Visualization of the Phosphoproteomic Data from AfCS with the Google Motion Chart Gadget

Huilei Xu[1], and Avi Ma'ayan[1,*]

[1]Department of Pharmacology and Systems Therapeutics, Mount Sinai School of Medicine, 1425 Madison Ave. New York, NY, 10029

[*]To whom correspondence should be addressed at avi.maayan@mssm.edu

Motivation

Results from multivariate molecular biological experiments become increasingly complex. Hence, the challenge of projecting high-dimensional data onto few dimensions for effective data visualization is becoming increasingly important in Systems Biology. Effective data visualization can summarize the activity of many variables over time as well as display relationships between variables. Dynamic interactive visualization tools can provide scientists with ways of visually identifying relationship and patterns, and improve communication of results on the web and in presentations. For this, interactive systems with animation have great potential since they add dimensions to static images limited to two dimensions. Interactivity and animation is particularly useful for showing time-series trends in multi-dimensional data.

Introduction

The Flash-based Motion Chart Google Gadget available through GoogleDocs is a recent advance in multi-dimensional data visualization. The Motion Chart Gadget is a component of the Trendalyzer software, which was developed for web-based animation of statistical results. The software was initially developed by Hans Rosling's from the Gapminder Foundation [1]. Gapminder Foundation was acquired by Google Inc. in 2007 and the Motion Chart gadget was recently added as a feature for the GoogleDocs Spreadsheets. This gadget drew positive attention for its usefulness for visualizing data from diverse disciplines such as socioeconomics. Here we demonstrate the use of this Gadget to visualize molecular biological data, the phosphoproteomics results published on the Data Center of the Signaling Gateway web-site: http://www.signaling-gateway.org/ [3].

Methods and Results

The Alliance for Cell Signaling (AfCS) has published phosphorylation fold-change of 21 signaling molecules after stimulation of serum starved RAW 264.7 macrophage-like mouse cells with 22 different ligands and pairwise combinations of these ligands over a time-series of 0, 1, 3, 10 and 30 minutes after stimulation [2]. The data is available on the Signaling Gateway Data Center web-site http://www.signaling-gateway.org [3]. Although visitors to this site can have access to view graphs that visualize the time-course data, i.e., by clicking on the ligand(s) of interest and then clicking on the specific phosphoprotein experimental results, this approach for data visualization is static and requires 422 plots to show the fold-change of all the phosphoproteins over time in response to the different specific ligands. Since the user must navigate from page to page to view the 422 diagrams, it is difficult for to see the fold-changes of all the 21 phosphoproteins at once for acquiring a global perspective. Hence, here we use the Motion Chart tool to visualize this multi-dimensional dataset. We employed the Flash-based Motion Chart Google Gadget for visualizing all 422 plots within one chart (Fig. 1A). We processed the data-files from AfCS using a Perl script (Script 1) to construct tables that can be loaded to the spreadsheet feature of GoogleDocs. We then applied the Motion Chart Google Gadget on this data set. The global dynamic fold-change of the cellular signaling proteins is captured within this one web-based dynamic graph with the option for user interactivity.

We also used the Motion Chart gadget to explore the dynamics of the 21 signaling molecules in response to the 22 ligands using principle component analysis (PCA). PCA is a mathematical method used to reduce the dimensionality of a dataset by aligning vectors in N-dimensional space for data compression and data clustering. Such analysis allowed us to compress the 21 phospho-protein measurements into two components/variables: the two most important principle components. These two variables represent a linear combination of the 21 original variables. We used this technique to track and cluster the global dynamical behavior of the cells in response to the different ligands over time. Using another prel script (Script 2) and few Matlab commands (see below), we computed the principle components for the combined experiments extracted from the AfCS data-files. For this analysis we plotted the first and second principle components for each ligand over the 0, 1, 3, 10 and 30 minute time course (Fig. 1B). Indeed, we identified three clusters, one that appears to be the same as the initial condition and this is where most ligand effects appear to settle into after 30 minutes; and two other clusters which are: IFNA, IFNB, IFNG and IL6 which show very similar and distinct dynamics over time, whereas 848, P3C, P2C follow a different trajectory where they seem to converge into the same global state at 30 minutes. We conjectured that this dynamics maybe explained by these ligands pushing the system away from an initial stable state and after some time the system settles back into a new stable state (new attractor). It is also noteworthy to mention that LPS is also close by but not all together with 848, P3C, and P2C cluster/attractor. These results are consistent with known macrophage and other immune cells cell-signaling biology: LPS, 848, P3C and P2C mimic early stage signaling of the detection process of a foreign object. The signaling molecules that dominate at this stage are JNK and p38. Later on, such cascade induces the production of interferon which then, through an autocrine loop activates interferon receptors that signal through the JAK/STAT pathway. From the combined analysis such patterns can be easily discerned.

In summary, our revisualization effort of the AfCS data provides a proof of concept that the Motion Chart is a useful tool for exploring multidimensional time-series data in biological context. The tool, and similar future gadgets, has the potential to be very useful for analysis and interpretations of multivariate results in Systems Biology.
The gadgets available at: http://spreadsheets.google.com/ccc?key=pjeGLzR6KSZ60dgSrpJhfTg for all ligands interactive visualization and
http://spreadsheets.google.com/ccc?key=pjeGLzR6KSZ7QmudhAzc9NA&hl=en for visualizing the principle components of all the single ligand experiments.

Figures

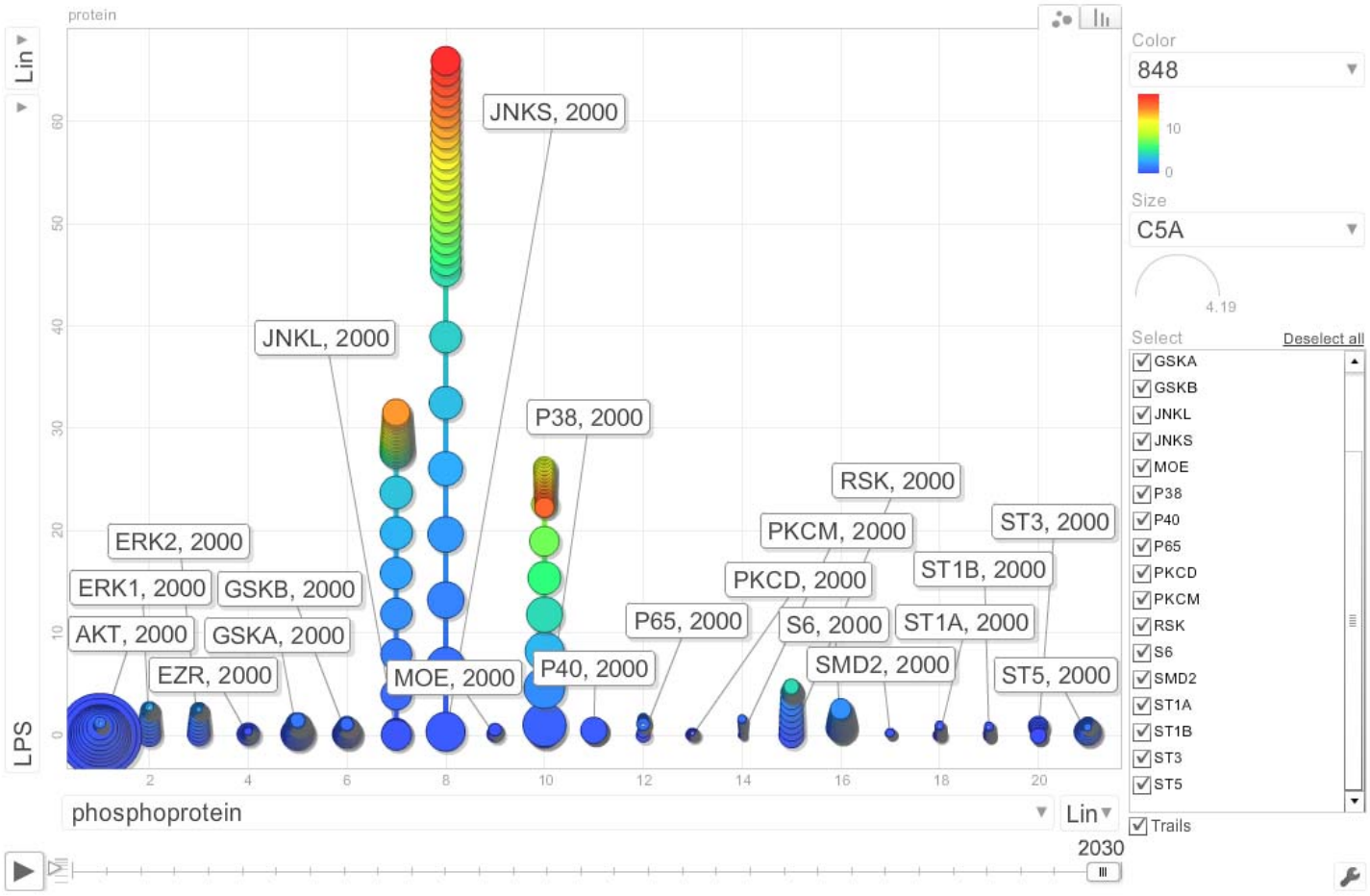

Fig. 1A Visualization of the temporal dynamics of phosphor-proteins in response to different ligands. The x-axis stands for the 21 signaling phosphoproteins. The y-axis provides the choices for one of the 22 different ligands. The time-series stands for the minutes passed: for example, 2001 means time 0 min, 2031 means 30 mins. With this beta-version of Motion Chart we could not use time in minutes. Lin/Log provides the option to view the foldchange in linear or log format.
This gadget is available at:
http://spreadsheets.google.com/ccc?key=pjeGLzR6KSZ60dgSrpJhfTg

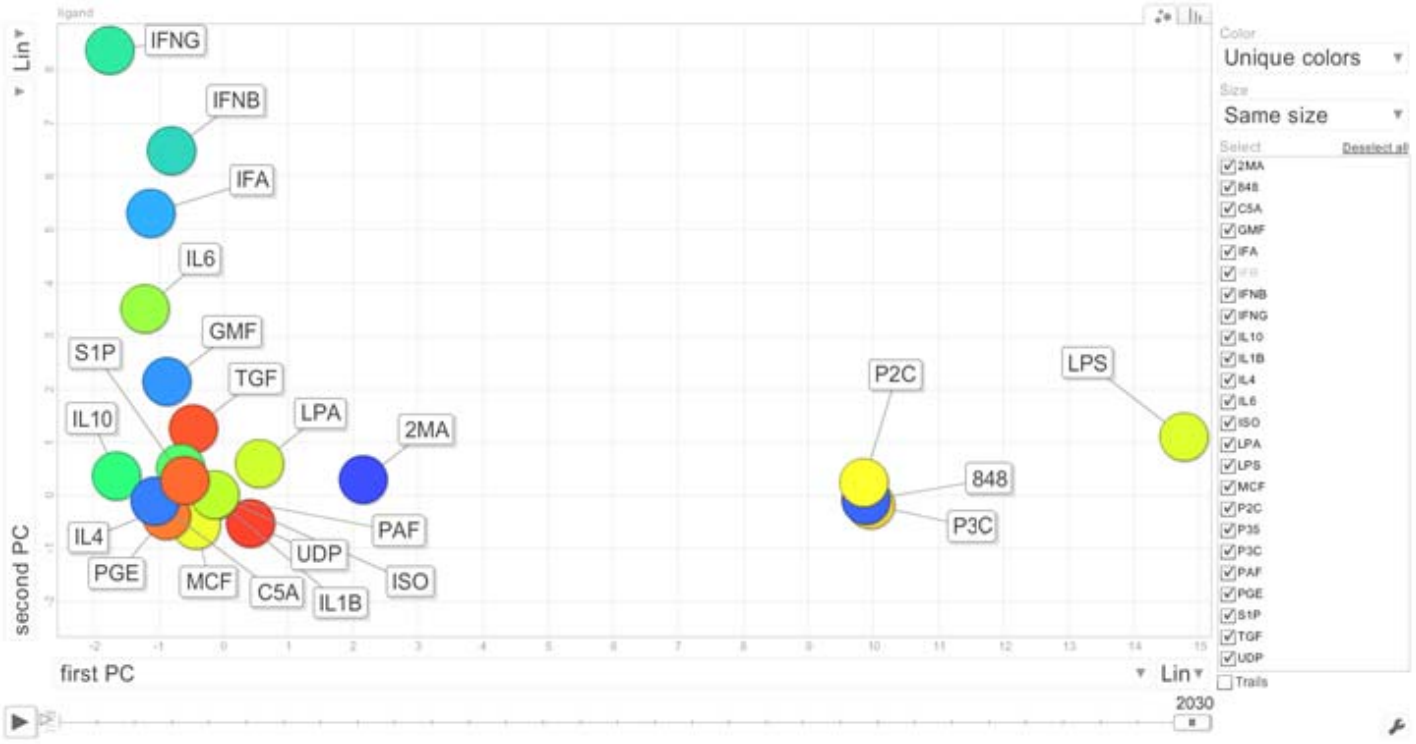

Fig. 1B Dynamic visualization of the 21 signaling molecules compressed into two principal components in response to the 22 ligands using PCA. This gadget is available at: http://spreadsheets.google.com/ccc?key=pjeGLzR6KSZ7QmudhAzc9NA&hl=en

Additional Materials

Script 1

The following is a Perl script used to convert the single ligand phosphoproteins data from the AfCS Data Center stored in text files into a format that can be uploaded into GoogleDocs to be used by MotionChart.

```
#!/perl/bin/perl
@files = `ls files/\*`;
foreach $file (@files)
{
   $index = 0;
   $file =~ s/\r//g;
   open (IN, "<$file");
   @lines = <IN>;
   foreach $line (@lines)
   {
         @splitted = split (/\s+/, $line);
         $flag = 0;
         for ( $i = 0; $i < $index; $i++ )
         {
             $temp = join ('_', $splitted[4] , $splitted[6]);
              if ( $temp eq $entries[$i] )
              {
                   $avg[$i]=$avg [$i]+$splitted[9];
                   $n[$i] = $n[$i]+1 ;
                   $flag = 1;
              }
        }
        if ($flag == 0)
        {
           $n[$index]=1;
           $avg[$index]=$splitted[9];
           $entries[$index] = join ('_', $splitted[4] , $splitted[6]);
           $index++;
        }
    }
    ($name1, $name2) = split (/\//, $file);
    ($name3, $name4) = split (/\./, $name2);
    $outfile =  $name3 . ".csv";
    open (FINAL,">$outfile") or die "can't open";
    for ( $a = 0; $a <= scalar(@entries) - 1; $a++ )
    {
                    $avg[$a]=$avg[$a]/$n[$a];
                    ($protein, $time )= split(/_/,$entries[$a]);
                    $time = ($time / 60) + 2000;
                    print FINAL "$protein, $time, $avg[$a], $n[$a]\n";
    }
    close(FINAL);
    close(IN);
}
```

Script 2

The following is a Perl script used to convert the table used to visualize the data in Fig. 1a into a format that can be imported into MatLab for the principle component analysis.

```
#!/usr/bin/perl
open (IN, "data.csv");
@list = <IN>;
$time_point = 0;
$index = 0;
for ($i = 0; $i < 22; $i++)
{
        for ($j = 0; $j < 6; $j++)
        {
                $time_point = 0;
                $index = 0;
                foreach $item (@list)
                {
                        if (($index == 0) && ($j != 5))
                        {
                                @names = split(/,/, $item);
                                $name = $names[$i + 3];
                                $name =~ s/\r//g;
                                $name =~ s/\n//g;
                                print $name;
                                print $j + 1;
                                print ", ";
                        }
                        else
                        {
                                @line = split(/,/, $item);
                        }
                        if ($time_point == ($j + 1))
                        {
                                print " ";
                                $line1 = $line[$i + 3];
                                $line1 =~ s/\r//g;
                                $line1 =~ s/\n//g;
                                print $line1;
                                print ", ";
                        }
                        $time_point++;
                        $index++;
                        if ($time_point == 0)
                        {
                                $time_point = 1;
                        }
                        if ($time_point == 6)
                        {
                                $time_point = 1;
                        }
                }
                if ($j != 5)
                {
                        print "\n";
                }
        }
}
```

Matlab commands

The following Matlab commands were used to generate the two principle components.

```
>> sr = data./repmat(stdr, 110, 1);
>> [coef,scores,variance, t2] = princomp(sr);
>> output = scores(:, 1:2);
```

Acknowledgements

We would like to thank Ben MacArthur for useful suggestions.

Funding: This research was partially funded by NIH Grant No. P50GM071558 Systems Biology Center in New York (SBCNY).

References

1. Ronnlund AR, Rosling O (2004) Free software for a world in motion. *Second International Conference on Creating, Connecting and Collaborating through Computing*, 29-30 Jan. 2004

2. Natarajan M, Lin K-M, Hsueh RC, Sternweis PC, Ranganathan R (2006) A global analysis of cross-talk in a mammalian cellular signalling network. *Nat Cell Biol*, **8**, 571-580.

3. Saunders B, Lyon S, Day M, Riley B, Chenette E, Subramaniam S (2008) The Molecule Pages database. *Nucl Acids Res*, **36 (suppl_1)**, D700-706.