viz uses static data so it is trivial to make the timeline start at the beginning of the first book and end at the last. For the judicial cases the timelines could start and end any time within the last ~100 years. We would make the time line extend from the first citation to the last.

Scalability: the chord view appears to be able to handle around ~100 items with ease. More than that would require some sort of clustering to group similar cases – perhaps grouping by outcome. A better way of scaling the data might be to use the “reduce” idiom and only show cases close to the current time on the time scale. On the other hand, there may be some cases with very few citations.

### Visual Encoding

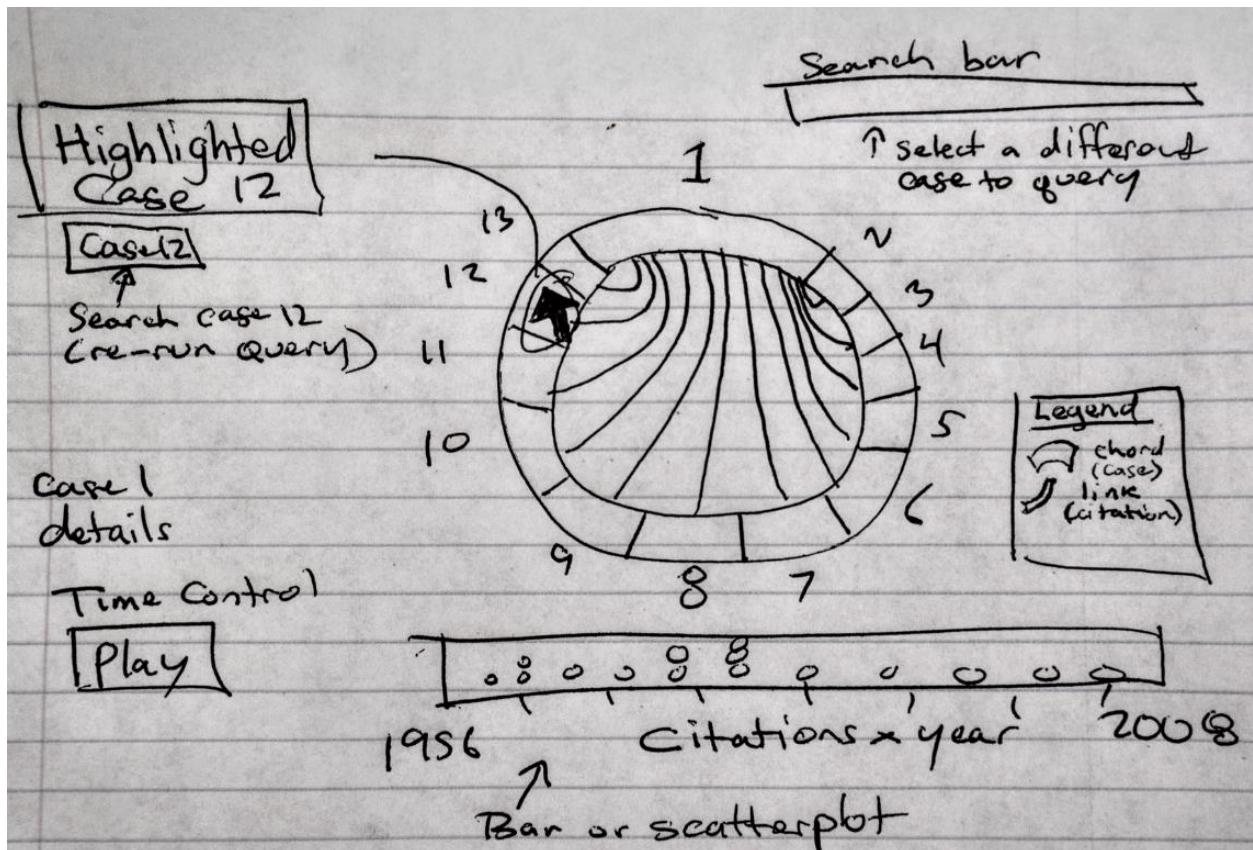


Figure 5: Sketch of overall info vis using the Chord Layout



The primary idea would be to encode the cases in the chords and to encode the citations via the lines linking the chords together.

Mousing over a chord would use Highlighting to manipulate the view so that we can clearly see the links. It would also show a window with details of that case.

The view will also be manipulated by allowing a change over time idiom to occur via the play button. As time passes we will see new citations appearing linking cases together.

We also embed a citation vs time view at the bottom.

## A Scenario of Use

The Knomos team has already designed a full suite of visualizations to make it easier for lawyers to find and cross reference cases. This visualization would just be an additional tool for legal workers to use during their research. I presume that this would be used as a browser to see which cases over time have shared citations and perhaps how some judgements change over time. The actual workflow of a lawyer still remains a mystery to me.

## Milestones

- Mon 11/16 Have the JavaScript code to query and parse all the data complete.
- Mon 11/23: Status Updates Due. Hope to have a working static (no interactivity) prototype of the main visualization complete.
- Mon 12/7 Completely working main visualization including full interactivity.
- Mon 12/15 Final Presentation: Finalize all secondary views (legend, details, buttons, formatting). Polish.
- Fri 12/18 Final Paper.

## Previous Work

<http://knomos.ca/>

## Appendix:

Sample JSON API response.

<http://app.knomos.ca/api/cases/bcca/2013/173/citations?pprint=t>

The year and the # (2013/173) specify the unique case. JSON response below:

```
{  'cited_by': {  'general_case_citations': [  {  'source_case': {
'case_num': 285,

'citation': '2015bcca285',

'date': '2015-06-25',

'name': '2015 '

'BCCA '

'285',

'pretty_citation': '2015 '

'BCCA '

'285',

'soc': 'R. '

'v. '

'Voong',

'title': '',

'tribunal_id': 'bcca',

'year': 2015},

Etc.
```

Sample JavaScript JQuery code to parse the data:

```
var xmlhttp = new XMLHttpRequest();
var url = "http://app.knomos.ca/api/cases/bcca/2015/285/citations?";
xmlhttp.onreadystatechange=function() {
    if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
        myFunction(xmlhttp.responseText);
    }
}
xmlhttp.open("GET", url, true);
xmlhttp.send();
function myFunction(response) {
    var arr = JSON.parse(response);  });
```