Simplifying the Creation and Deployment of Collaborative Data Analysis and Visualization Tools

Emanuele Santos

Advisors: Claudio Silva and Juliana Freire

Dissertation Defense
17 June 2010

Thesis Statement

Effective exploration of scientific data requires a **combination of novel software and frameworks** that provide users with different **goals and levels of expertise** the ability to **create**, **modify** and **share** their **own analysis and visualization workflows and tools**

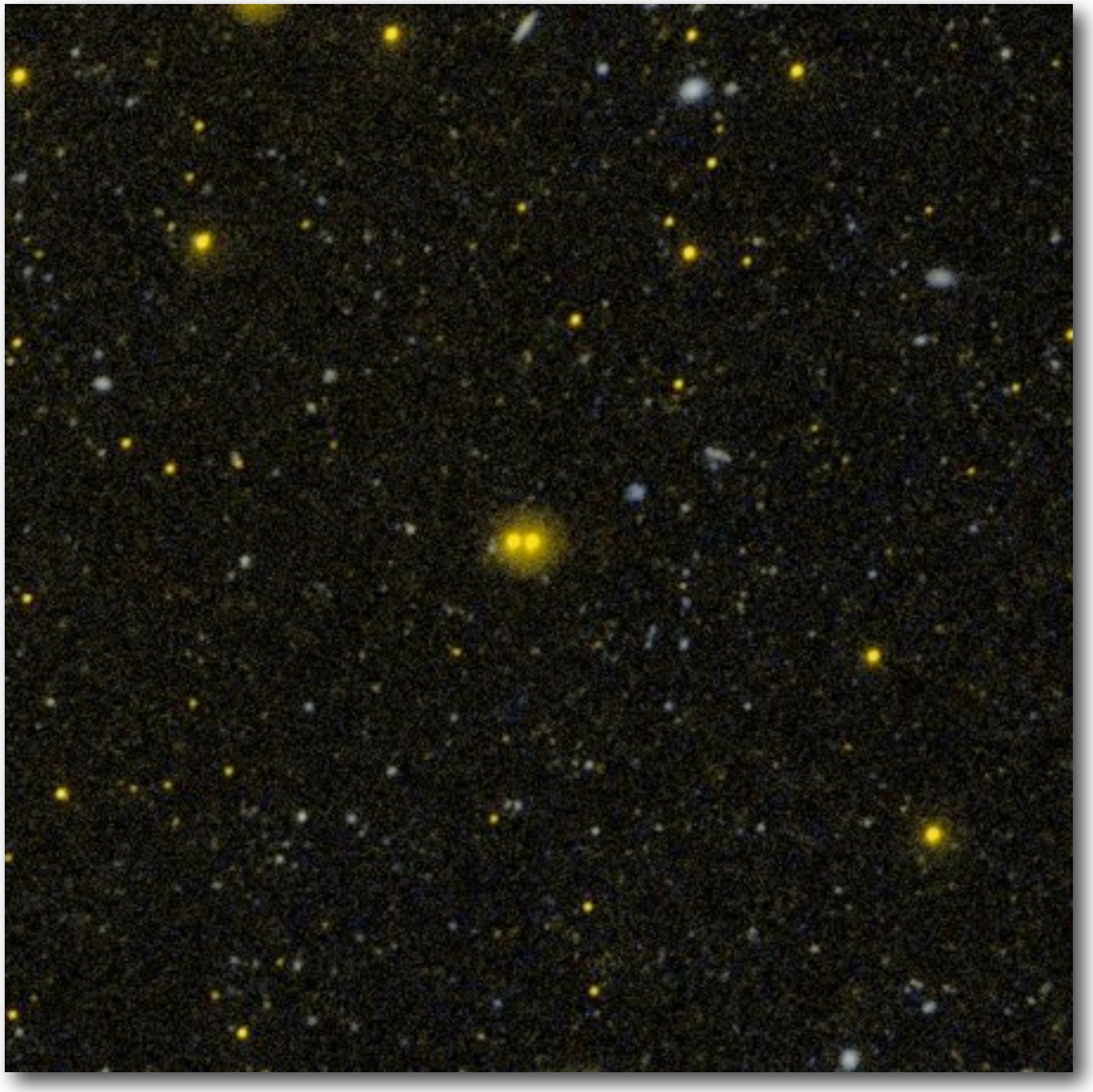
Motivation

- Data exploration through visualization has become very important for scientific research
- Challenges faced by computational scientists
  - Most of them are not familiar with advanced visualization techniques and tools
  - The printed medium is not adequate for reporting complex computation results
  - Collaboration among large interdisciplinary groups
- Example in Astrophysics

The routine of an Astrophysicist

?

Goal: Understand various astrophysics phenomena

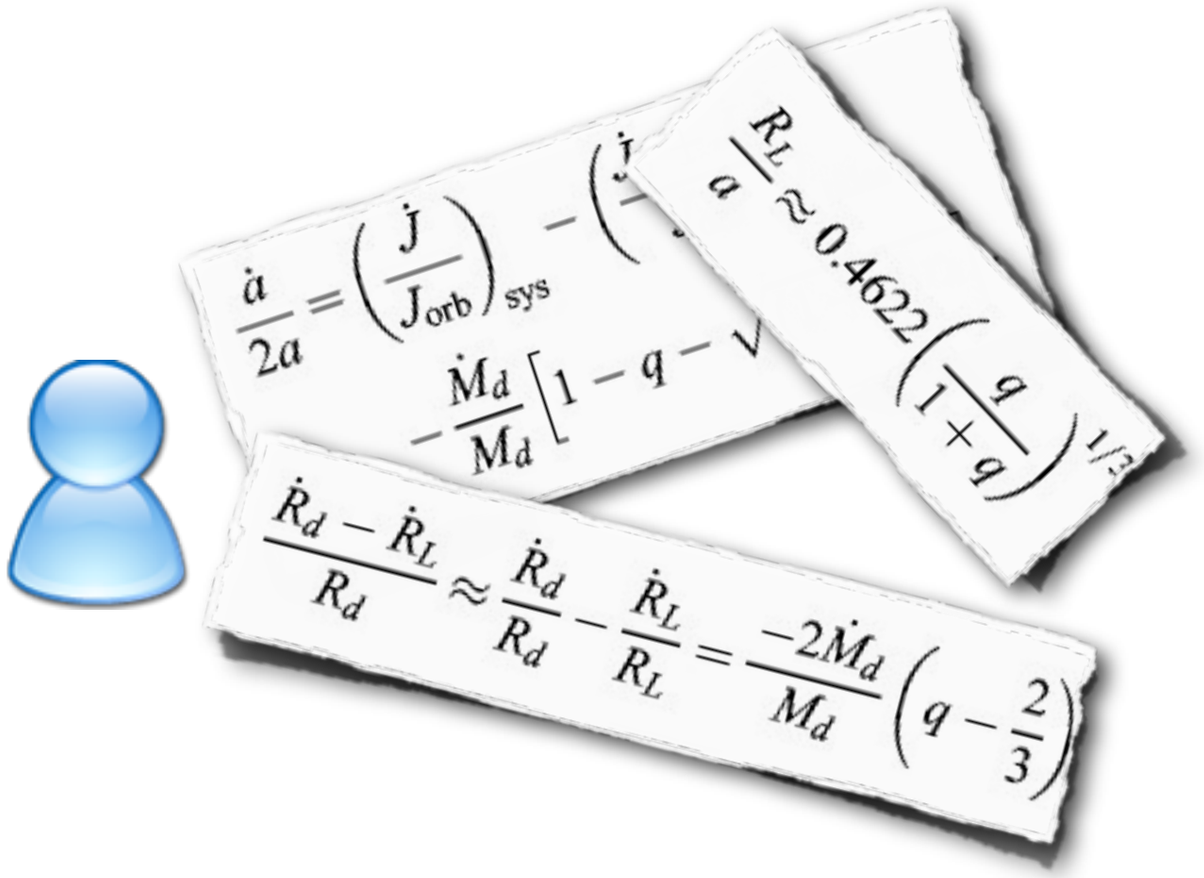


Binary Star Systems

The routine of an Astrophysicist
Developing algorithms

Simulation of mass-transfer instabilities in binary star systems

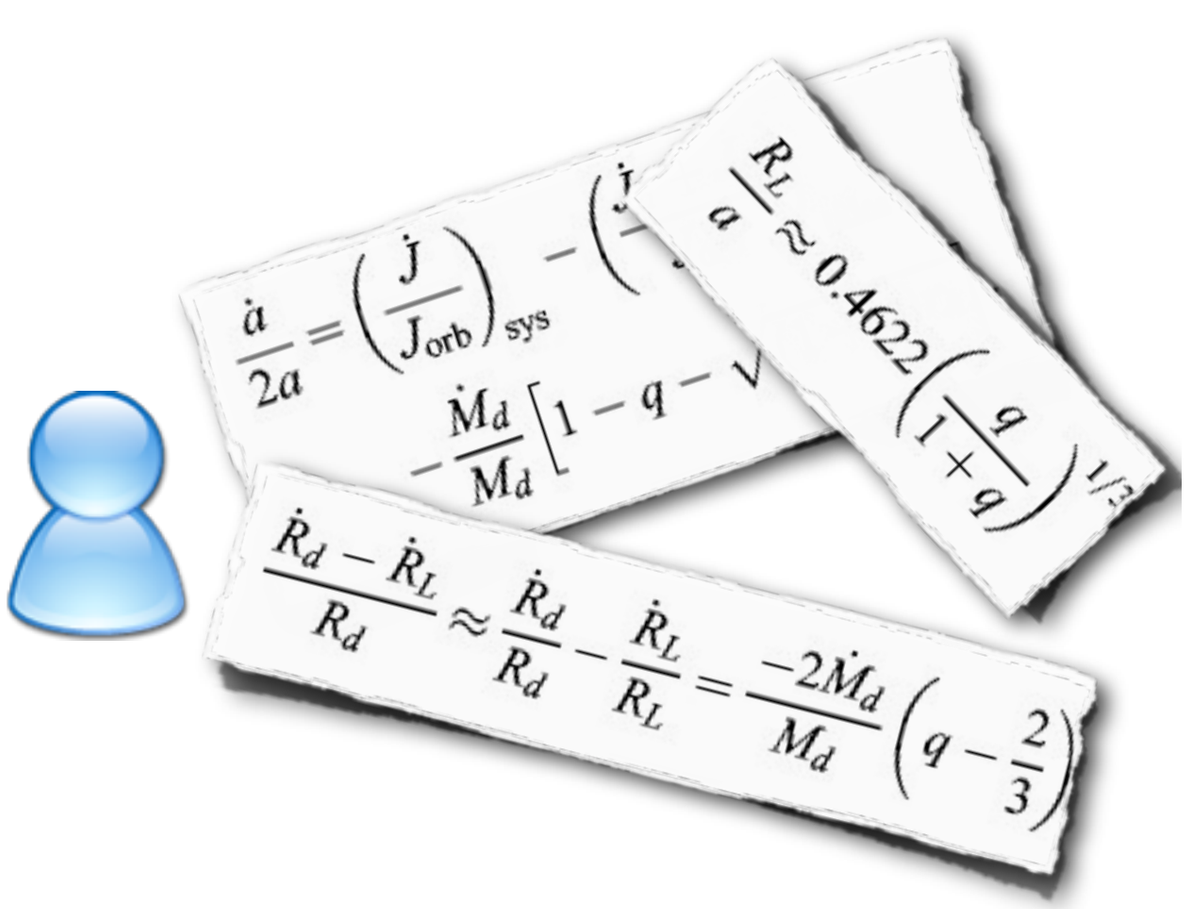
The routine of an Astrophysicist
Developing algorithms



Equations

Simulation of mass-transfer instabilities in binary star systems

The routine of an Astrophysicist
Developing algorithms

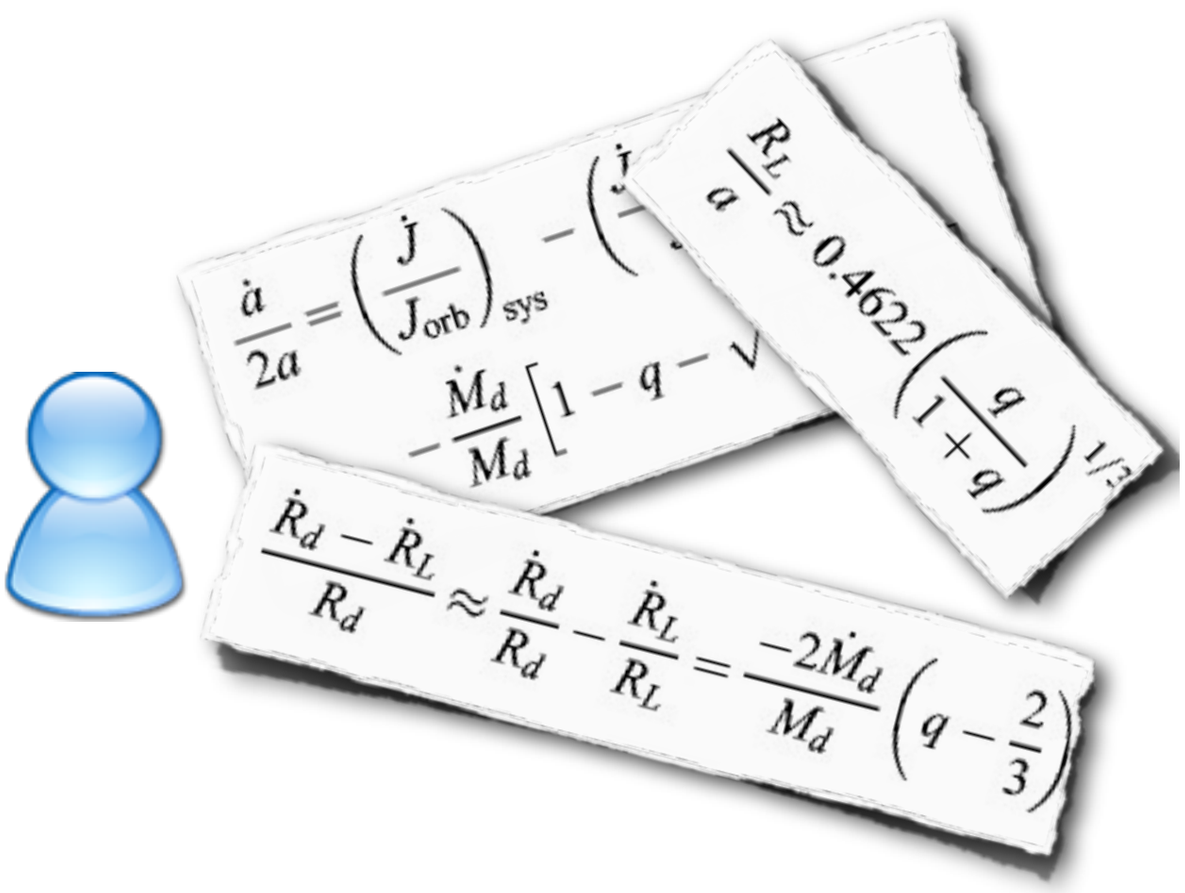


Equations

Code

Simulation of mass-transfer instabilities in binary star systems

The routine of an Astrophysicist
Developing algorithms



Equations

Code

Data

Simulation of mass-transfer instabilities in binary star systems

The routine of an Astrophysicist
Data Analysis and Visualization

Explore results of simulations

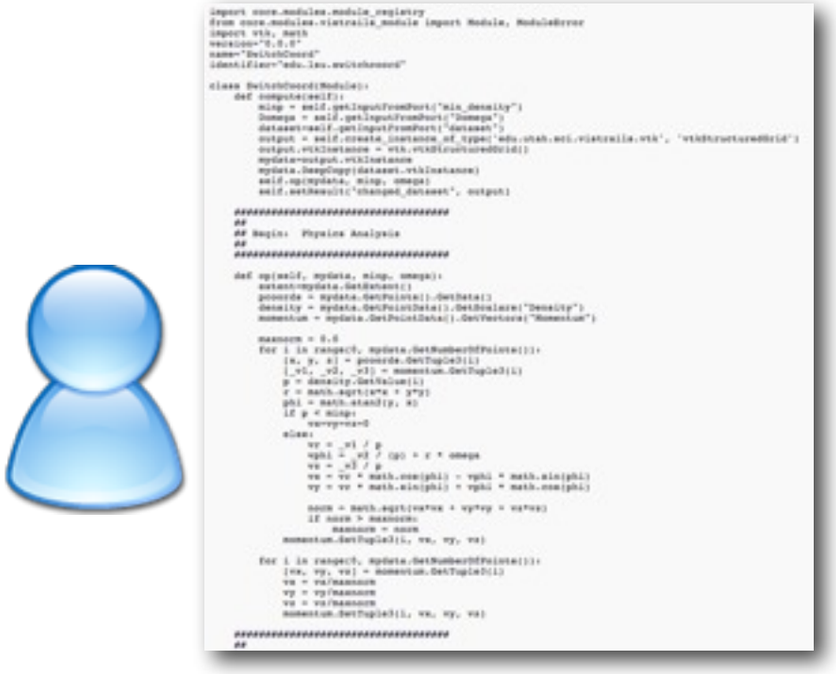
The routine of an Astrophysicist
Data Analysis and Visualization

```
import core.modules.module_registry
from core.modules.vistrails_module import Module, ModuleError
import vtk, math
version="0.0.0"
name="SwitchCoord"
identifier="edu.lsu.switchcoord"

class SwitchCoord(Module):
    def compute(self):
        ming = self.getInputFromPort("min_density")
        Domega = self.getInputFromPort("Domega")
        dataset=self.getInputFromPort("dataset")
        output = self.create_instance_of_type('edu.utah.sci.vistrails.vtk', 'vtkStructuredGrid')
        output.vtkInstance = vtk.vtkStructuredGrid()
        mydata=output.vtkInstance
        mydata.DeepCopy(dataset.vtkInstance)
        self.op(mydata, ming, omega)
        self.setResult('changed_dataset', output)

    ###################################
    ##
    ## Begin:  Physics Analysis
    ##
    ###################################

    def op(self, mydata, ming, omega):
        extent=mydata.GetExtent()
        pcoords = mydata.GetPoints().GetData()
        density = mydata.GetPointData().GetScalars("Density")
        momentum = mydata.GetPointData().GetVectors("Momentum")

        maxnorm = 0.0
        for i in range(0, mydata.GetNumberOfPoints()):
            [x, y, z] = pcoords.GetTuple3(i)
            [_vi, _vj, _vk] = momentum.GetTuple3(i)
            p = density.GetValue(i)
            r = math.sqrt(x*x + y*y)
            phi = math.atan2(y, x)
            if p < ming:
                vx=vy=vz=0
            else:
                vr = _vi / p
                vphi = _vj / (p) + r * omega
                vz = _vk / p
                vx = vr * math.cos(phi) - vphi * math.sin(phi)
                vy = vr * math.sin(phi) + vphi * math.cos(phi)

                norm = math.sqrt(vx*vx + vy*vy + vz*vz)
                if norm > maxnorm:
                    maxnorm = norm
            momentum.SetTuple3(i, vx, vy, vz)

        for i in range(0, mydata.GetNumberOfPoints()):
            [vx, vy, vz] = momentum.GetTuple3(i)
            vx = vx/maxnorm
            vy = vy/maxnorm
            vz = vz/maxnorm
            momentum.SetTuple3(i, vx, vy, vz)

    ###################################
    ##
```

Analysis code

Explore results of simulations

The routine of an Astrophysicist
Data Analysis and Visualization



Analysis code

Integrate analysis
with Vis tools

Explore results of simulations

The routine of an Astrophysicist
Data Analysis and Visualization

Analysis code

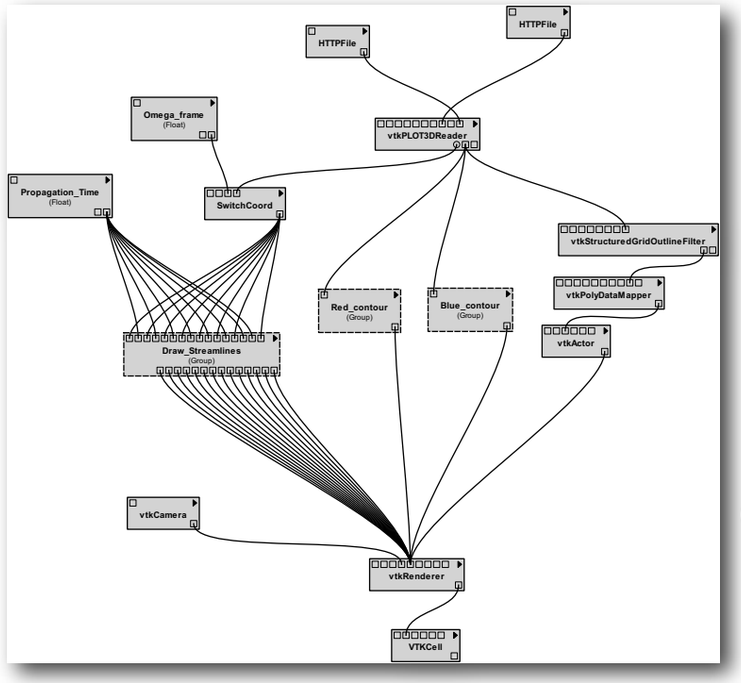
expert

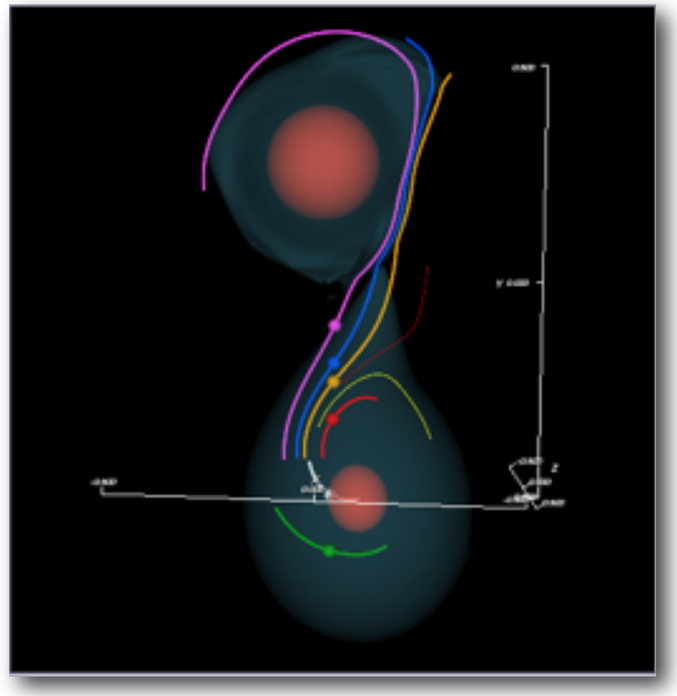
designer

Integrate analysis
with Vis tools

Explore results of simulations

The routine of an Astrophysicist
Data Analysis and Visualization





expert

designer

Analysis code

Integrate analysis with Vis tools

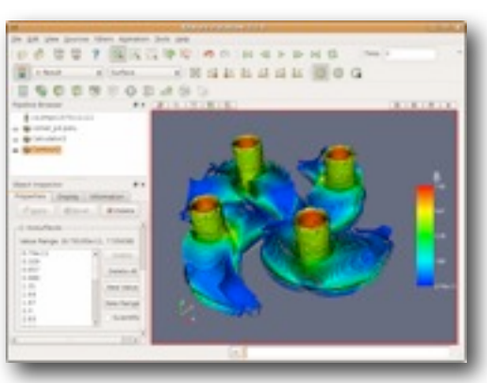
Explore results of simulations

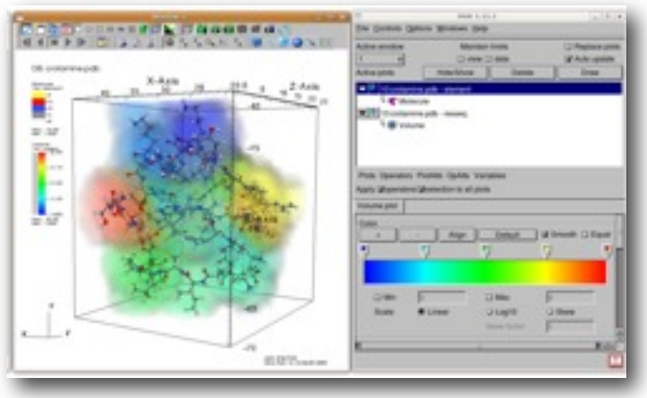
Complexity of Data Analysis and Visualization Tools

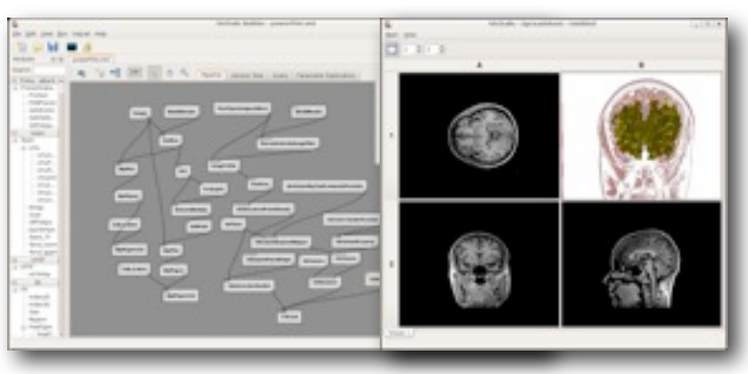
- In general, they are too complicated for scientists

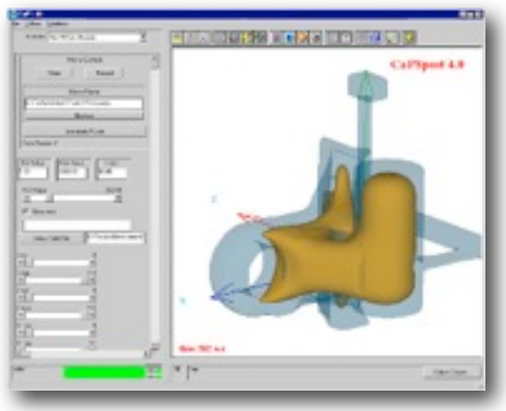
Complexity of
Data Analysis and Visualization Tools

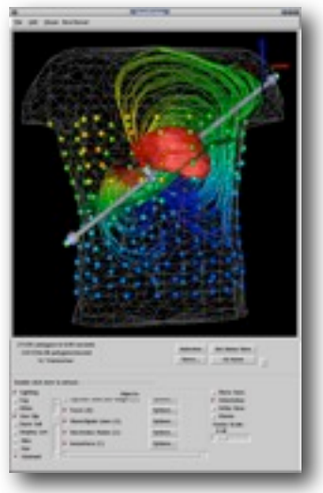
- In general, they are too complicated for scientists





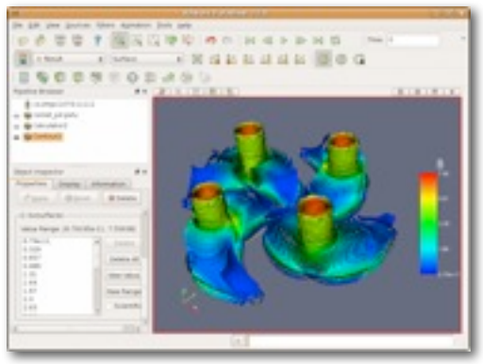


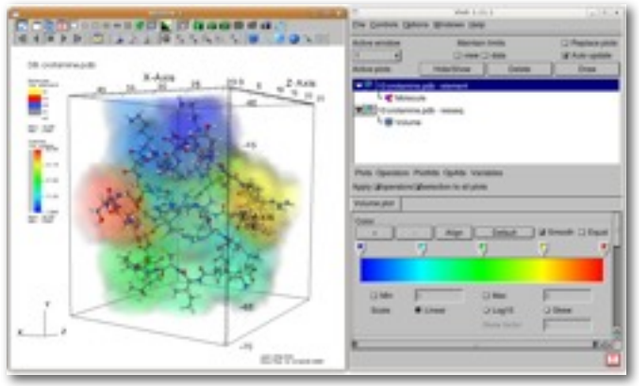


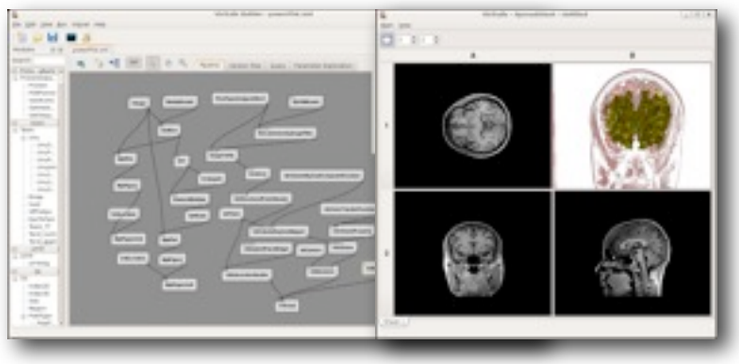


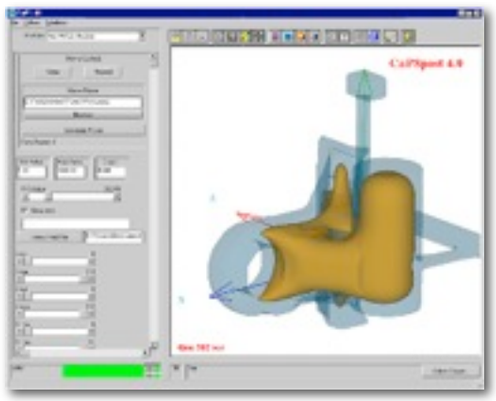
Complexity of
Data Analysis and Visualization Tools

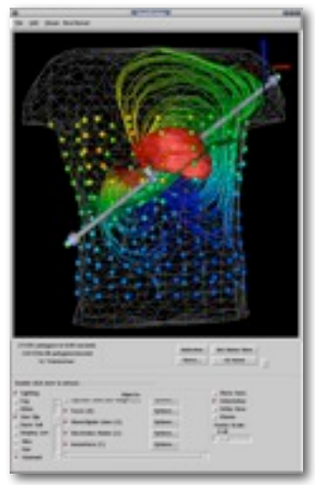
- In general, they are too complicated for scientists





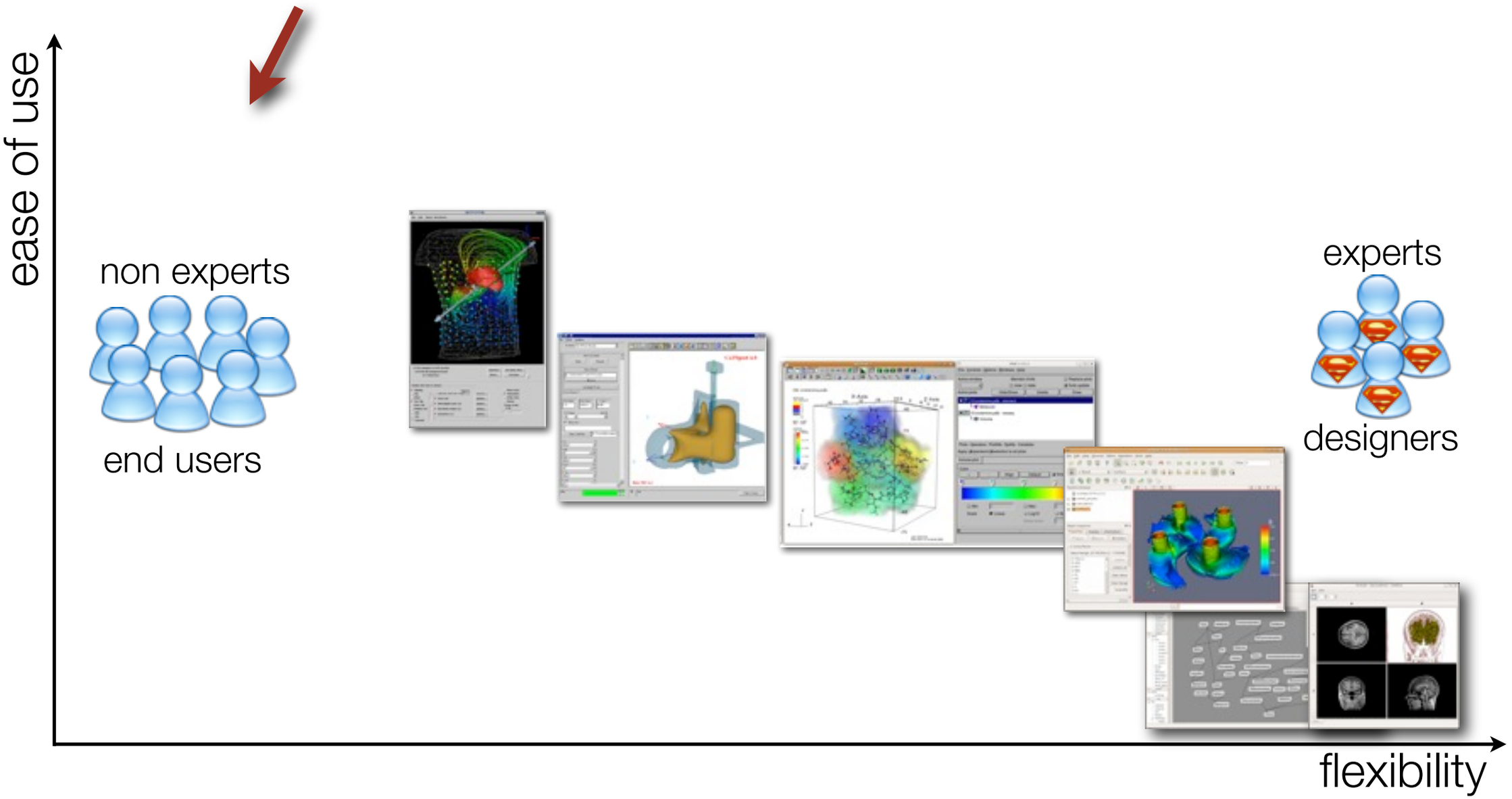




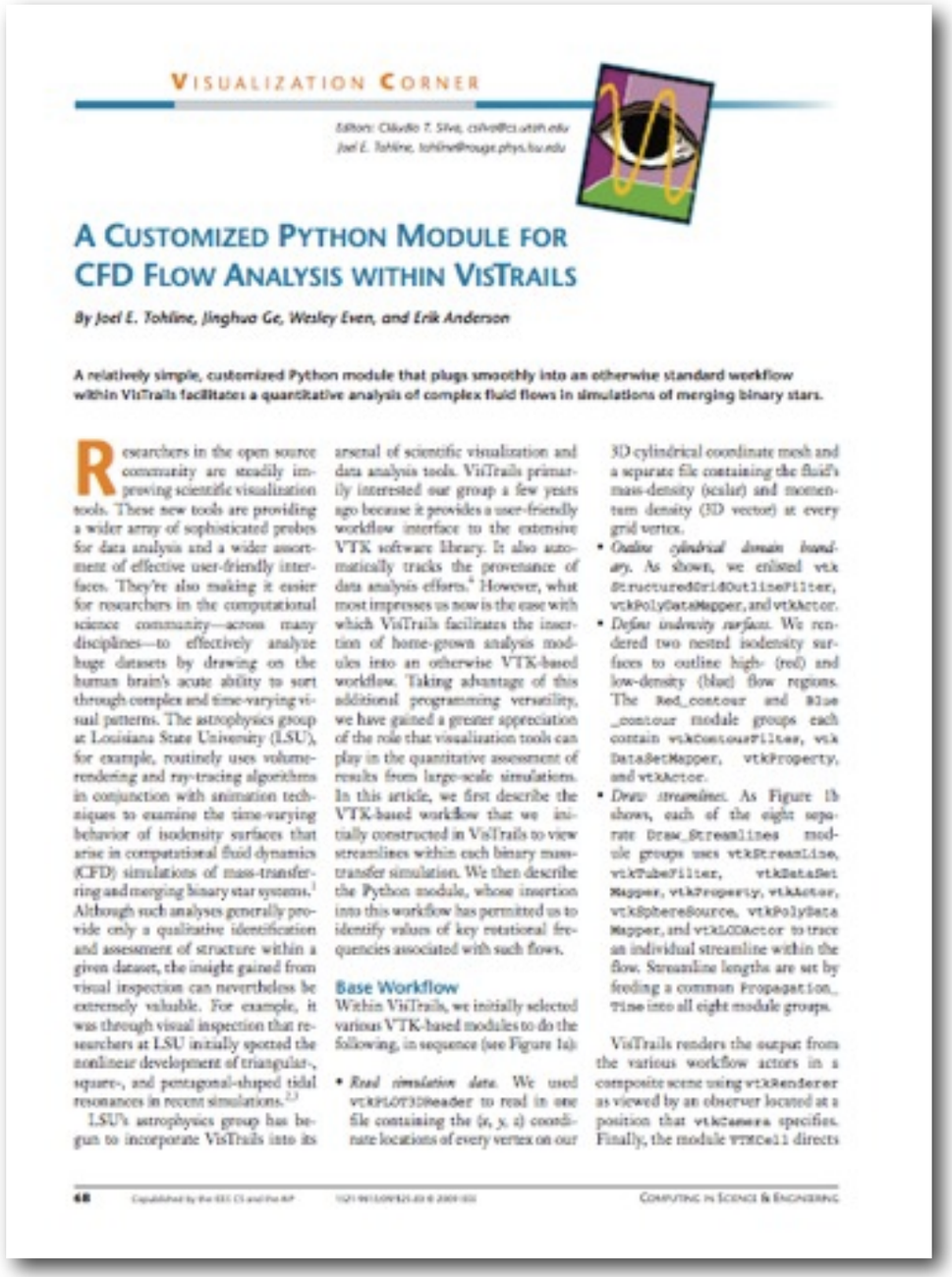


Complexity of
Data Analysis and Visualization Tools

- In general, they are too complicated for scientists



The routine of an Astrophysicist Publishing Results



VISUALIZATION CORNER

Editors: Cláudio T. Silva, csilva@cs.utah.edu
Joel E. Tohline, tohline@rouge.phys.lsu.edu

A CUSTOMIZED PYTHON MODULE FOR CFD FLOW ANALYSIS WITHIN VISTRAILS

By Joel E. Tohline, Jinghua Ge, Wesley Even, and Erik Anderson

A relatively simple, customized Python module that plugs smoothly into an otherwise standard workflow within VisTrails facilitates a quantitative analysis of complex fluid flows in simulations of merging binary stars.

Researchers in the open source community are steadily improving scientific visualization tools. These new tools are providing a wider array of sophisticated probes for data analysis and a wider assortment of effective user-friendly interfaces. They're also making it easier for researchers in the computational science community—across many disciplines—to effectively analyze huge datasets by drawing on the human brain's acute ability to sort through complex and time-varying visual patterns. The astrophysics group at Louisiana State University (LSU), for example, routinely uses volume-rendering and ray-tracing algorithms in conjunction with animation techniques to examine the time-varying behavior of isodensity surfaces that arise in computational fluid dynamics (CFD) simulations of mass-transferring and merging binary star systems.[1] Although such analyses generally provide only a qualitative identification and assessment of structure within a given dataset, the insight gained from visual inspection can nevertheless be extremely valuable. For example, it was through visual inspection that researchers at LSU initially spotted the nonlinear development of triangular-, square-, and pentagonal-shaped tidal resonances in recent simulations.[2,3]

LSU's astrophysics group has begun to incorporate VisTrails into its arsenal of scientific visualization and data analysis tools. VisTrails primarily interested our group a few years ago because it provides a user-friendly workflow interface to the extensive VTK software library. It also automatically tracks the provenance of data analysis efforts.[4] However, what most impresses us now is the ease with which VisTrails facilitates the insertion of home-grown analysis modules into an otherwise VTK-based workflow. Taking advantage of this additional programming versatility, we have gained a greater appreciation of the role that visualization tools can play in the quantitative assessment of results from large-scale simulations. In this article, we first describe the VTK-based workflow that we initially constructed in VisTrails to view streamlines within each binary mass-transfer simulation. We then describe the Python module, whose insertion into this workflow has permitted us to identify values of key rotational frequencies associated with such flows.

Base Workflow

Within VisTrails, we initially selected various VTK-based modules to do the following, in sequence (see Figure 1a):

- *Read simulation data.* We used `vtkPLOT3DReader` to read in one file containing the (x, y, z) coordinate locations of every vertex on our 3D cylindrical coordinate mesh and a separate file containing the fluid's mass-density (scalar) and momentum density (3D vector) at every grid vertex.
- *Outline cylindrical domain boundary.* As shown, we enlisted `vtkStructuredGridOutlineFilter`, `vtkPolyDataMapper`, and `vtkActor`.
- *Define isodensity surfaces.* We rendered two nested isodensity surfaces to outline high- (red) and low-density (blue) flow regions. The `Red_contour` and `Blue_contour` module groups each contain `vtkContourFilter`, `vtkDataSetMapper`, `vtkProperty`, and `vtkActor`.
- *Draw streamlines.* As Figure 1b shows, each of the eight separate `Draw_Streamlines` module groups uses `vtkStreamLine`, `vtkTubeFilter`, `vtkDataSetMapper`, `vtkProperty`, `vtkActor`, `vtkSphereSource`, `vtkPolyDataMapper`, and `vtkLODActor` to trace an individual streamline within the flow. Streamline lengths are set by feeding a common `Propagation_Time` into all eight module groups.

VisTrails renders the output from the various workflow actors in a composite scene using `vtkRenderer` as viewed by an observer located at a position that `vtkCamera` specifies. Finally, the module `VTKCell` directs

Tohline et al., CiSE 11(3), 2009

Printed document

The routine of an Astrophysicist Publishing Results

VISUALIZATION CORNER

Editors: Cláudio T. Silva, csilva@cs.utah.edu
Joel E. Tohline, tohline@rouge.phys.lsu.edu

A CUSTOMIZED PYTHON MODULE FOR CFD FLOW ANALYSIS WITHIN VISTRAILS

By Joel E. Tohline, Jinghua Ge, Wesley Even, and Erik Anderson

A relatively simple, customized Python module that plugs smoothly into an otherwise standard workflow within VisTrails facilitates a quantitative analysis of complex fluid flows in simulations of merging binary stars.

Researchers in the open source community are steadily improving scientific visualization tools. These new tools are providing a wider array of sophisticated probes for data analysis and a wider assortment of effective user-friendly interfaces. They're also making it easier for researchers in the computational science community—across many disciplines—to effectively analyze huge datasets by drawing on the human brain's acute ability to sort through complex and time-varying visual patterns. The astrophysics group at Louisiana State University (LSU), for example, routinely uses volume-rendering and ray-tracing algorithms in conjunction with animation techniques to examine the time-varying behavior of isodensity surfaces that arise in computational fluid dynamics (CFD) simulations of mass-transferring and merging binary star systems.[1] Although such analyses generally provide only a qualitative identification and assessment of structure within a given dataset, the insight gained from visual inspection can nevertheless be extremely valuable. For example, it was through visual inspection that researchers at LSU initially spotted the nonlinear development of triangular-, square-, and pentagonal-shaped tidal resonances in recent simulations.[2,3]

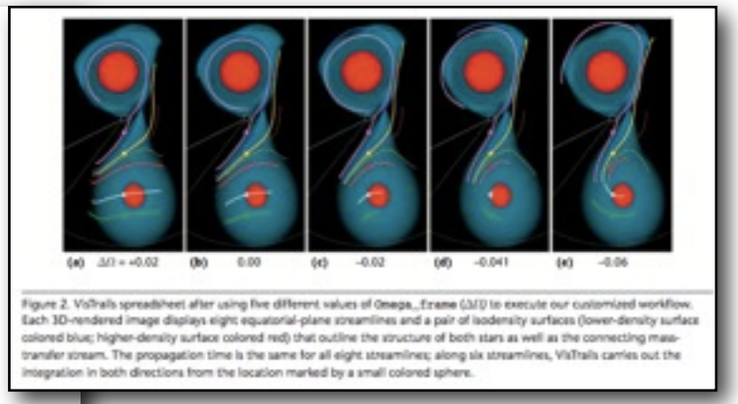
LSU's astrophysics group has begun to incorporate VisTrails into its arsenal of scientific visualization and data analysis tools. VisTrails primarily interested our group a few years ago because it provides a user-friendly workflow interface to the extensive VTK software library. It also automatically tracks the provenance of data analysis efforts.[4] However, what most impresses us now is the ease with which VisTrails facilitates the insertion of home-grown analysis modules into an otherwise VTK-based workflow. Taking advantage of this additional programming versatility, we have gained a greater appreciation of the role that visualization tools can play in the quantitative assessment of results from large-scale simulations. In this article, we first describe the VTK-based workflow that we initially constructed in VisTrails to view streamlines within each binary mass-transfer simulation. We then describe the Python module, whose insertion into this workflow has permitted us to identify values of key rotational frequencies associated with such flows.

Base Workflow

Within VisTrails, we initially selected various VTK-based modules to do the following, in sequence (see Figure 1a):

- *Read simulation data.* We used `vtkPLOT3DReader` to read in one file containing the (x, y, z) coordinate locations of every vertex on our 3D cylindrical coordinate mesh and a separate file containing the fluid's mass-density (scalar) and momentum density (3D vector) at every grid vertex.
- *Outline cylindrical domain boundary.* As shown, we enlisted `vtkStructuredGridOutlineFilter`, `vtkPolyDataMapper`, and `vtkActor`.
- *Define isodensity surfaces.* We rendered two nested isodensity surfaces to outline high- (red) and low-density (blue) flow regions. The `Red_contour` and `Blue_contour` module groups each contain `vtkContourFilter`, `vtkDataSetMapper`, `vtkProperty`, and `vtkActor`.
- *Draw streamlines.* As Figure 1b shows, each of the eight separate `Draw_Streamlines` module groups uses `vtkStreamLine`, `vtkTubeFilter`, `vtkDataSetMapper`, `vtkProperty`, `vtkActor`, `vtkSphereSource`, `vtkPolyDataMapper`, and `vtkLODActor` to trace an individual streamline within the flow. Streamline lengths are set by feeding a common `Propagation_Time` into all eight module groups.

VisTrails renders the output from the various workflow actors in a composite scene using `vtkRenderer` as viewed by an observer located at a position that `vtkCamera` specifies. Finally, the module `VTKCell` directs



(a) $\Delta\Omega = +0.02$ (b) 0.00 (c) −0.02 (d) −0.041 (e) −0.06

Figure 2. VisTrails spreadsheet after using five different values of `Omega_frame` ($\Delta\Omega$) to execute our customized workflow. Each 3D-rendered image displays eight equatorial-plane streamlines and a pair of isodensity surfaces (lower-density surface colored blue; higher-density surface colored red) that outline the structure of both stars as well as the connecting mass-transfer stream. The propagation time is the same for all eight streamlines; along six streamlines, VisTrails carries out the integration in both directions from the location marked by a small colored sphere.

Tohline et al., CiSE 11(3), 2009

Printed document

The routine of an Astrophysicist Publishing Results

VISUALIZATION CORNER

Editors: Cláudio T. Silva, csilva@cs.utah.edu
Joel E. Tohline, tohline@rouge.phys.lsu.edu

A CUSTOMIZED PYTHON MODULE FOR CFD FLOW ANALYSIS WITHIN VISTRAILS

By Joel E. Tohline, Jinghua Ge, Wesley Even, and Erik Anderson

A relatively simple, customized Python module that plugs smoothly into an otherwise standard workflow within VisTrails facilitates a quantitative analysis of complex fluid flows in simulations of merging binary stars.

Researchers in the open source community are steadily improving scientific visualization tools. These new tools are providing a wider array of sophisticated probes for data analysis and a wider assortment of effective user-friendly interfaces. They're also making it easier for researchers in the computational science community—across many disciplines—to effectively analyze huge datasets by drawing on the human brain's acute ability to sort through complex and time-varying visual patterns. The astrophysics group at Louisiana State University (LSU), for example, routinely uses volume-rendering and ray-tracing algorithms in conjunction with animation techniques to examine the time-varying behavior of isodensity surfaces that arise in computational fluid dynamics (CFD) simulations of mass-transferring and merging binary star systems.[1] Although such analyses generally provide only a qualitative identification and assessment of structure within a given dataset, the insight gained from visual inspection can nevertheless be extremely valuable. For example, it was through visual inspection that researchers at LSU initially spotted the nonlinear development of triangular-, square-, and pentagonal-shaped tidal resonances in recent simulations.[2,3]

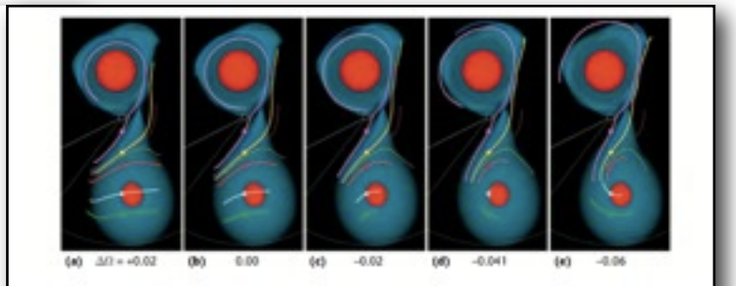
LSU's astrophysics group has begun to incorporate VisTrails into its arsenal of scientific visualization and data analysis tools. VisTrails primarily interested our group a few years ago because it provides a user-friendly workflow interface to the extensive VTK software library. It also automatically tracks the provenance of data analysis efforts.[4] However, what most impresses us now is the ease with which VisTrails facilitates the insertion of home-grown analysis modules into an otherwise VTK-based workflow. Taking advantage of this additional programming versatility, we have gained a greater appreciation of the role that visualization tools can play in the quantitative assessment of results from large-scale simulations. In this article, we first describe the VTK-based workflow that we initially constructed in VisTrails to view streamlines within each binary mass-transfer simulation. We then describe the Python module, whose insertion into this workflow has permitted us to identify values of key rotational frequencies associated with such flows.

Base Workflow

Within VisTrails, we initially selected various VTK-based modules to do the following, in sequence (see Figure 1a):

- *Read simulation data.* We used `vtkPLOT3DReader` to read in one file containing the (*x*, *y*, *z*) coordinate locations of every vertex on our 3D cylindrical coordinate mesh and a separate file containing the fluid's mass-density (scalar) and momentum density (3D vector) at every grid vertex.
- *Outline cylindrical domain boundary.* As shown, we enlisted `vtkStructuredGridOutlineFilter`, `vtkPolyDataMapper`, and `vtkActor`.
- *Define isodensity surfaces.* We rendered two nested isodensity surfaces to outline high- (red) and low-density (blue) flow regions. The `Red_contour` and `Blue_contour` module groups each contain `vtkContourFilter`, `vtkDataSetMapper`, `vtkProperty`, and `vtkActor`.
- *Draw streamlines.* As Figure 1b shows, each of the eight separate `Draw_Streamlines` module groups uses `vtkStreamLine`, `vtkTubeFilter`, `vtkDataSetMapper`, `vtkProperty`, `vtkActor`, `vtkSphereSource`, `vtkPolyDataMapper`, and `vtkLODActor` to trace an individual streamline within the flow. Streamline lengths are set by feeding a common `Propagation_Time` into all eight module groups.

VisTrails renders the output from the various workflow actors in a composite scene using `vtkRenderer` as viewed by an observer located at a position that `vtkCamera` specifies. Finally, the module `VTKCell` directs



(a) $\Delta\Omega = +0.02$ (b) 0.00 (c) −0.02 (d) −0.041 (e) −0.06

Figure 2. VisTrails spreadsheet after using five different values of Omega_frame ($\Delta\Omega$) to execute our customized workflow. Each 3D-rendered image displays eight equatorial-plane streamlines and a pair of isodensity surfaces (lower-density surface colored blue; h
transfer stream
integration in b

Tohline et al., CiSE 11(3), 2009

Printed document

SWITCHCOORD PYTHON MODULE

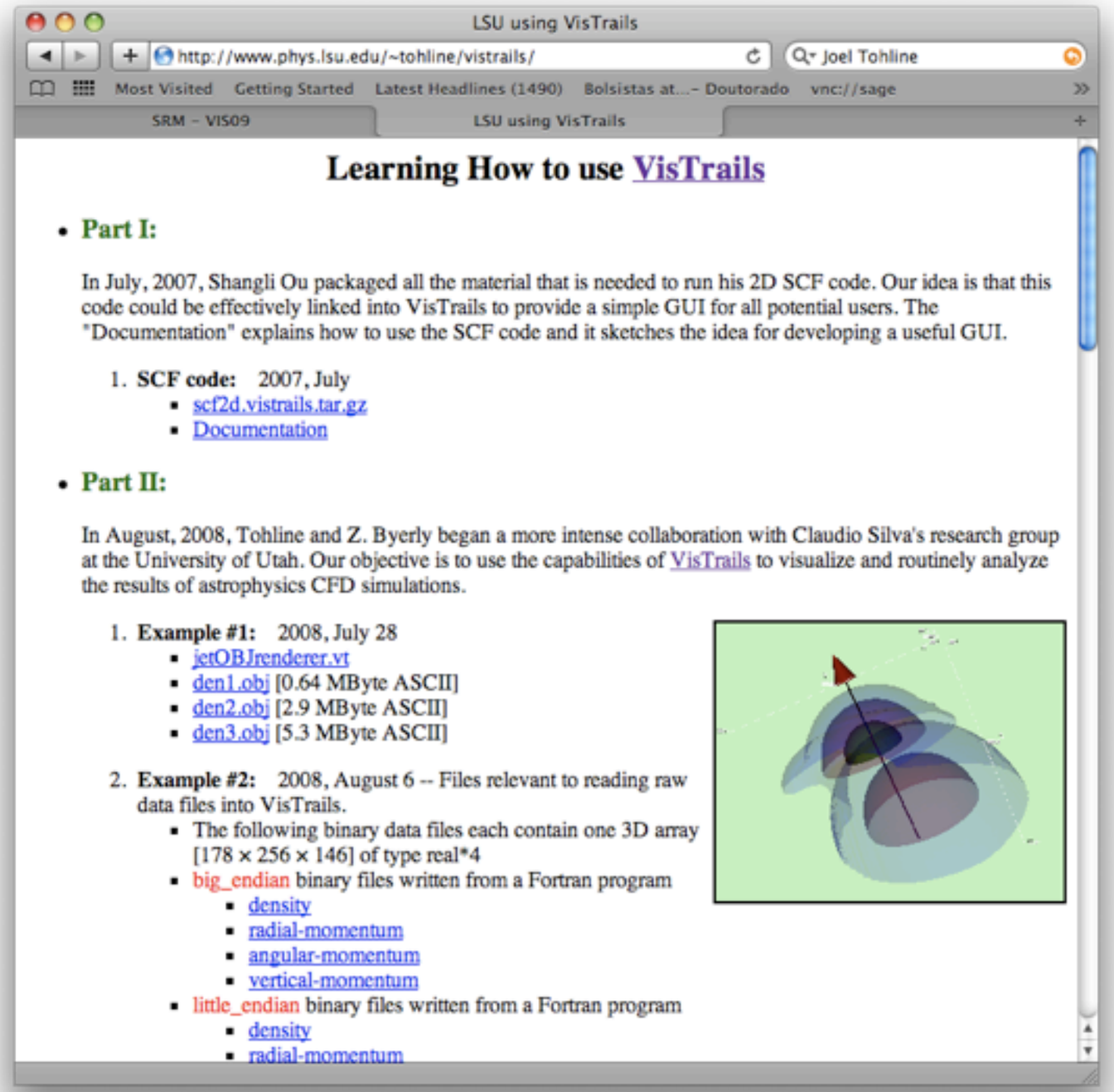
Here we present a complete listing of the Python code from our customized `SwitchCoord` program module. At the beginning of the "Physics Analysis" segment of the code, we assign names to the data arrays that have been acquired as input from `vtkPLOT3DReader`: *pcoords* is a tuple that identifies the Cartesian-based coordinate location of each grid vertex, *density* is a scalar that specifies the mass density, and *momentum* is a tuple that specifies the values of the cylindrical-coordinate-based vector momentum at each grid vertex. We detail the remaining logic of the physics analysis in the main text.

```
import core.modules.module_registry
from core.modules.vistrails_module import
Module, ModuleError
import vtk, math
version="0.0.0"
name="SwitchCoord"
identifier="edu.lsu.switchcoord"

class SwitchCoord(Module):
  def compute(self):
    minp = self.
getInputFromPort("min_density")
    Domega = self.getInputFromPort("Domega")
    dataset=self.getInputFromPort("dataset")
    output = self.create_instance_of_type(
      'edu.utah.sci.vistrails.vtk',
      'vtkStructuredGrid')
    output.vtkInstance = vtk.
vtkStructuredGrid()
    mydata=output.vtkInstance
    mydata.DeepCopy(dataset.vtkInstance)
    self.op(mydata, minp, Domega)
    self.setResult("changed_dataset",
output)

  ###################################
  ##
  ## Begin: Physics Analysis
  ##
  ###################################

  def op(self, mydata, minp, Domega):
    extent=mydata.GetExtent()
    pcoords = mydata.GetPoints().GetData()
    density = mydata.GetPointData().
GetScalars("Density")
    momentum = mydata.GetPointData().
GetVectors("Momentum")

    maxnorm = 0.0
    for i in range(0, mydata.
GetNumberOfPoints()):
      [x, y, z] = pcoords.GetTuple3(i)
      [_v1, _v2, _v3] = momentum.
GetTuple3(i)
      p = density.GetValue(i)
      r = math.sqrt(x*x + y*y)
```

Limitations of hardcopy documents

- Limited space
- Represent a snapshot of a given problem and solution
- Format doesn't allow for transmission of supporting files such as accompanying images, source code, or demonstrations of work to interested readers
  - "It's impossible to verify most of the results that computational scientists present at conference and in papers." [Donoho et al., 2009]
  - "Scientific and mathematical journals are filled with pretty pictures of computational experiments that the reader has no hope of repeating." [LeVeque, 2009]

The routine of an Astrophysicist Sharing Analysis and Collaborating



Web

- Users have to go over many (complicated) steps to reproduce figures:
  - install software
  - download extra libraries
  - download example files
  - learn how to use the software

Sharing Analysis

- Visualizations are hard to share and reuse
  - Powerpoint files, screenshots, videos
  - File dependencies
  - Tools that are general and complex
  - Often code is not multi-platform
- Visualization portals (such as Many Eyes)
  - Limited to a set of predefined visualizations
  - Do not support private data

Thesis Statement (Recap)

Effective exploration of scientific data requires a **combination of novel software and frameworks** that provide users with different **goals and levels of expertise** the ability to **create**, **modify** and **share** their **own analysis and visualization workflows and tools**

Outline

- Motivation
  - Scientific Research Scenario
  - Thesis Statement
- Framework
  - VisMashup
  - CrowdLabs
  - Provenance Rich Publications
- Demonstration
- Summary and Conclusions
- Future Work

Context and Requirements

- Provenance
- Multiple libraries
- Access to High-Performance Computing Resources
- Different communities want their customized GUI
- Test results using an ordinary Web browser
- Control who can access the information
- The tools for sharing and publishing results need to be integrated with the tools the scientists already use

Context and Requirements

✓Provenance

✓Multiple libraries

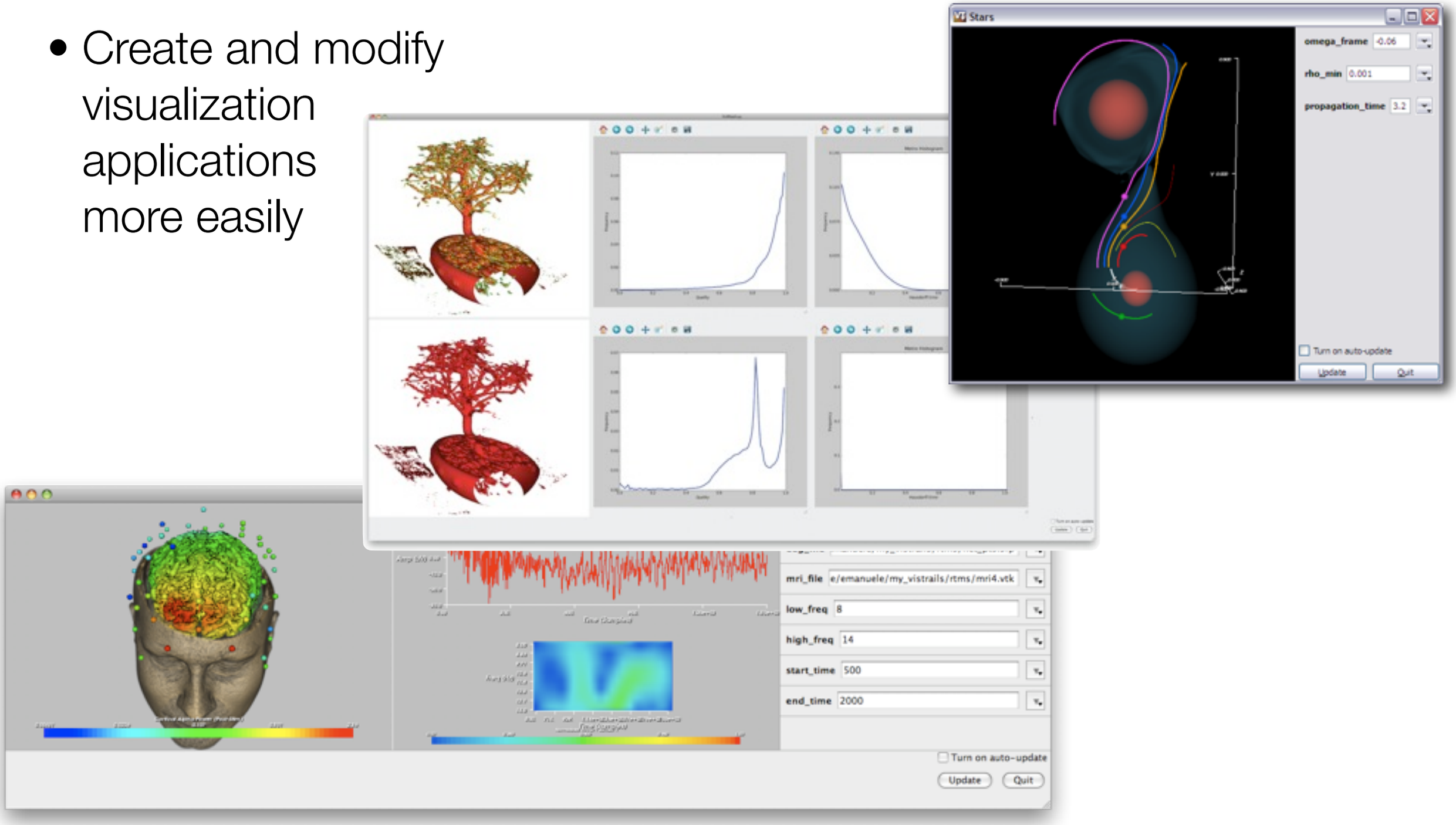
✓Access to High-Performance Computing Resources

VisTrails

- Different communities want their customized GUI
- Test results using an ordinary Web browser
- Control who can access the information
- The tools for sharing and publishing results need to be integrated with the tools the scientists already use

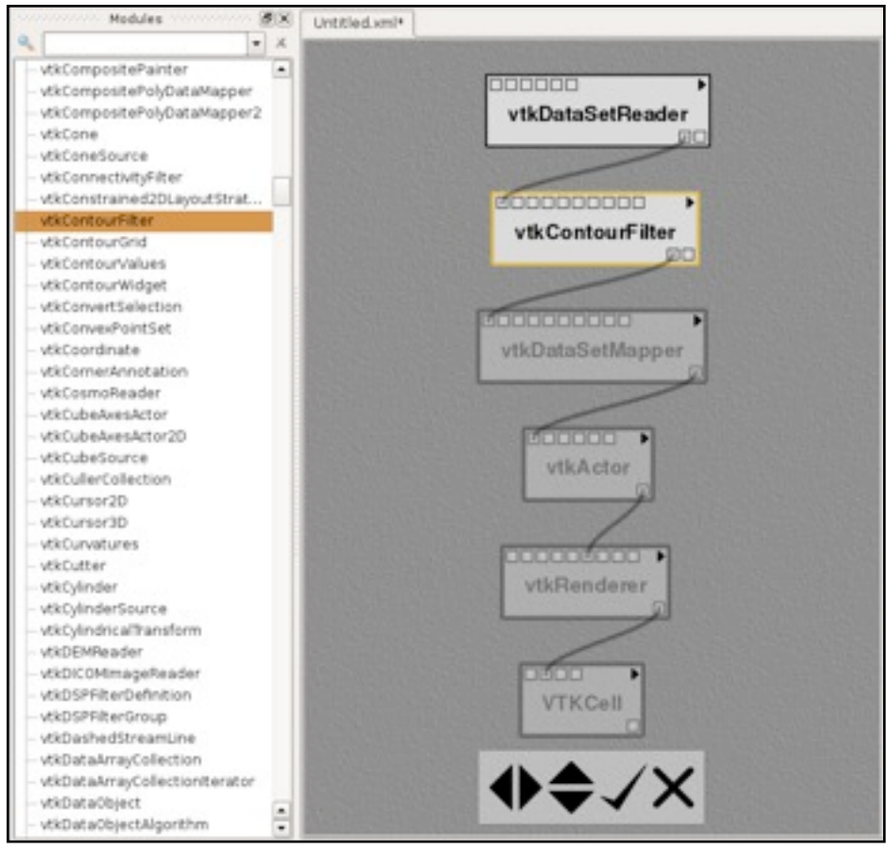
VisMashup

- Create and modify visualization applications more easily

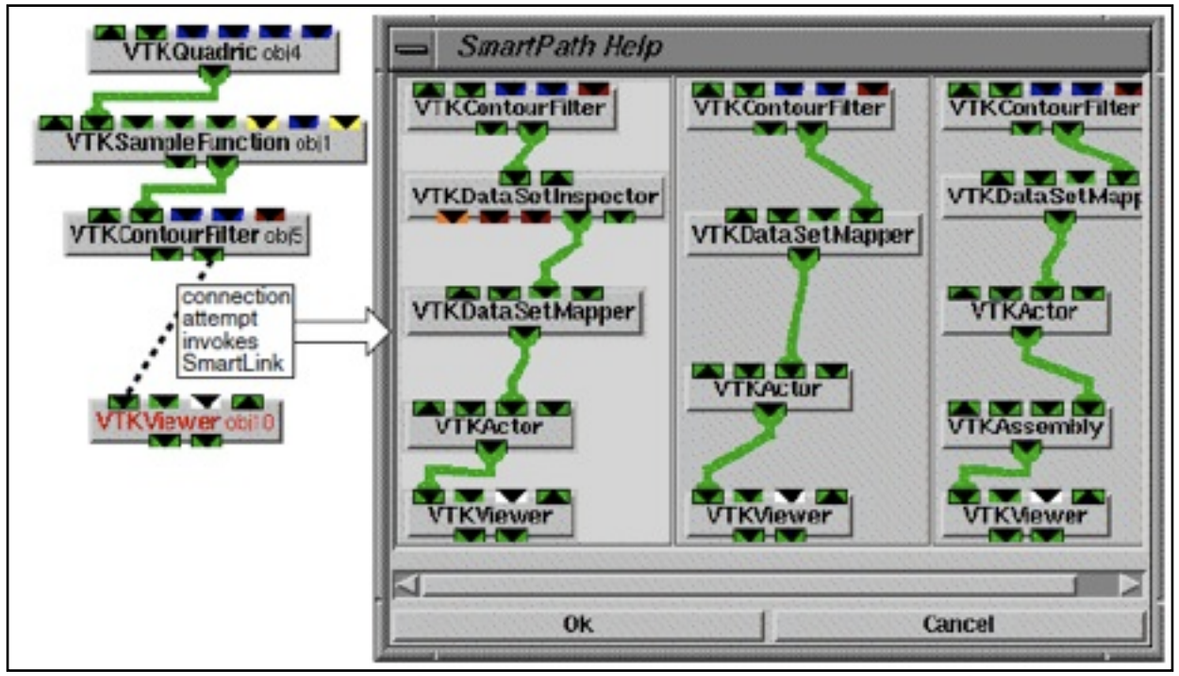


(joint work with Lauro Lins, Juliana Freire, James Ahrens and Cláudio Silva, Vis 2009)

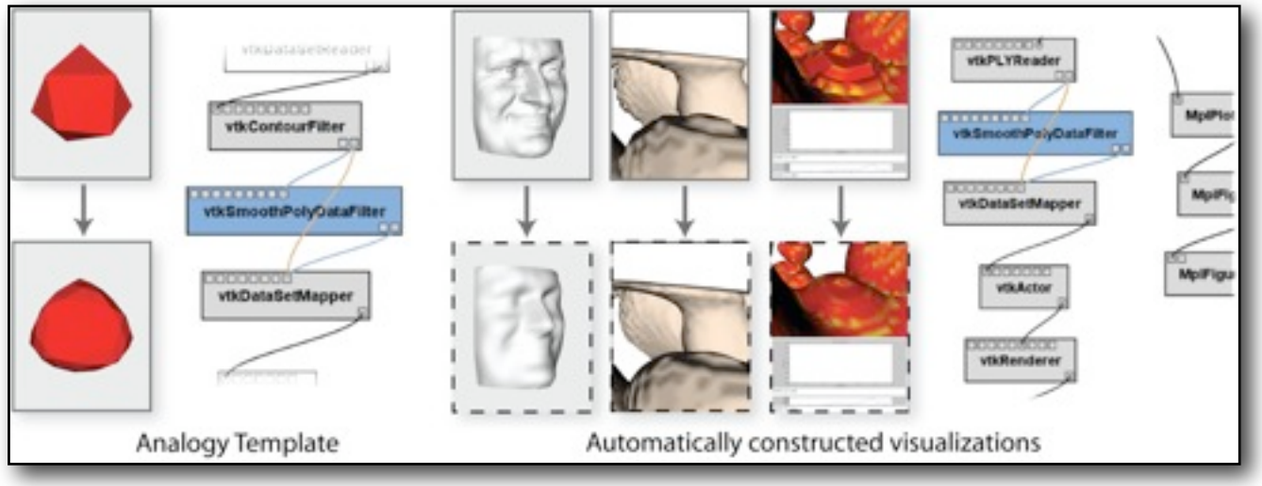
Other approaches
Helping to create visualizations



VisComplete: Koop et al., Vis 2008

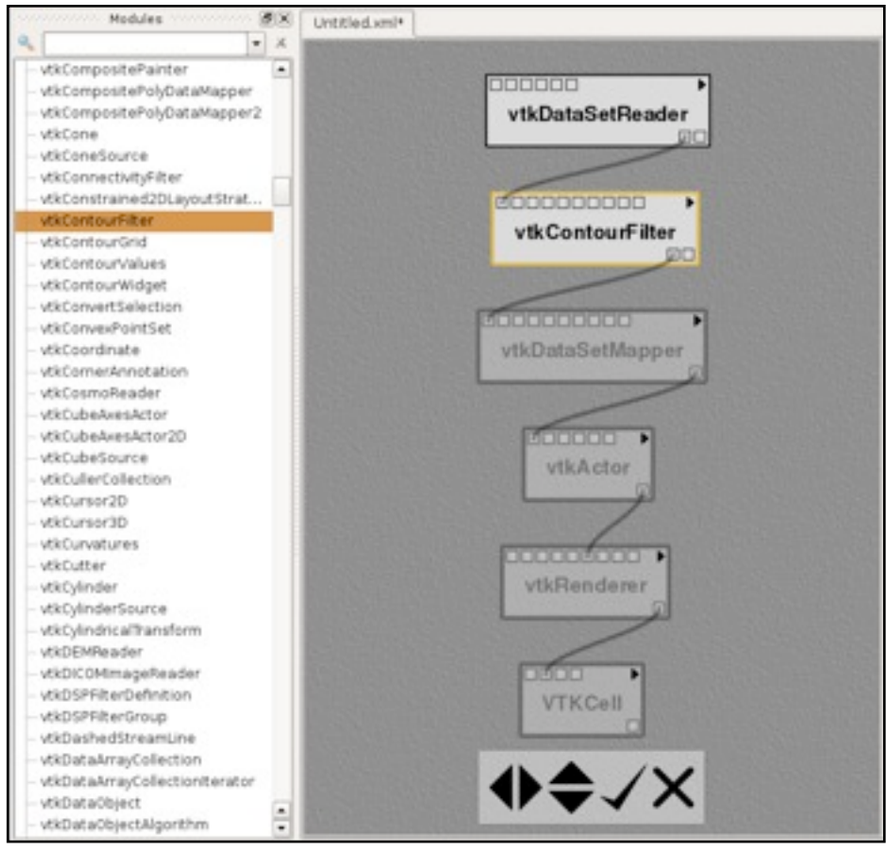


SmartLink: Telea & van Wijk, VisSym 1999

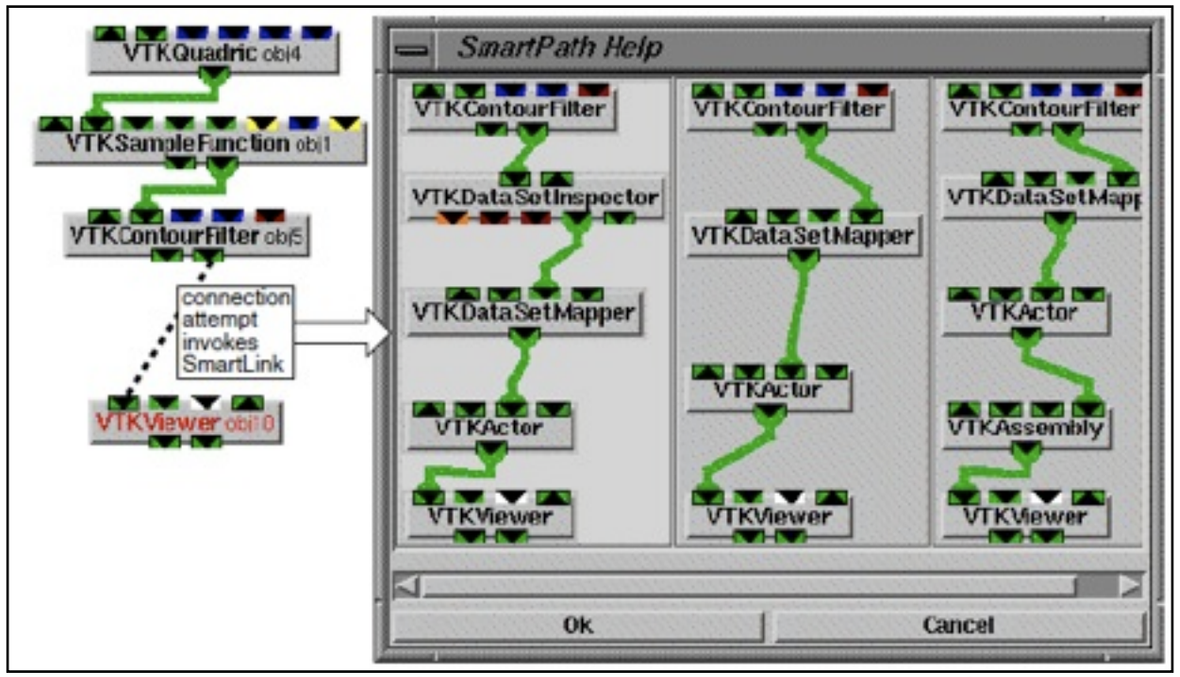


Vis by Analogy: Scheidegger et al., Vis 2007

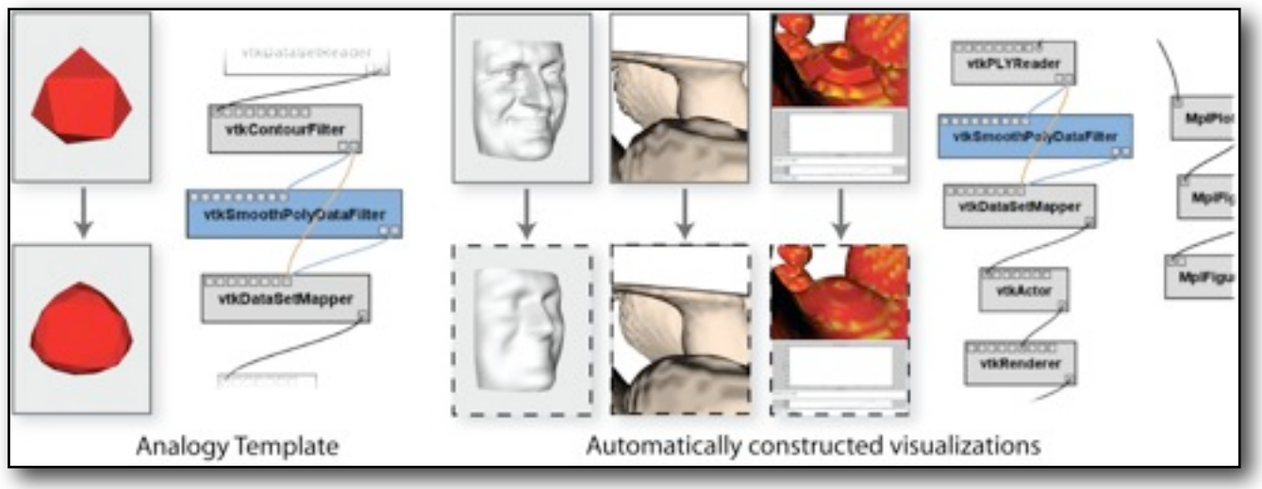
Other approaches
Helping to create visualizations



VisComplete: Koop et al., Vis 2008



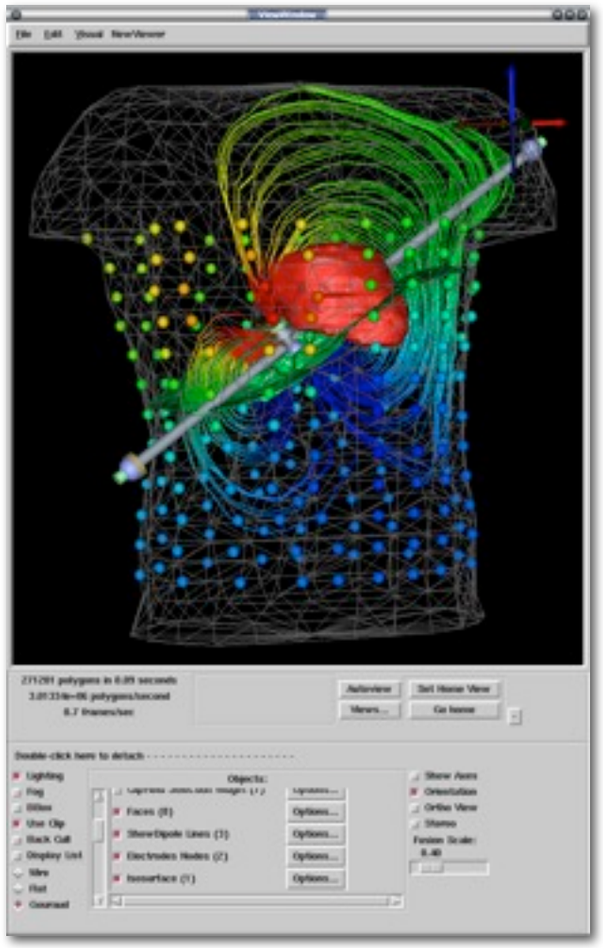
SmartLink: Telea & van Wijk, VisSym 1999



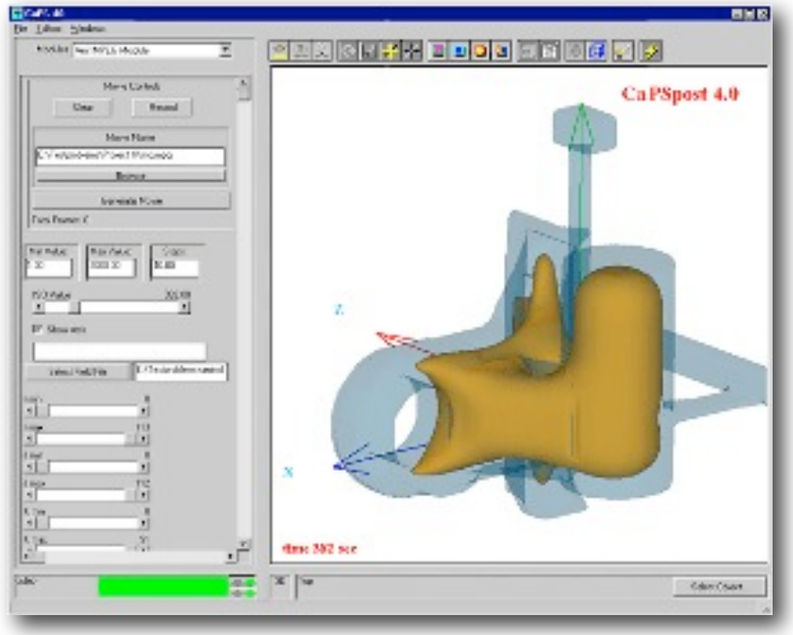
Vis by Analogy: Scheidegger et al., Vis 2007

Detailed knowledge of the underlying computational components is still necessary

Other approaches
Helping to create visualizations with custom apps



BioFEM: a SCIRun PowerApp



www.avs.com

Applications are costly to develop, manually crafted for a given pipeline

Other approaches
Reduced functionality interfaces

RESEARCH CONTRIBUTIONS

Human Aspects of Computing

Henry F. Ledgard Editor

Training Wheels in a User Interface

JOHN M. CARROLL and CAROLINE CARRITHERS

ABSTRACT: Naive users of high-function application systems can become frustrated and confused by the errors they make in the early stages of learning. A training interface for a commercial word processor was designed to make typical and troublesome error states "unreachable," thus eliminating the sources of some new-user learning problems. Creating a training environment from the basic function of the system itself afforded substantially faster learning coupled with better learning achievement and better performance on a comprehension post-test. A control group spent almost a quarter of their time recovering from the error states that the training interface blocked off. We speculate on how this training strategy might be refined, and more generally, on how function should be organized in a user interface.

Empirical characterizations of computer novices learning to use application systems paint a dreary picture of side tracks and error tangles from which recovery, or even diagnosis, is difficult (e.g., [6]). In this paper, we show that a "training wheels" interface—designed to block typical side tracks and error states—can facilitate the learning process for new users.

THE TRAINING WHEELS INTERFACE

We studied a stand-alone, commercial word processing system and observed new users, people who had never used a computer before, trying to learn the system's basic function. From this, we developed an inventory of typical new-user errors. The system's interface was then modified to block these error states, that is, to make them unreachable. This training wheels interface afforded the basic function for document creation, revision, and printing, but no advanced word processing function (e.g., table formatting). A variety of specific errors were identified and blocked. The most important (in terms of the severity of their consequences) were these seven (the errors are numbered and labeled for later reference):

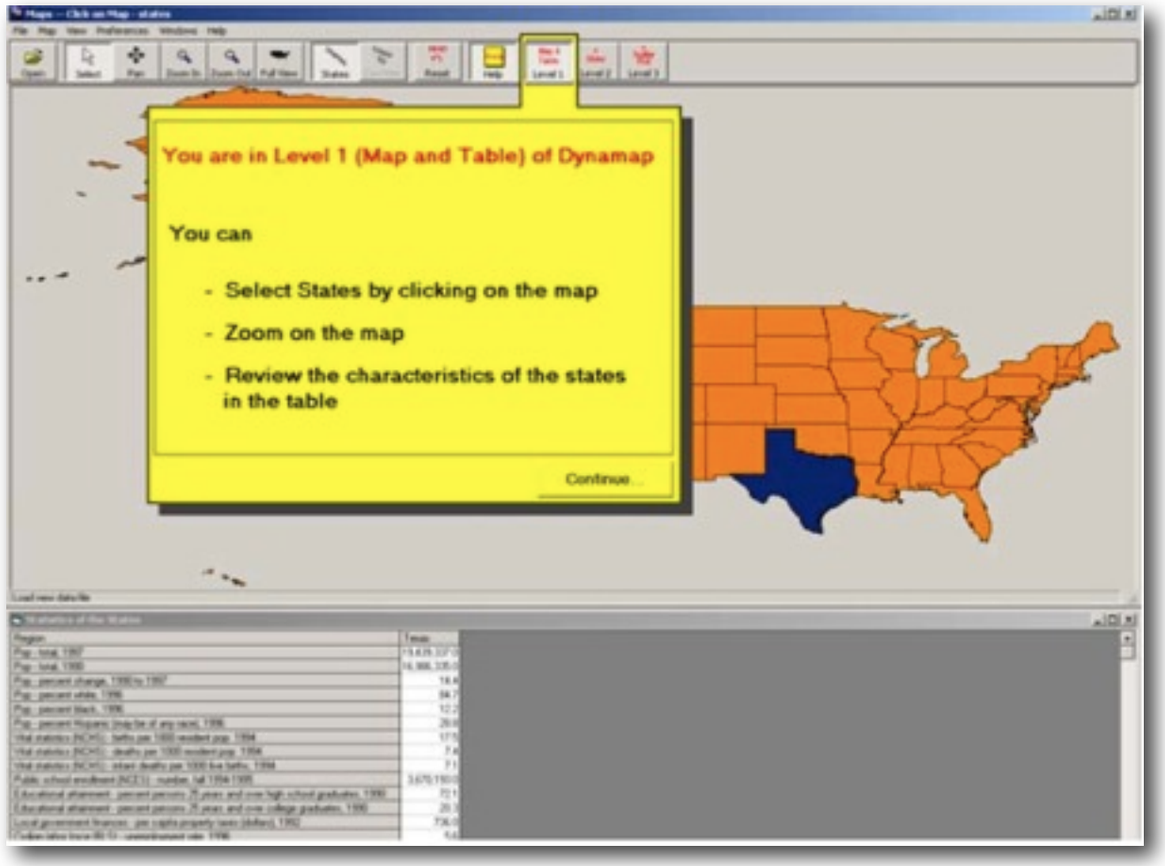
1. ***The Exotic Menu Choice Error.*** New users often recklessly tried out menu choices in their early encounters with the system. These users typically became tangled in advanced menus (for example, those pertaining to system and diskette maintenance) and were unable to get back to the simple function they had originally set out to learn. This error was eliminated by making only basic menu choices accessible in the training wheels interface. When inappropriate, exotic choices were selected, the message "`XXX not available on the Training System`" was displayed (where "`XXX`" stands for the name of the selected function), and the advanced function was not engaged.

2. ***The Print First Error.*** New users want to undertake real work immediately. We observed novices who requested a print job before they had even created and stored a document. This error was

This paper was presented at the 24th Annual Meeting of the Psychonomic Society, on November 17, 1983, in San Diego, California.

© 1984 ACM 0001-0782/84/0800-0800 75¢

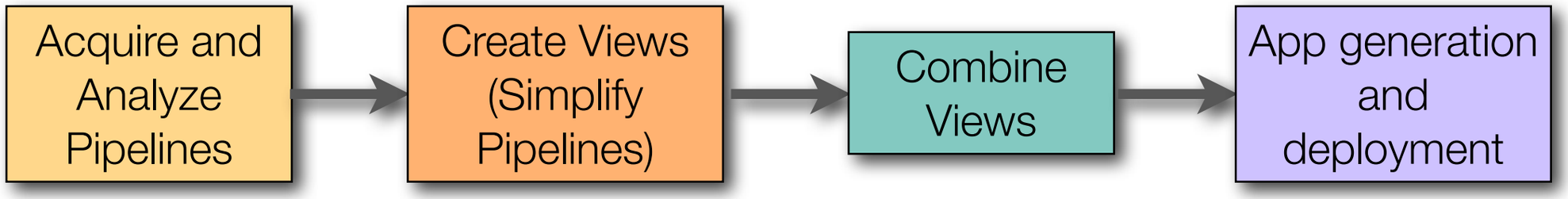
Training Wheels: Carroll & Carrithers, 1984



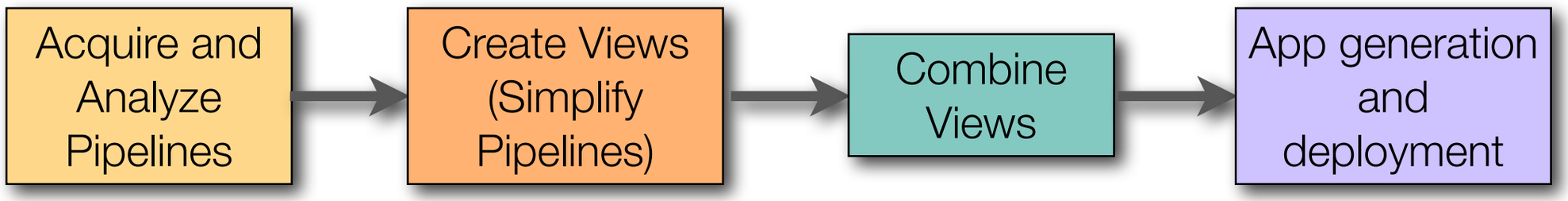
Multi-layerered interfaces: Shneiderman, 2003

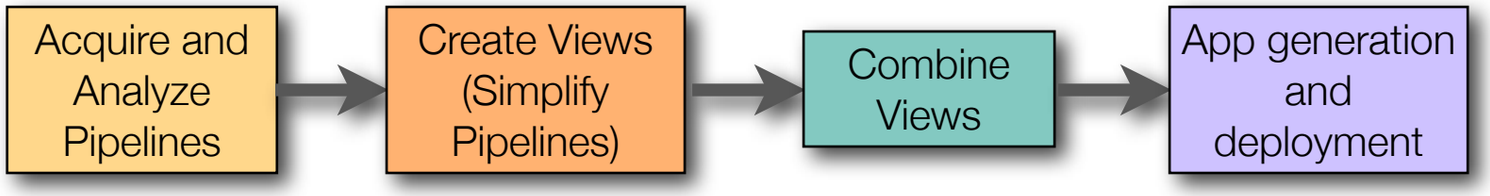
VisMashup

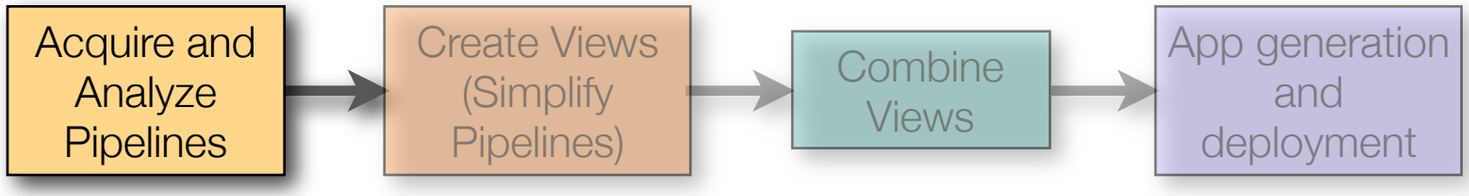
- Simplifies the creation, maintenance, deployment and use of customized visualization applications (mashups)
- Uses dataflows as the underlying model
- Keeps detailed provenance information of the application development process and use



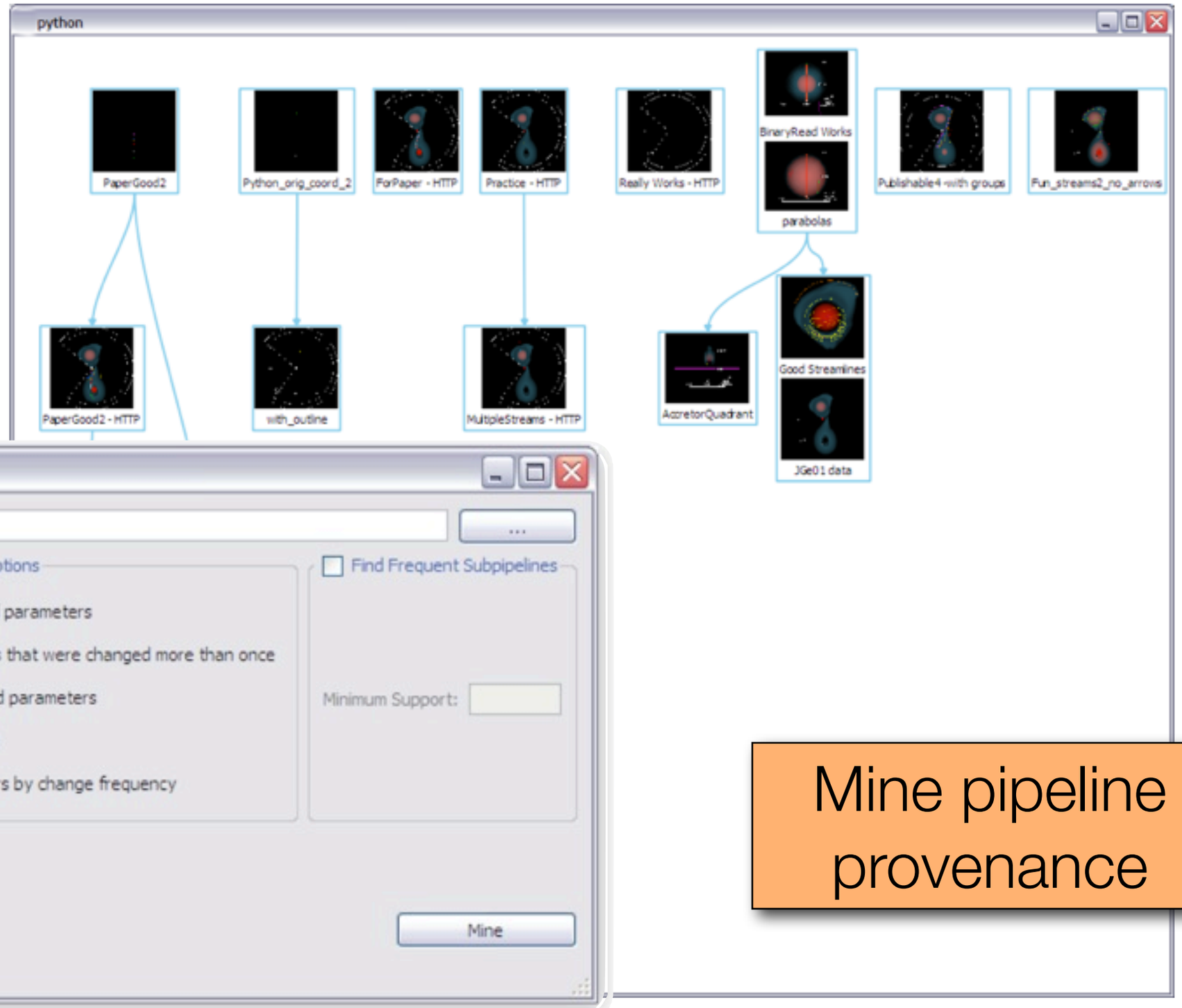
VisMashup Overview

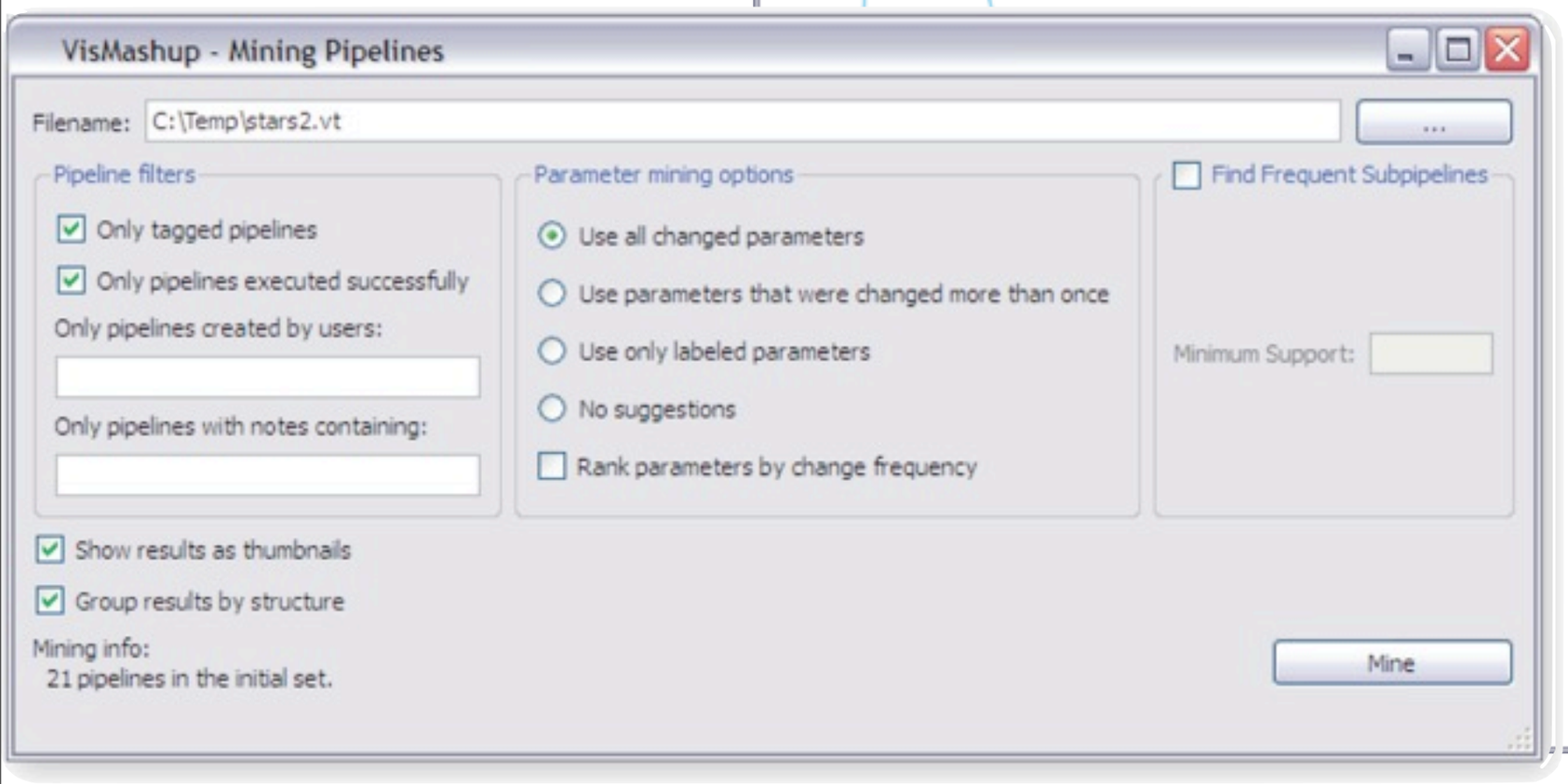



Acquire and Analyze Pipelines
Create Views (Simplify Pipelines)
Combine Views
App generation and deployment


Acquire and Analyze Pipelines
Create Views (Simplify Pipelines)
Combine Views
App generation and deployment

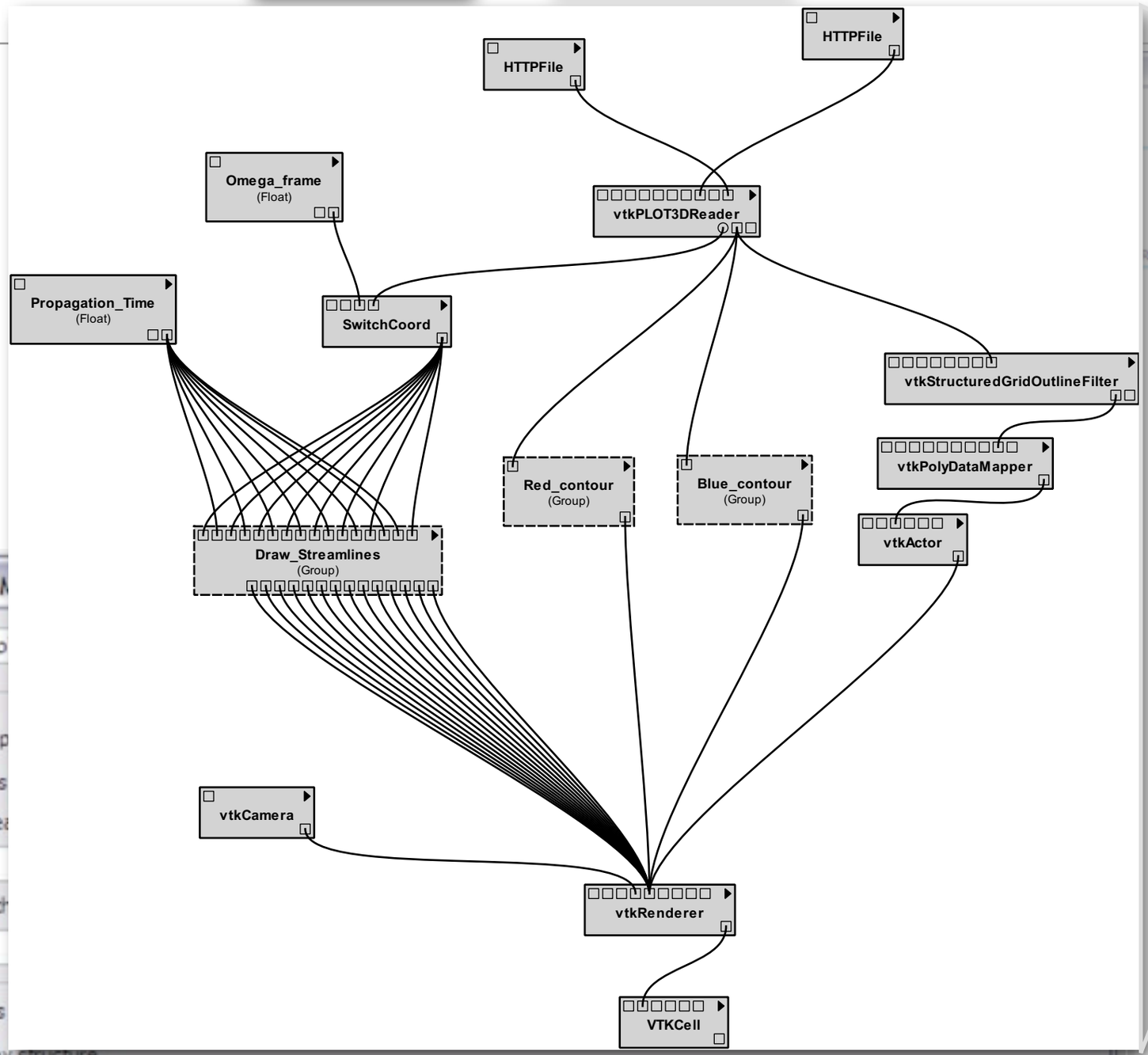
Acquire and Analyze Pipelines → Create Views (Simplify Pipelines) → Combine Views → App generation and deployment

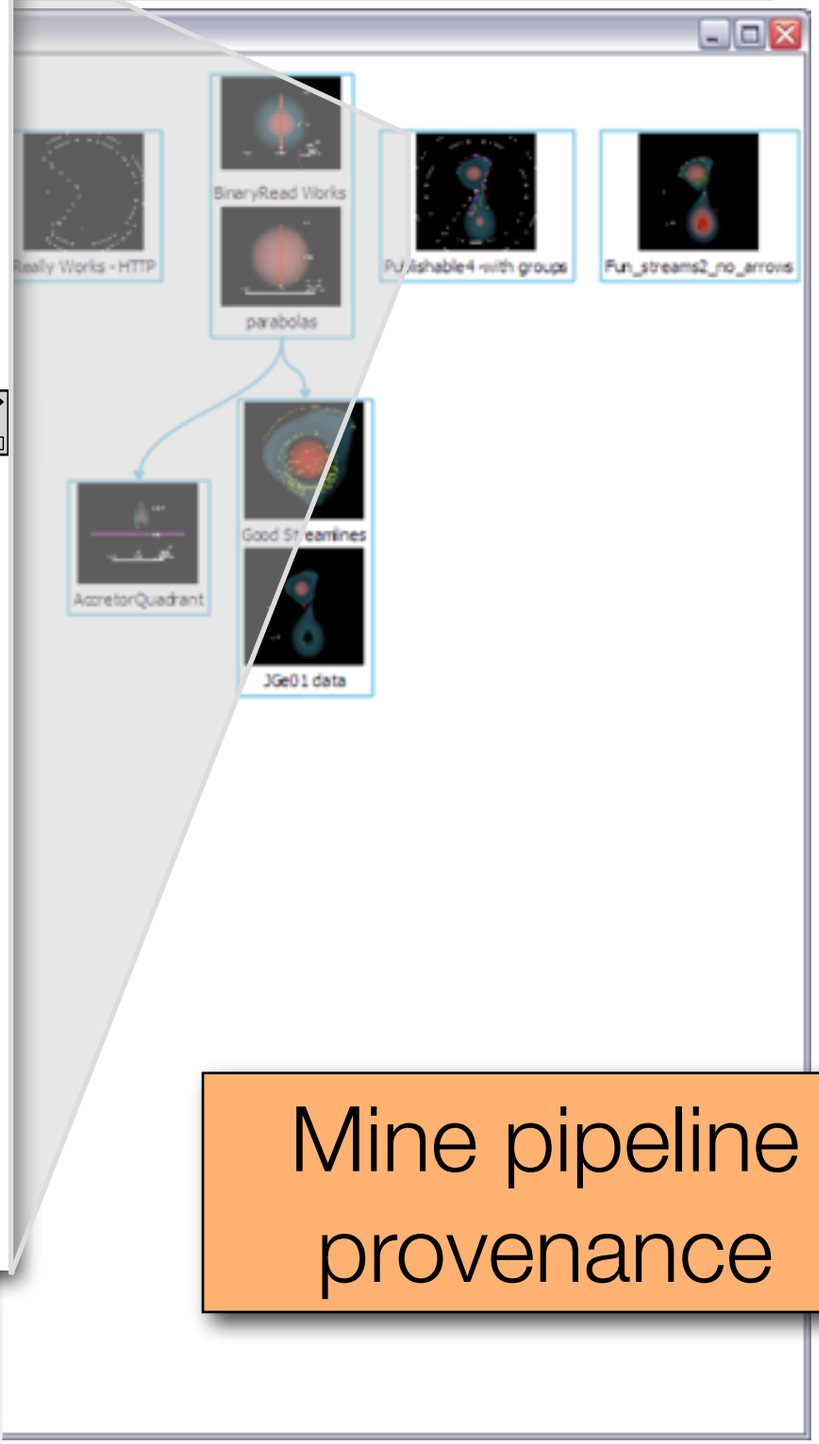


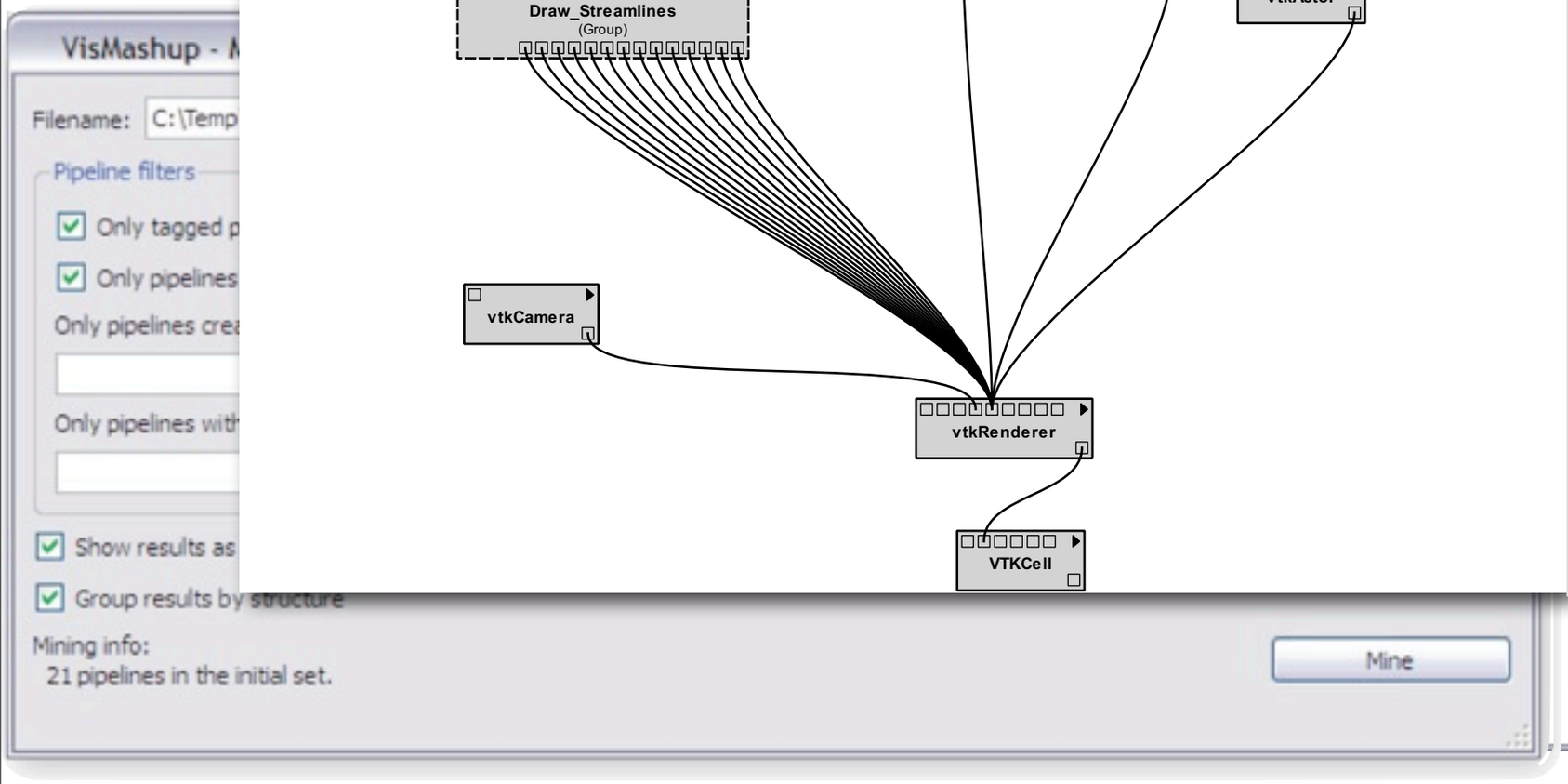


Mine pipeline provenance

Acquire and Analyze Pipelines → Create Views (Simplify Pipelines) → Combine Views → App generation and deployment



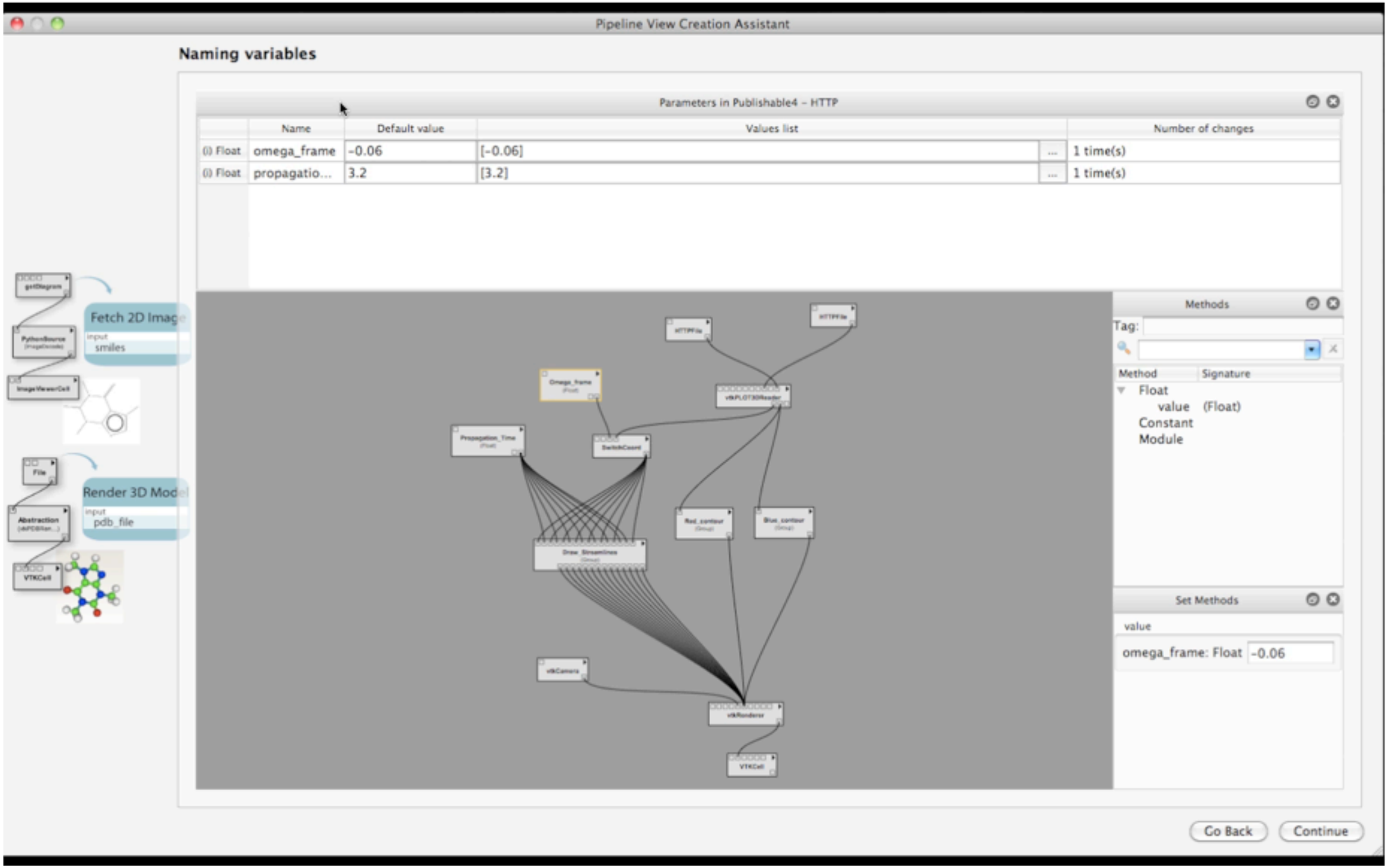




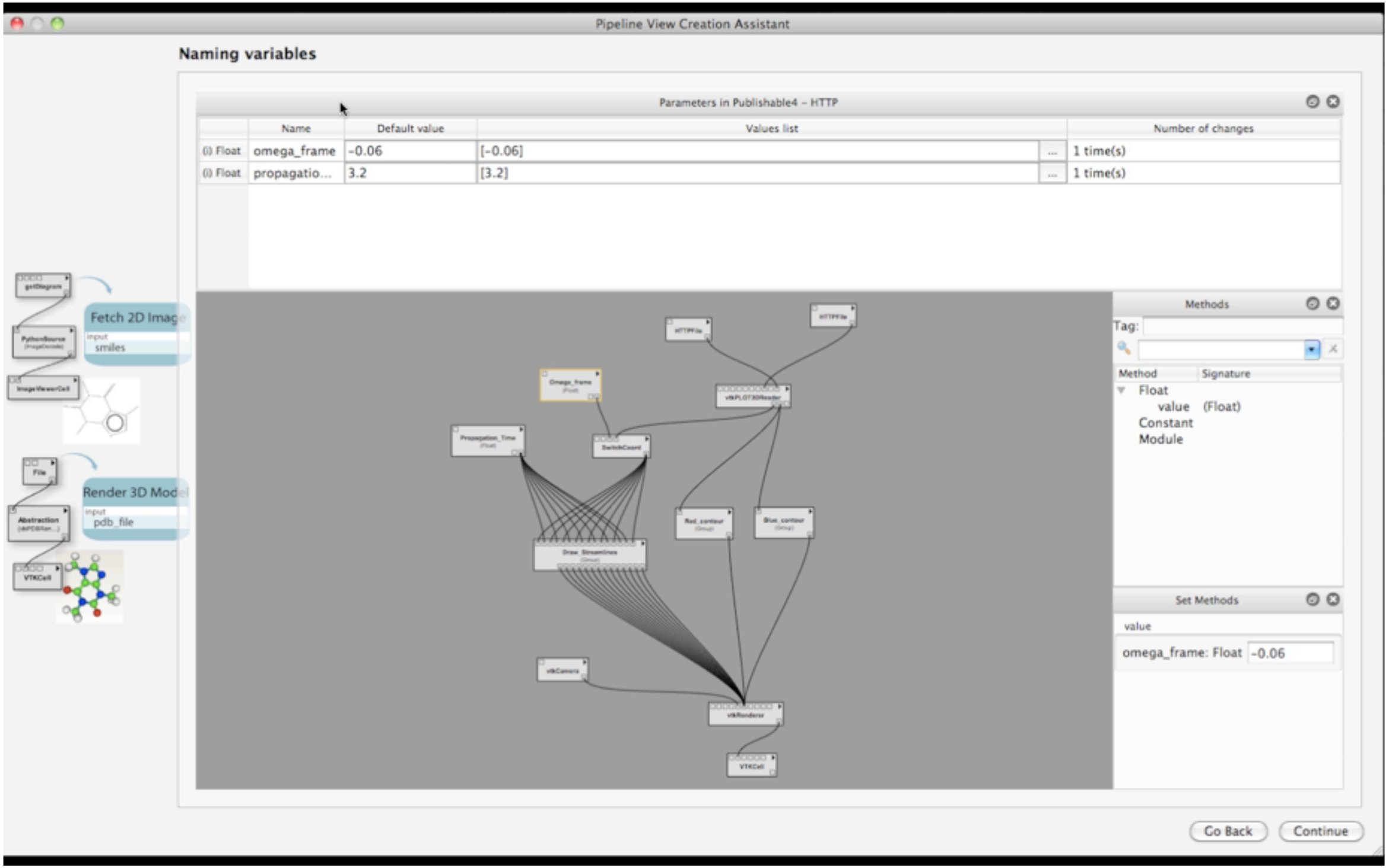
Mine pipeline provenance

Acquire and Analyze Pipelines → **Create Views (Simplify Pipelines)** → Combine Views → App generation and deployment

---

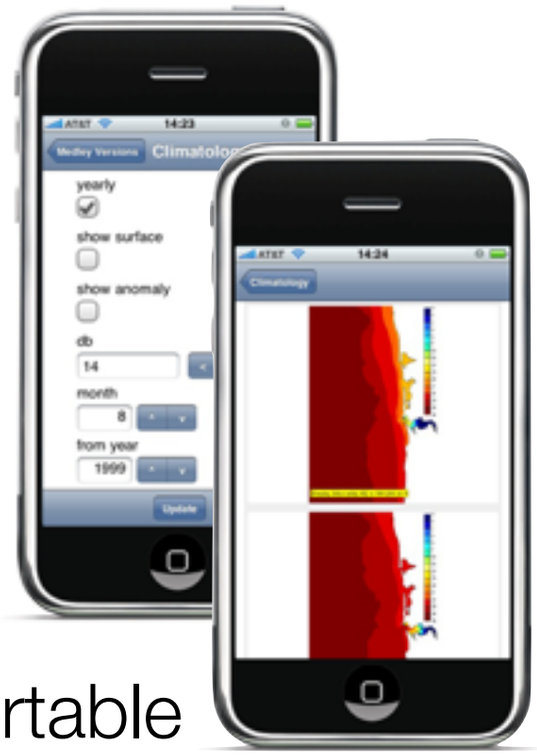


Acquire and Analyze Pipelines → Create Views (Simplify Pipelines) → Combine Views → App generation and deployment



VisMashup

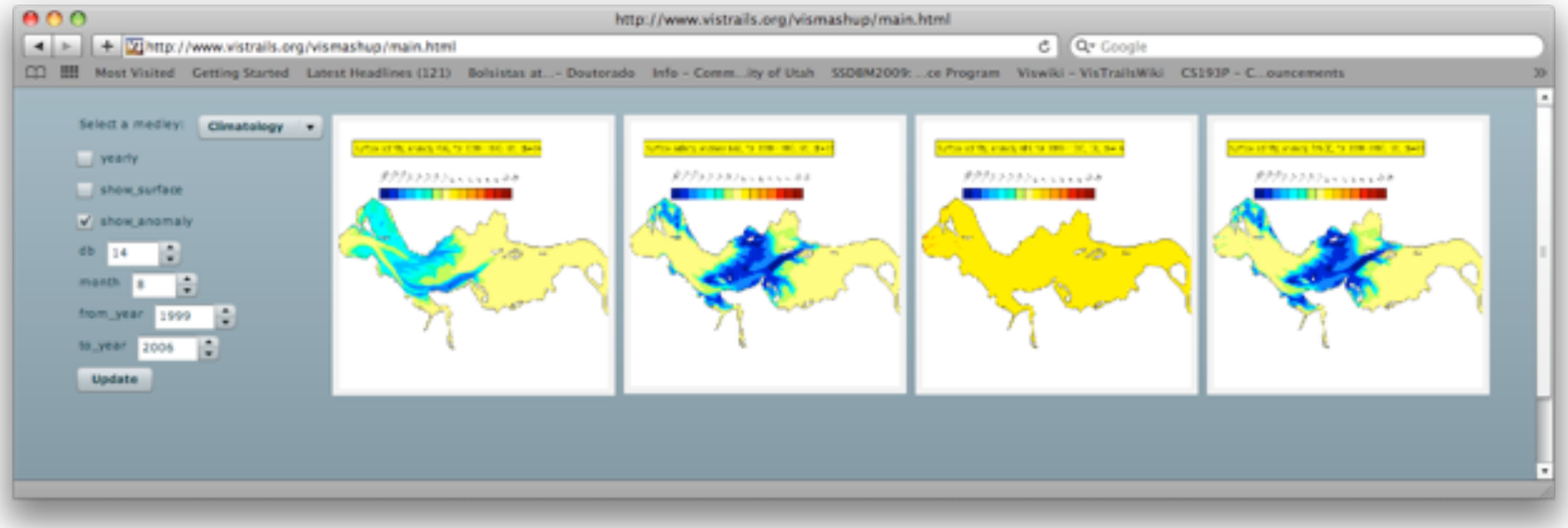
- Deploy visualization applications on different configurations

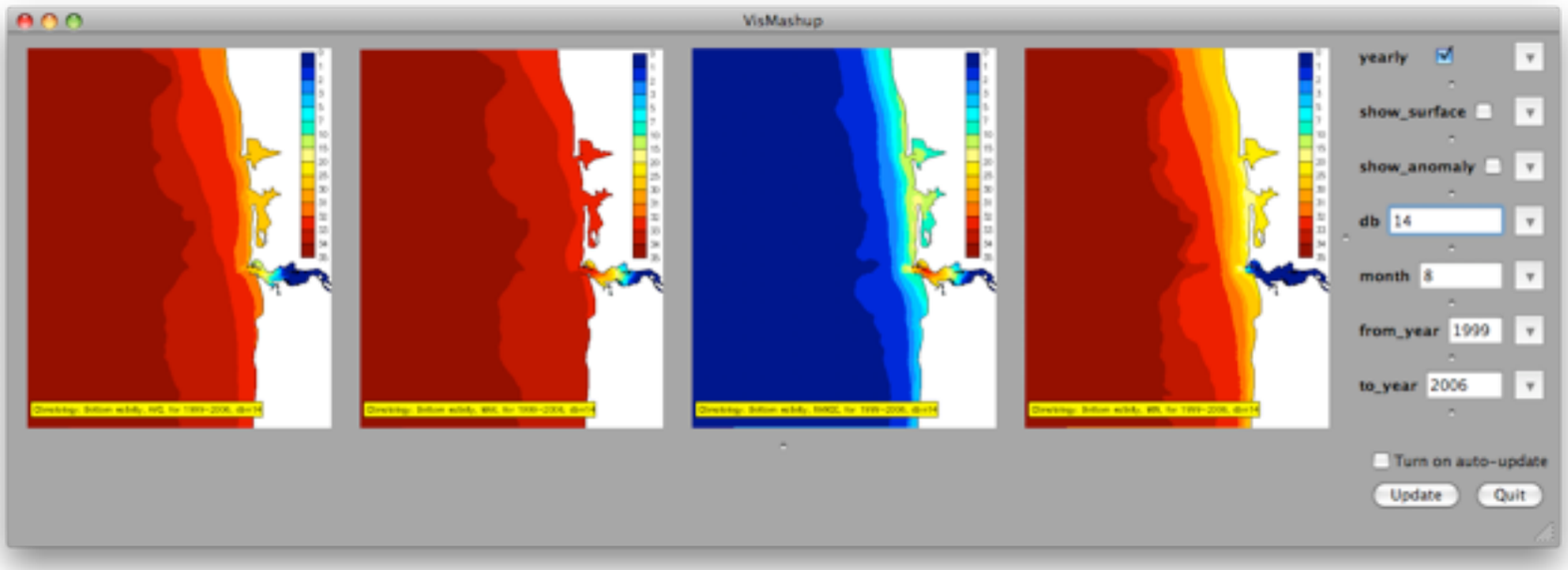


Portable Devices

(joint work with Wendel Silva, Patricia Hendricks, Juliana Freire and Cláudio Silva)

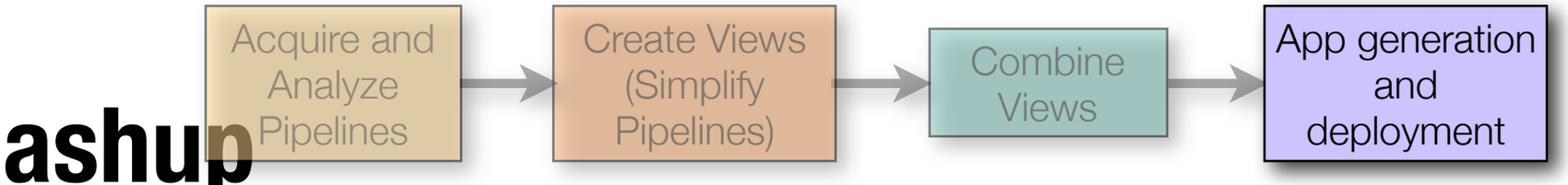
Web



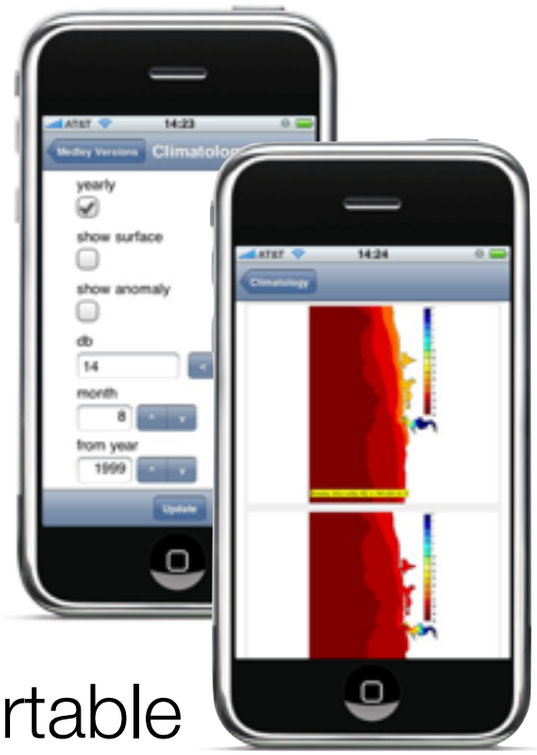


Desktop

VisMashup

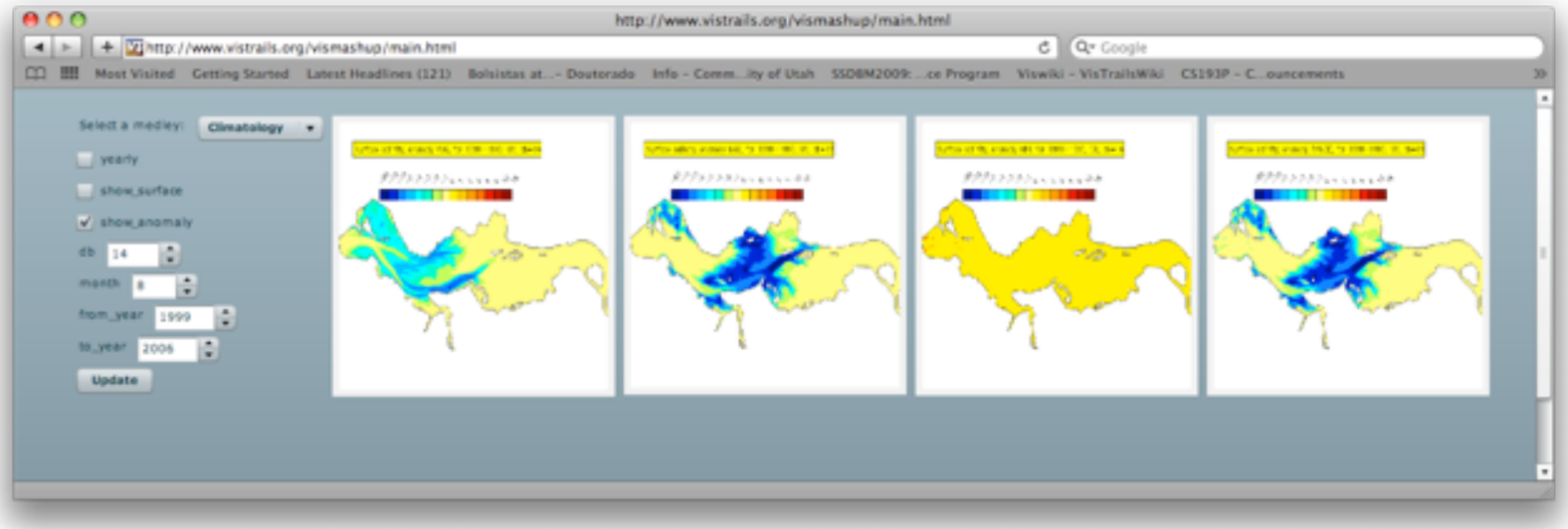


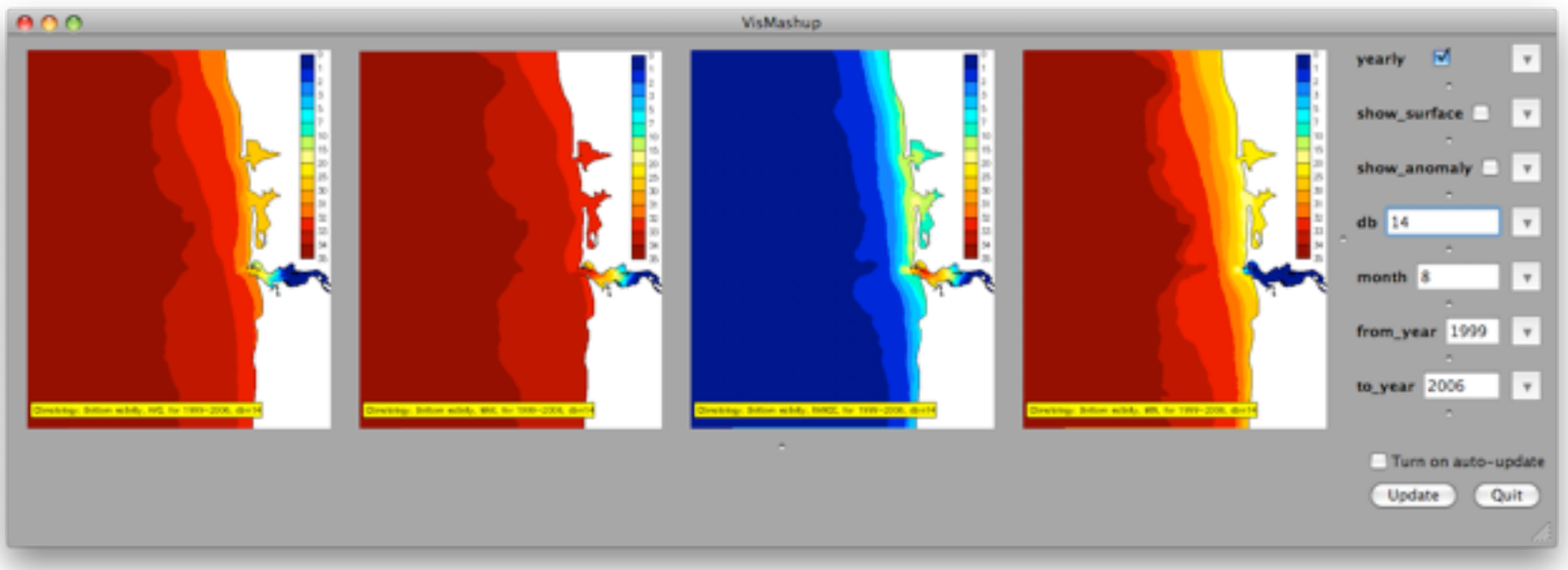
- Deploy visualization applications on different configurations



Portable Devices

Web





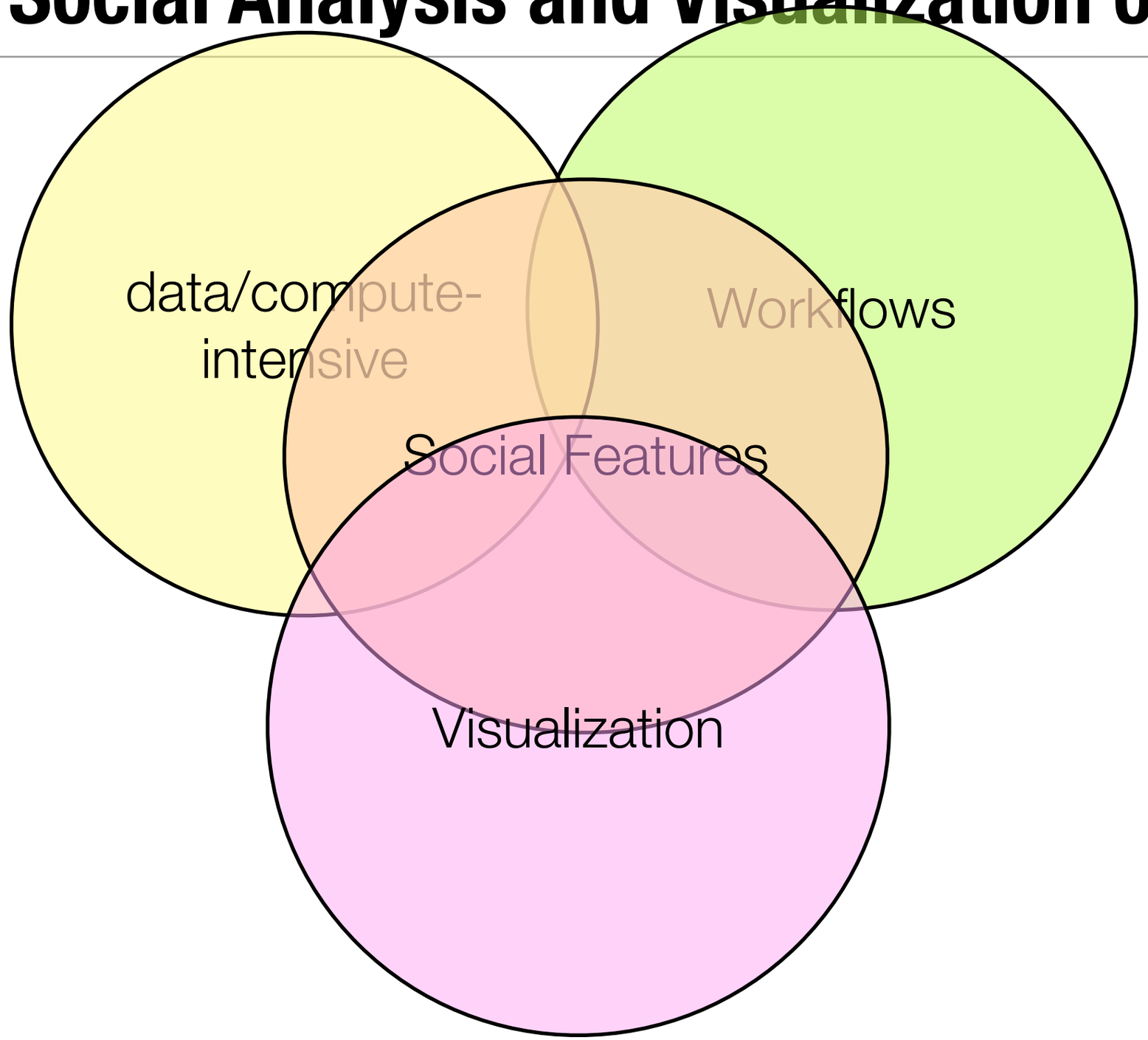
Desktop

(joint work with Wendel Silva, Patricia Hendricks, Juliana Freire and Cláudio Silva)

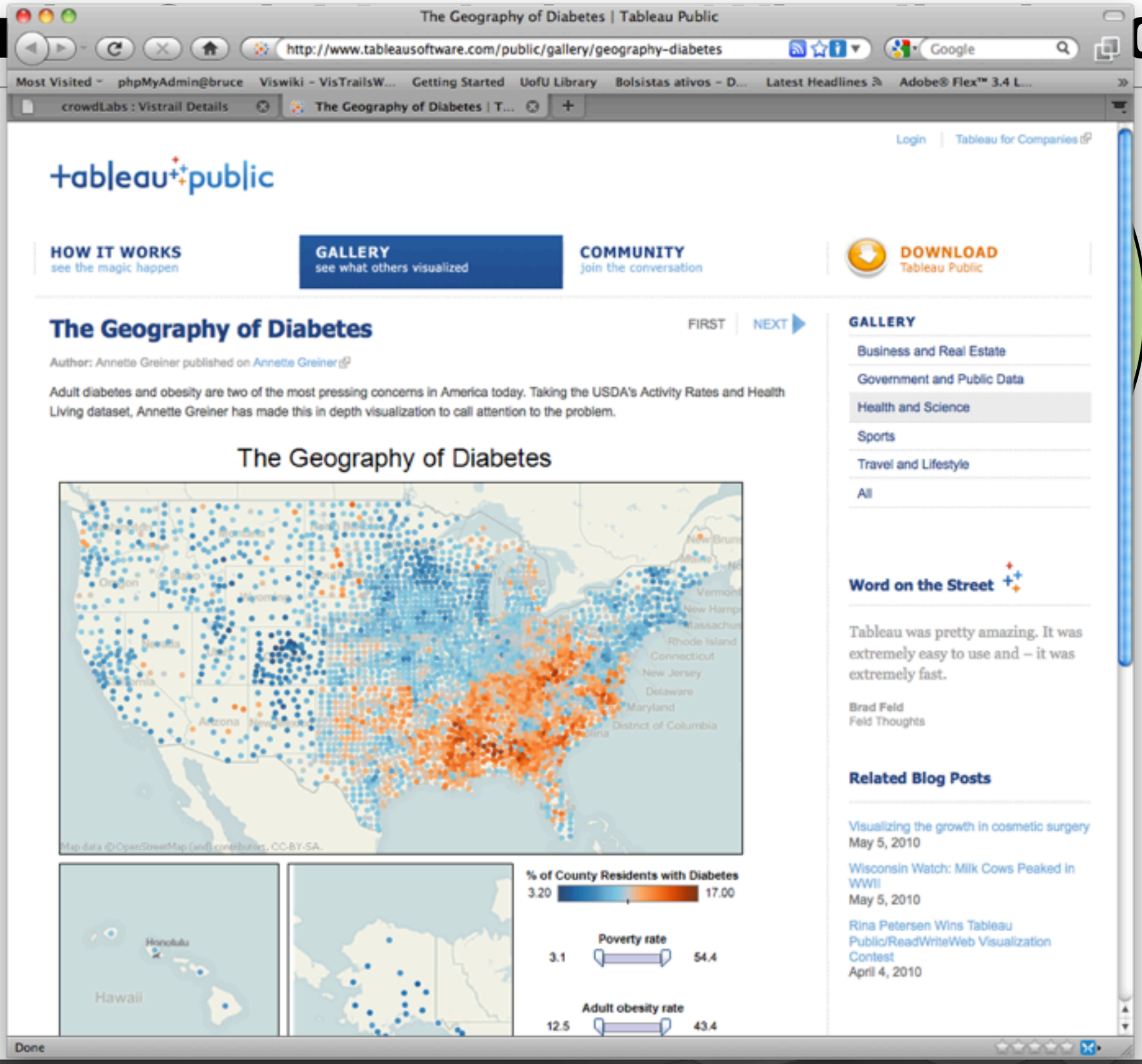
Outline

- Motivation
  - Scientific Research Scenario
- Framework
  - VisMashup
  - **CrowdLabs**
  - Provenance Rich Publications
- Demonstration
- Summary and Conclusions
- Future Work

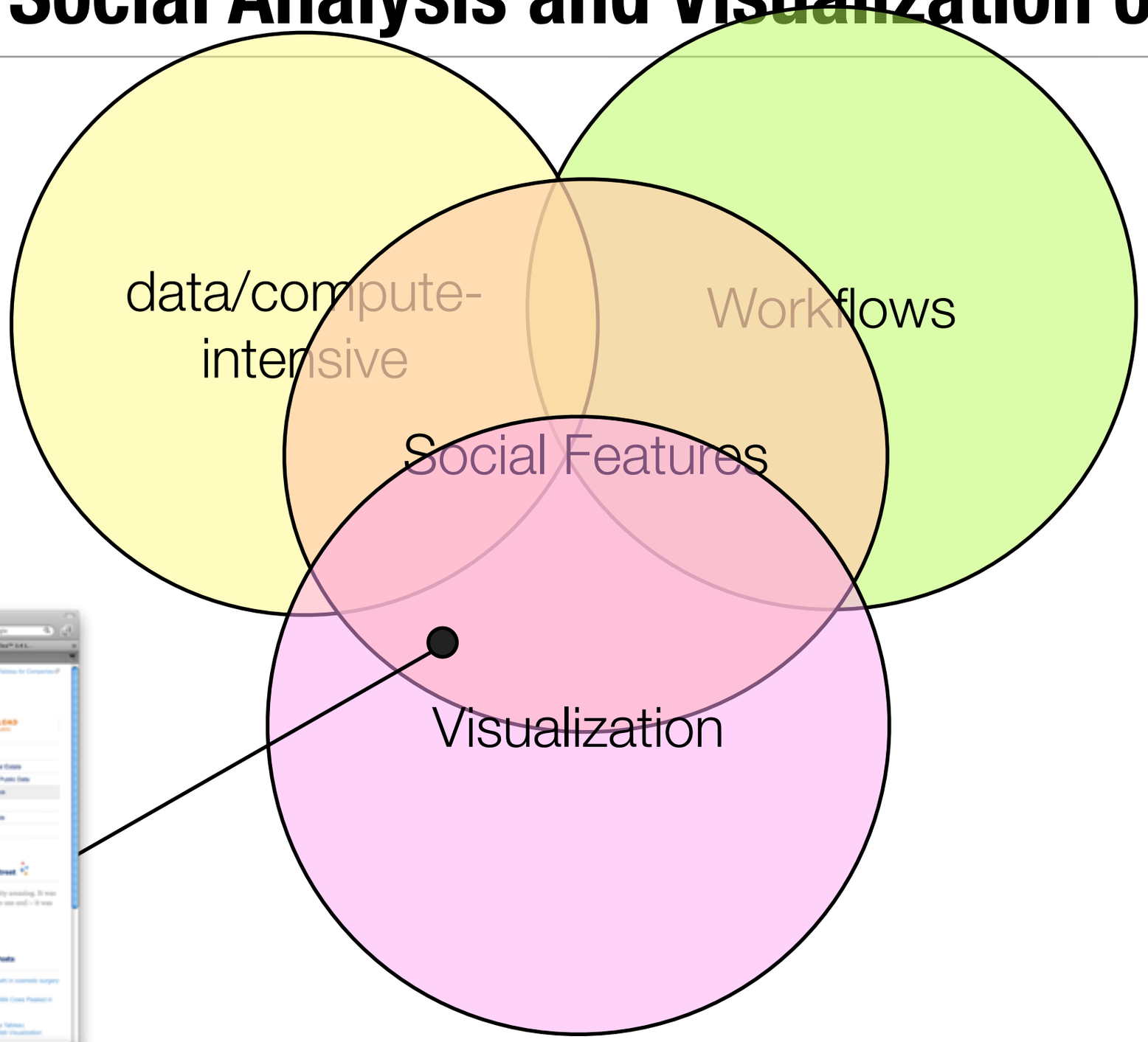
Supporting Social Analysis and Visualization of data



Suppor of data



Supporting Social Analysis and Visualization of data



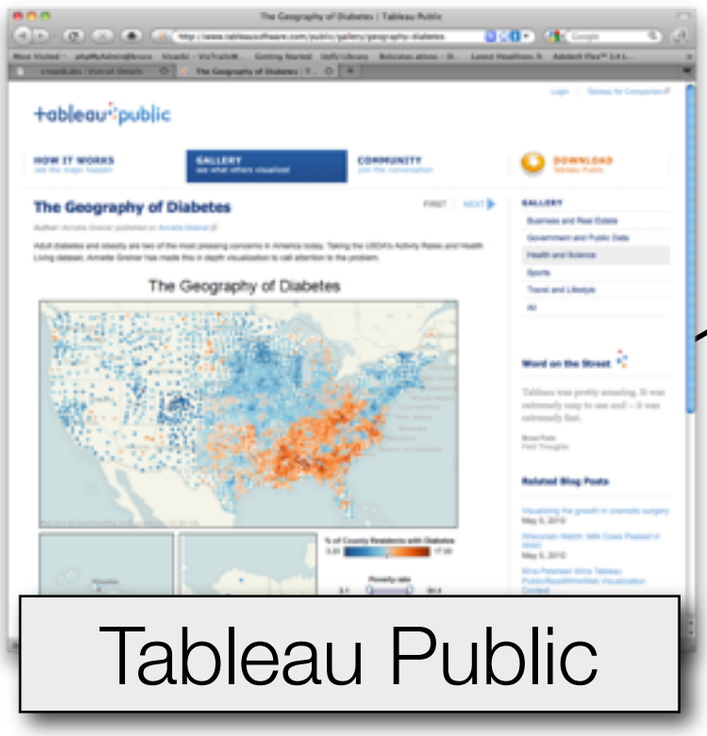
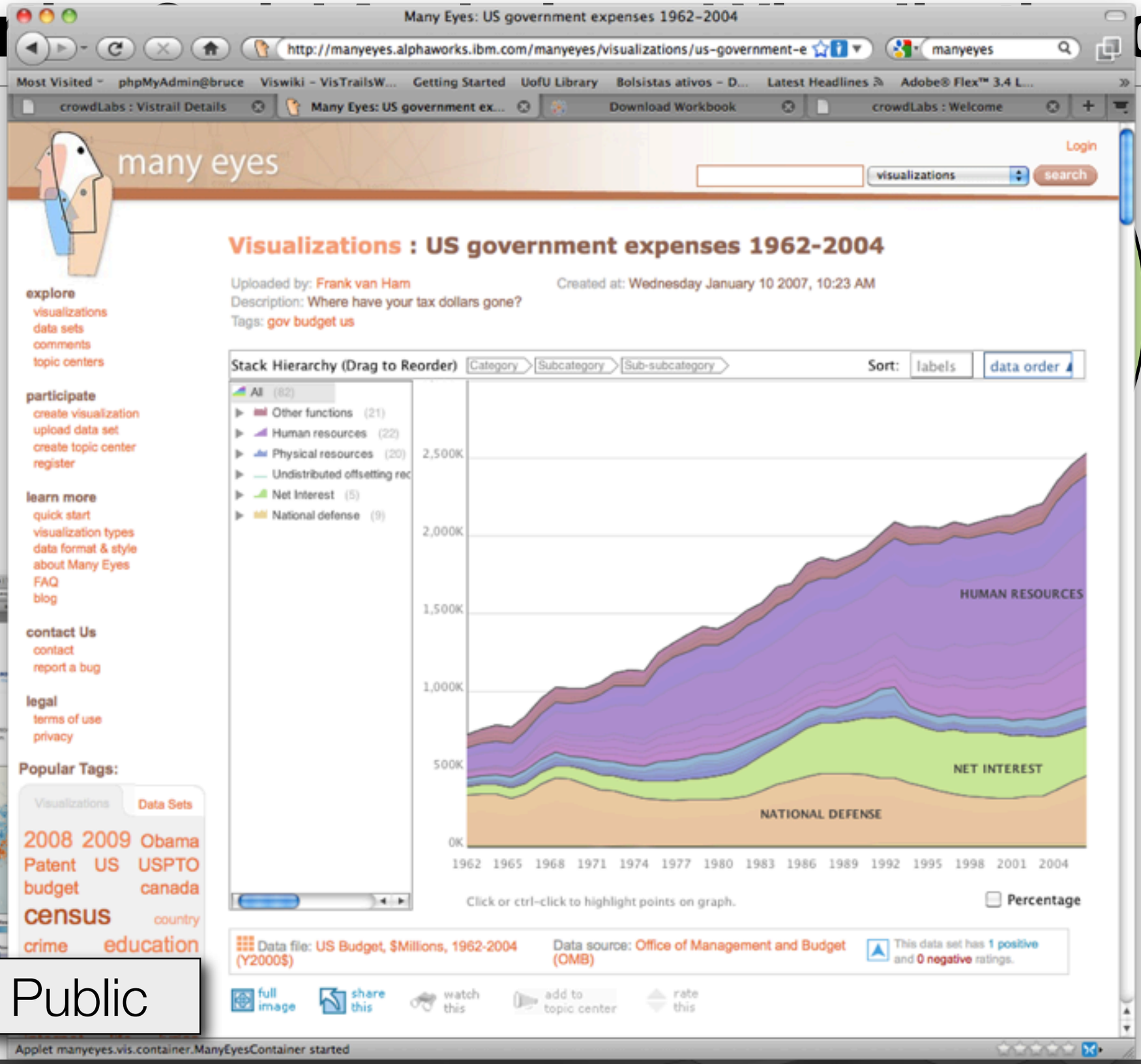


Tableau Public

Suppor of data



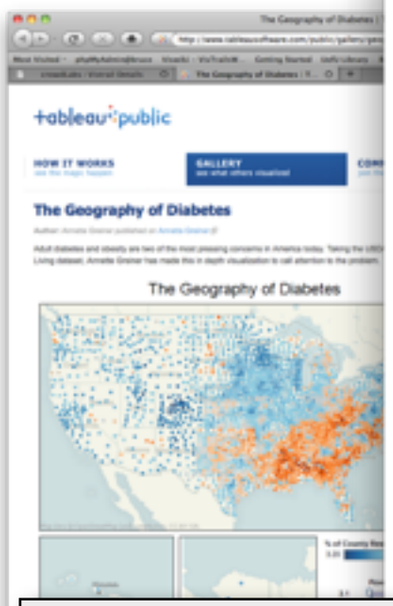
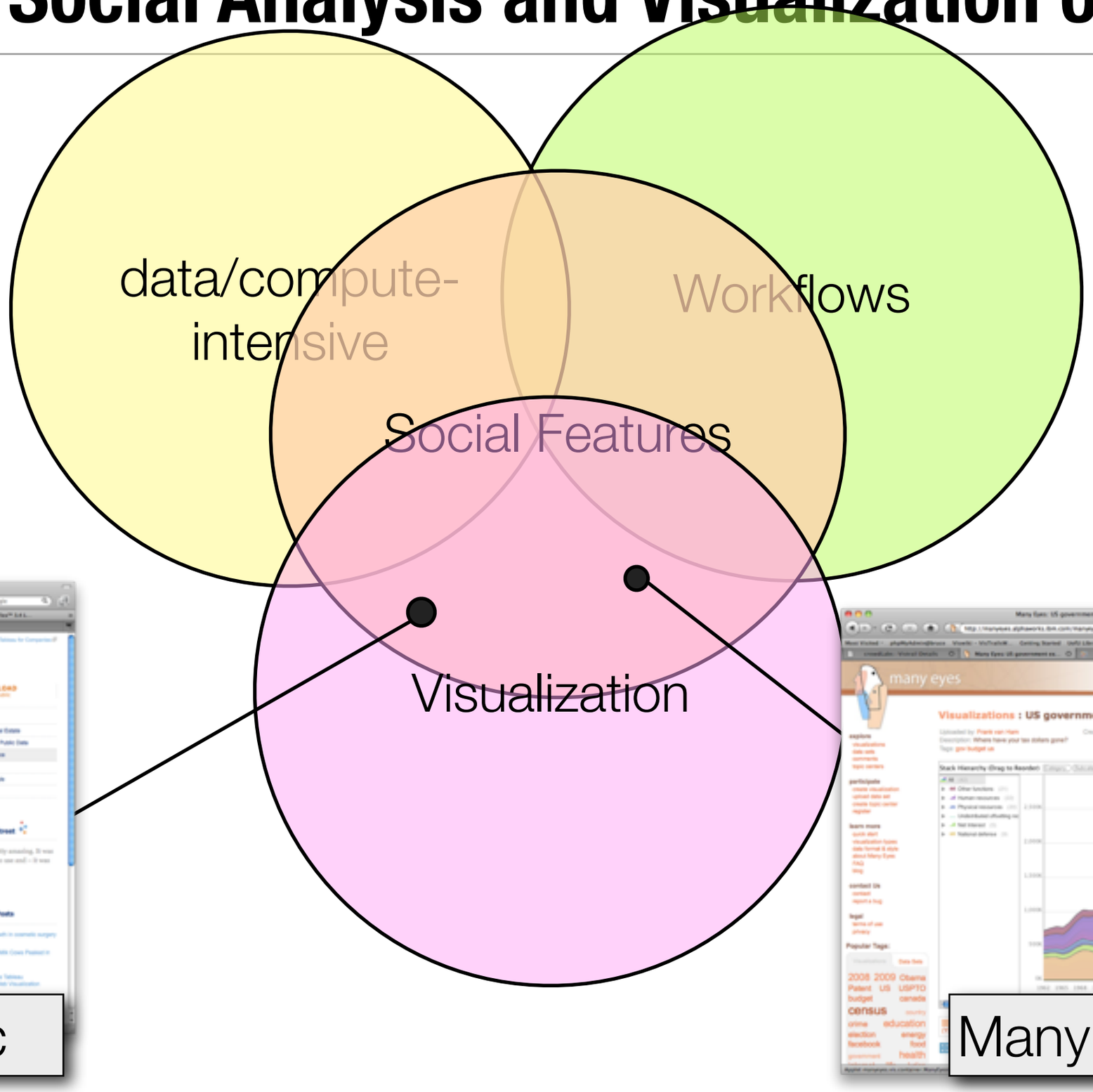


Tableau Public

Supporting Social Analysis and Visualization of data



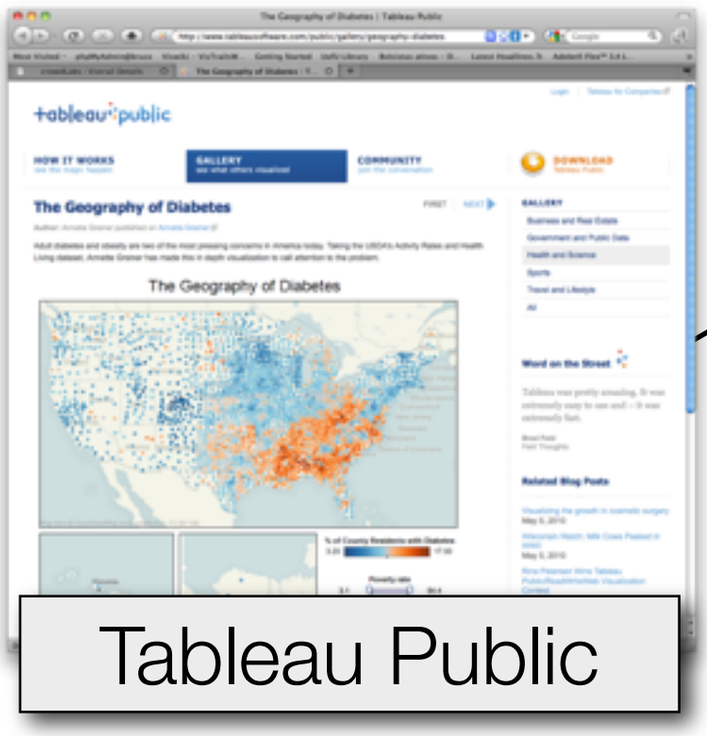
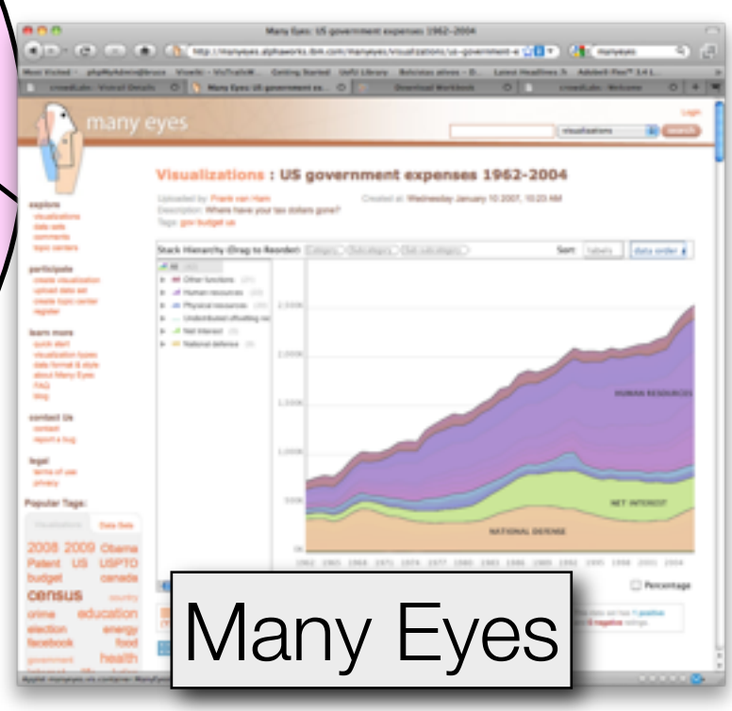
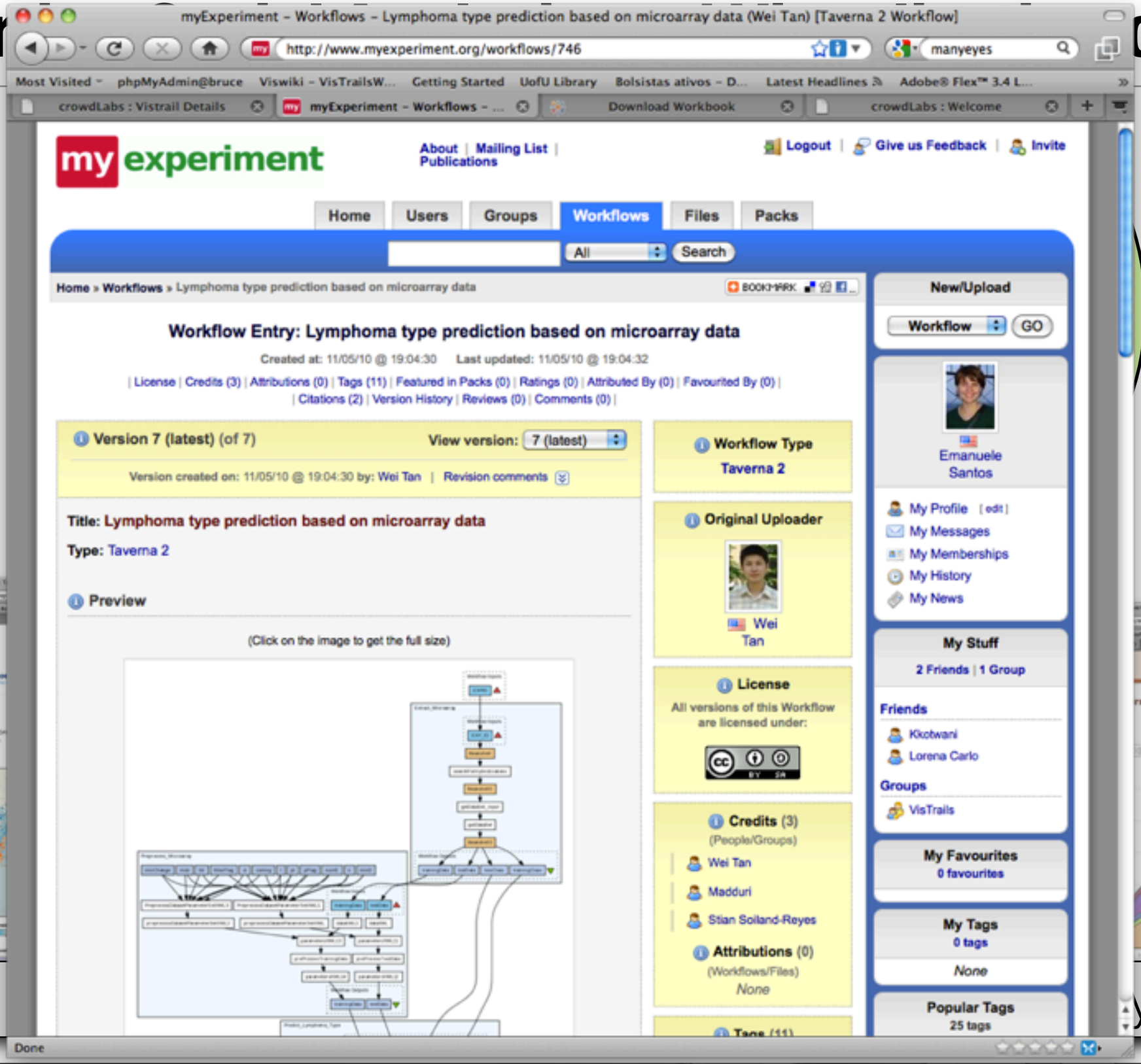


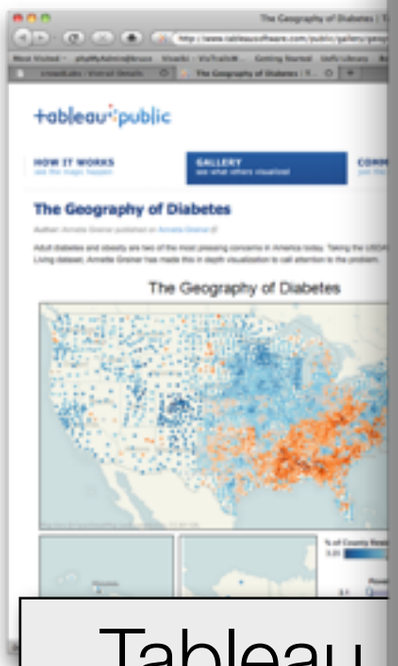
Tableau Public



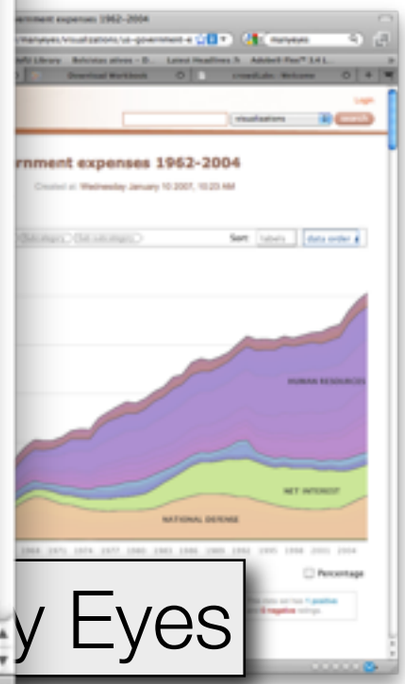
Many Eyes

Suppor of data



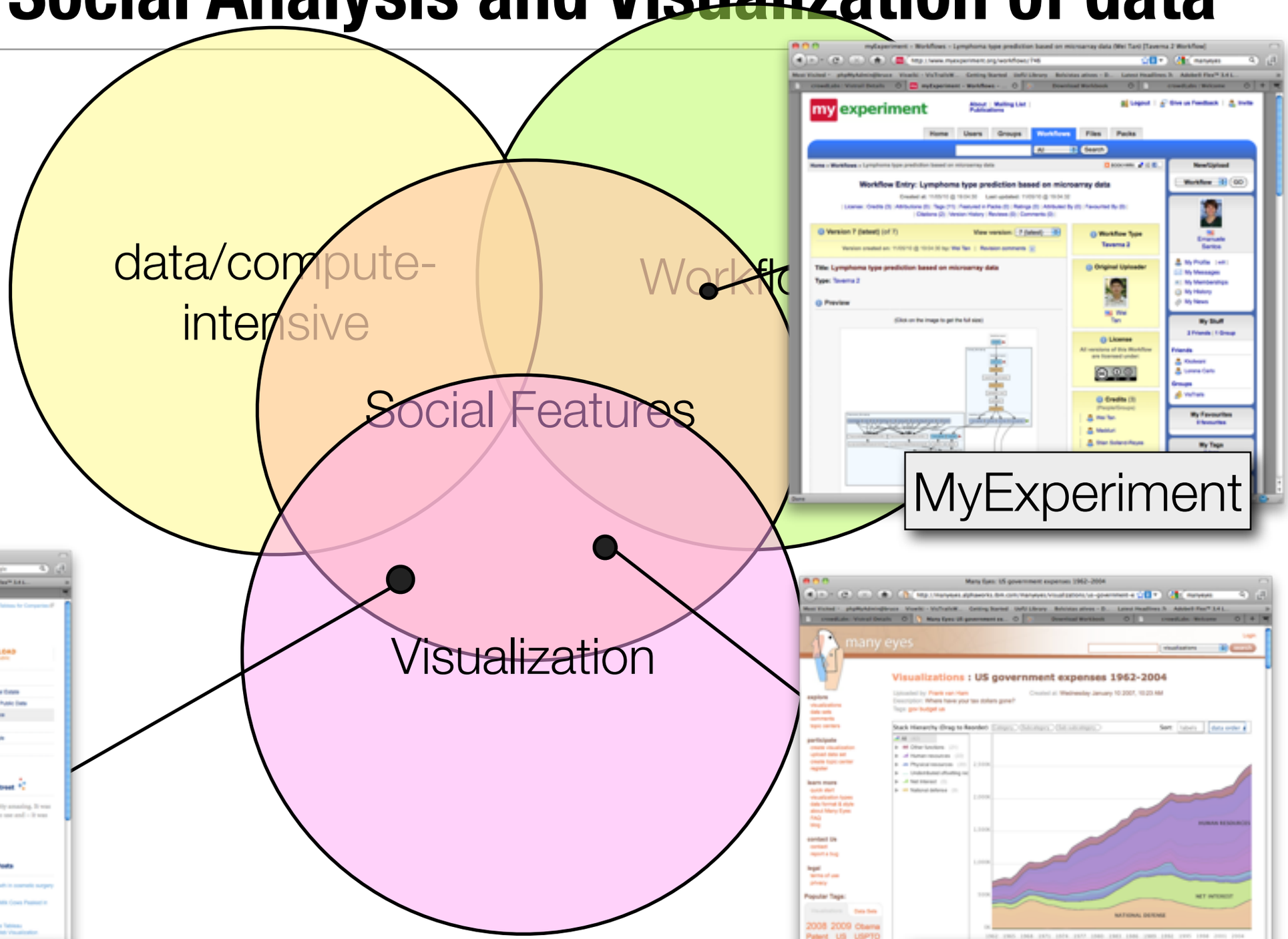


Tableau

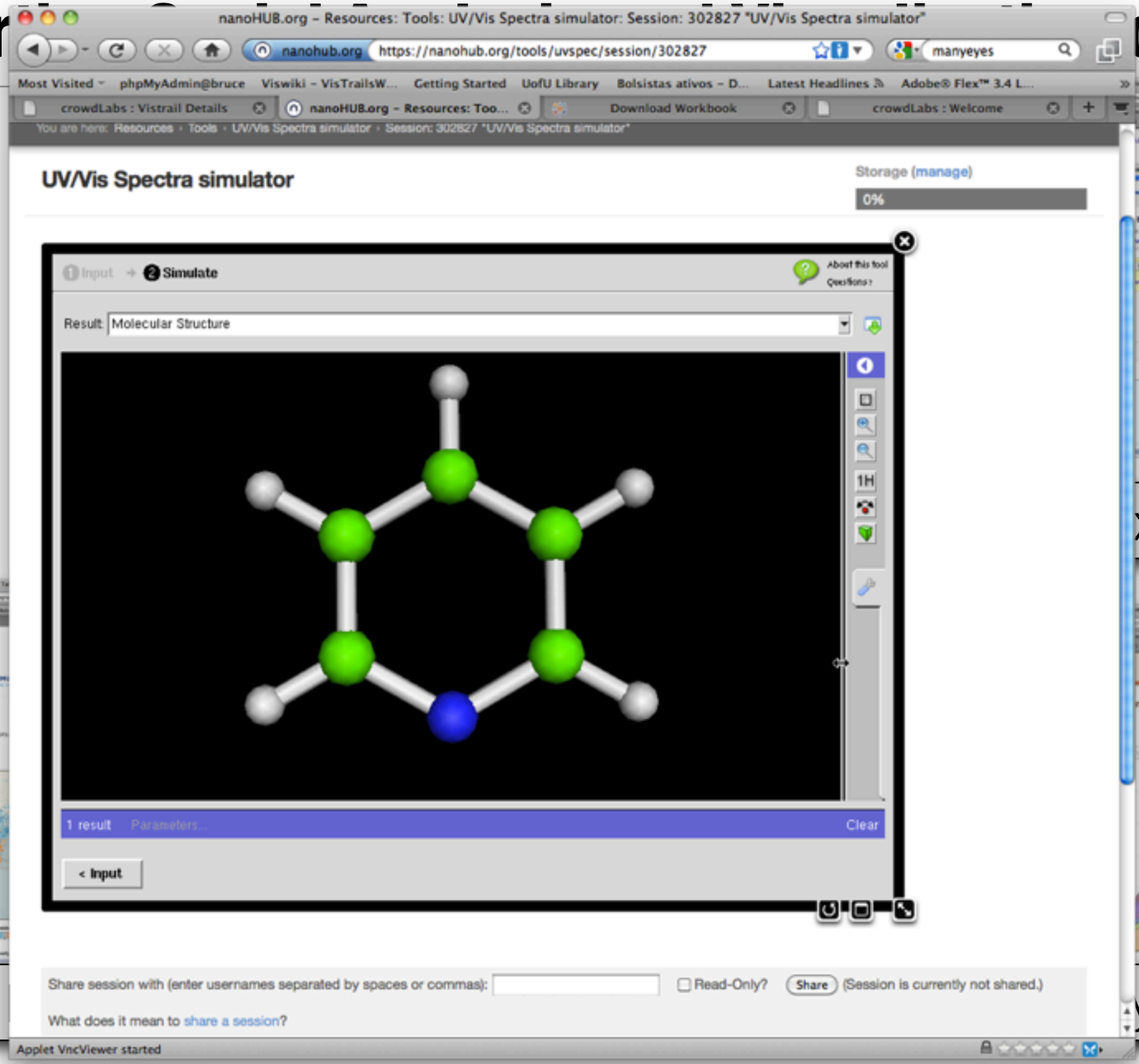


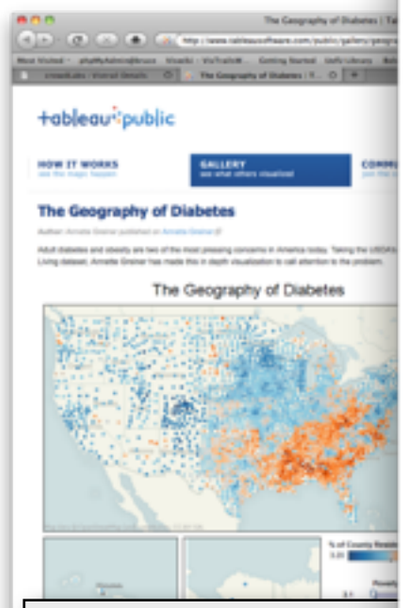
y Eyes

Supporting Social Analysis and Visualization of data



Suppor of data



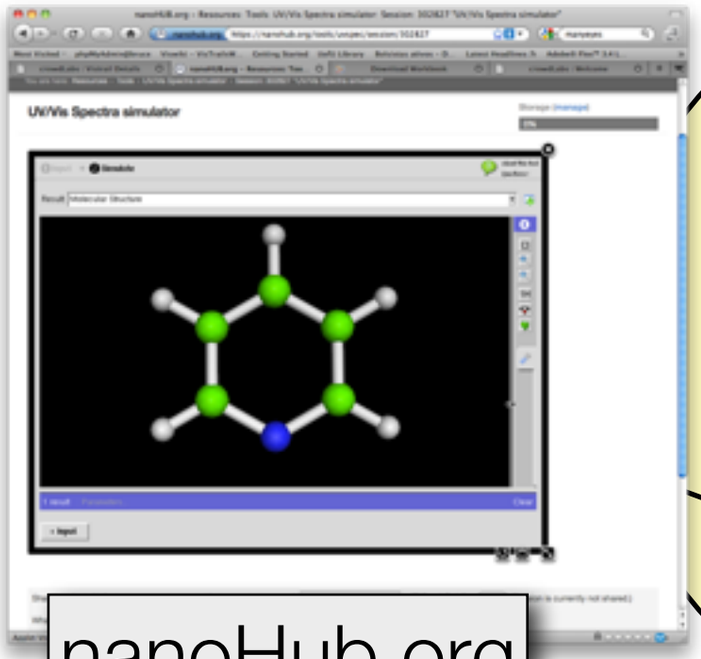


Tableau

xperiment

y Eyes

Supporting Social Analysis and Visualization of data



nanoHub.org

MyExperiment

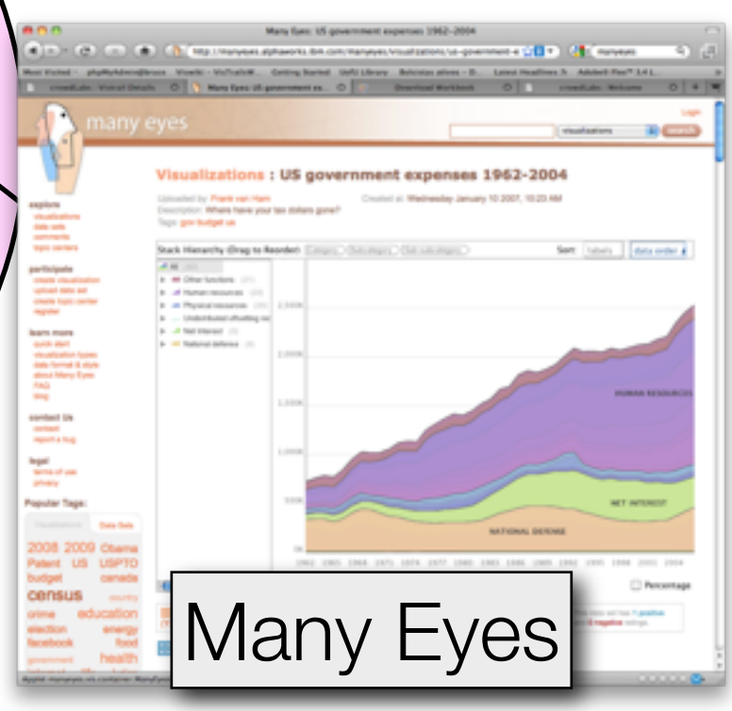
data/compute-intensive

Workflo

Social Features

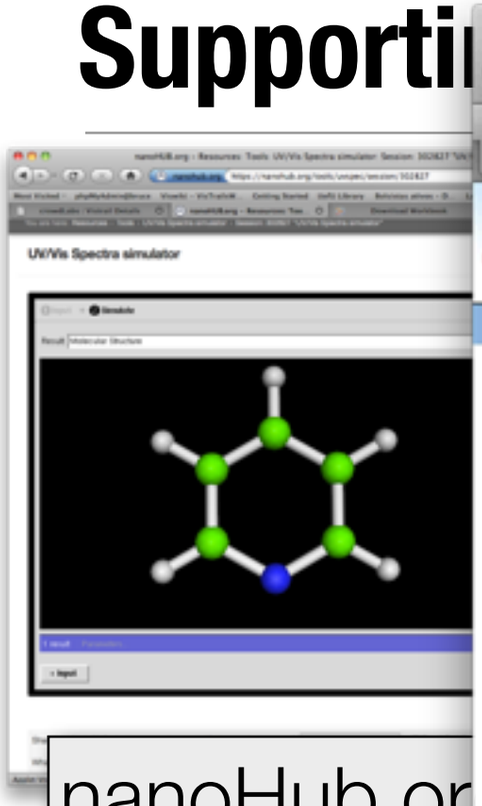
Visualization

Tableau Public



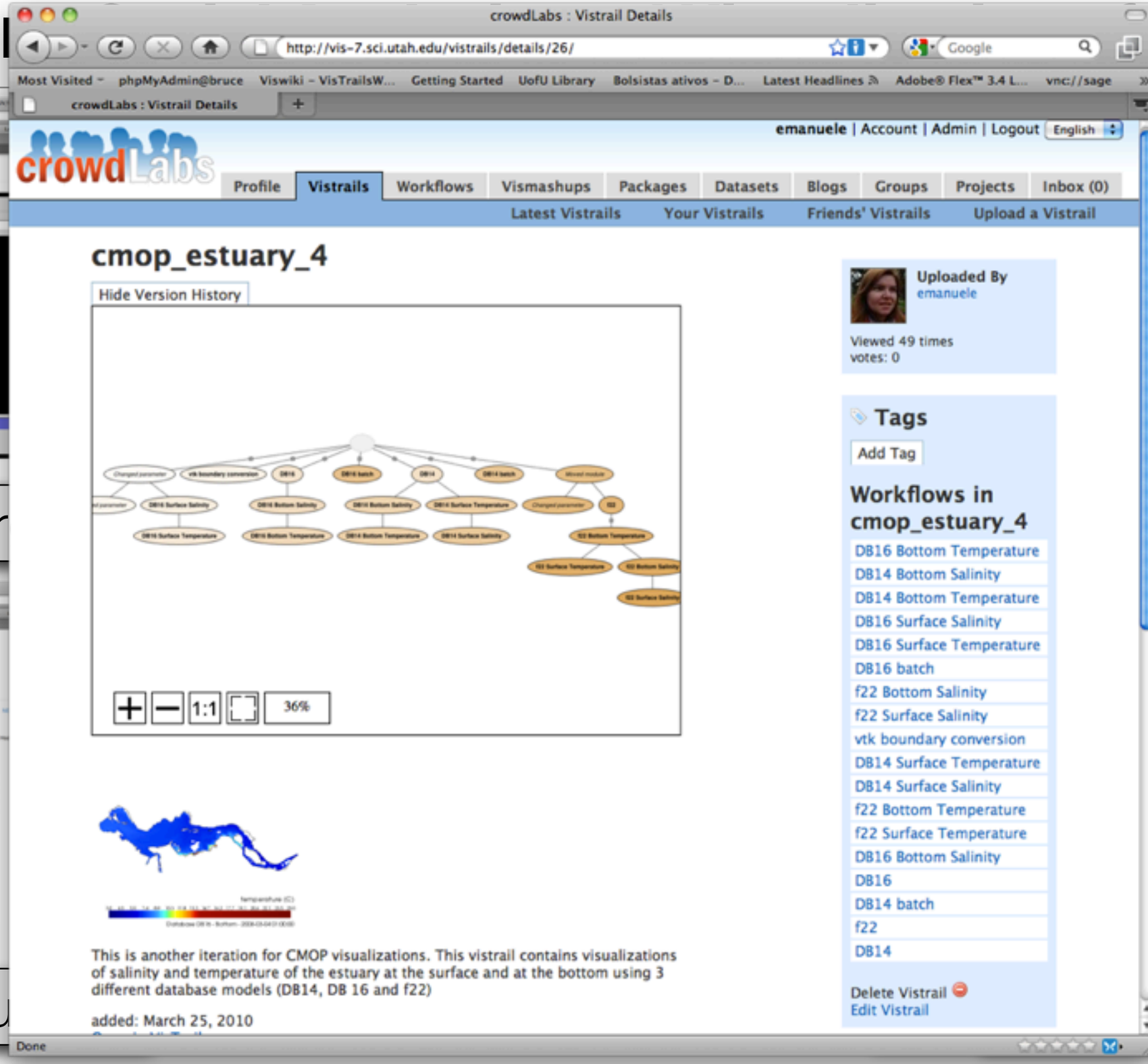
Many Eyes

Supporti data



nanoHub.or

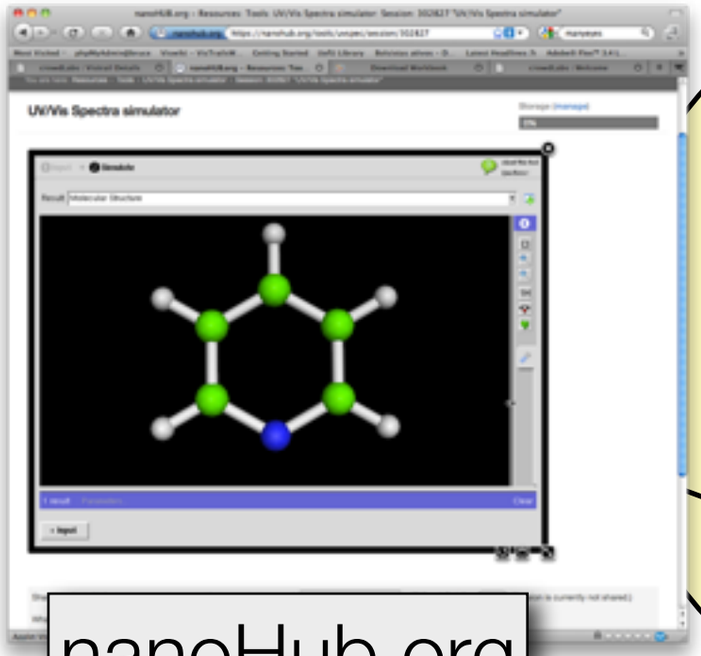
Tableau Pu



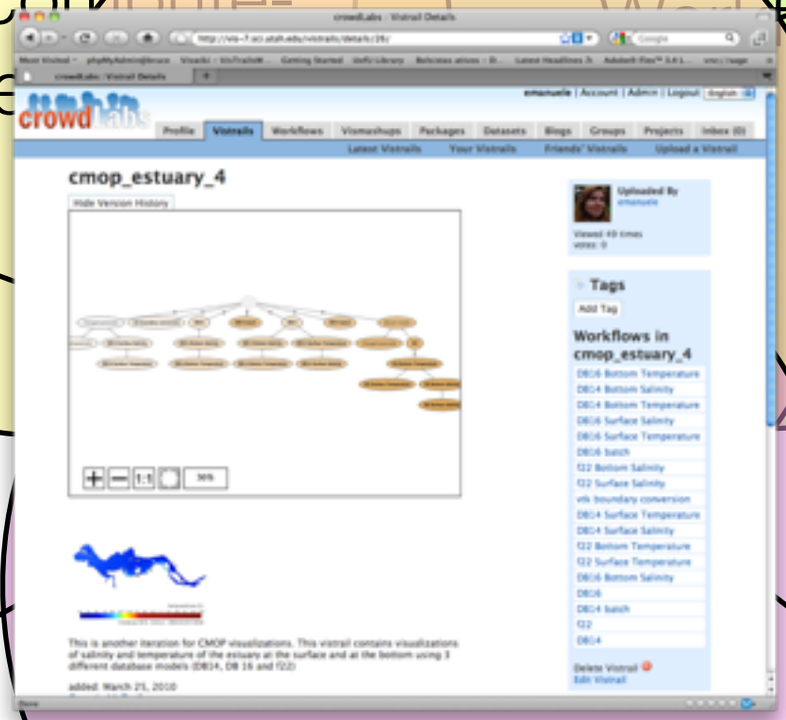
eriment

yes

Supporting Social Analysis and Visualization of data



nanoHub.org



MyExperiment

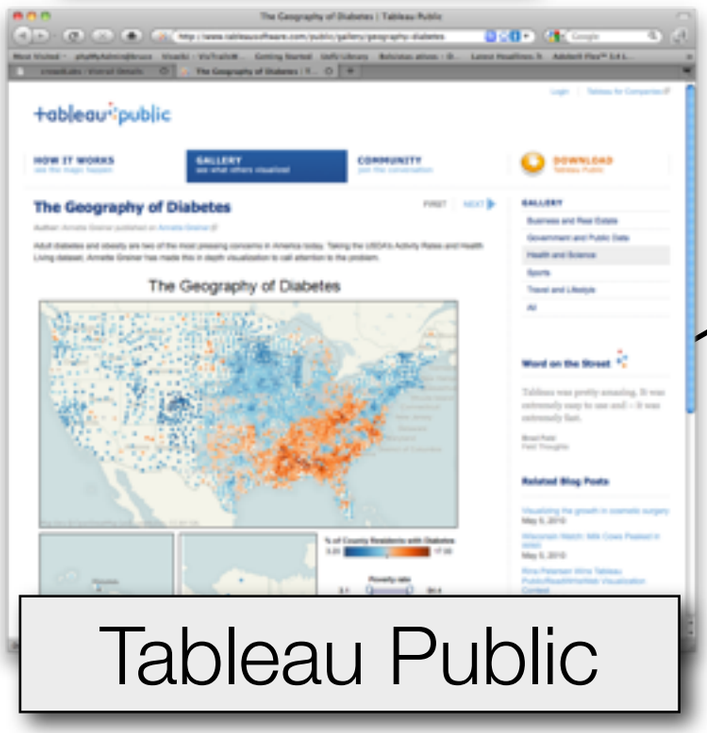
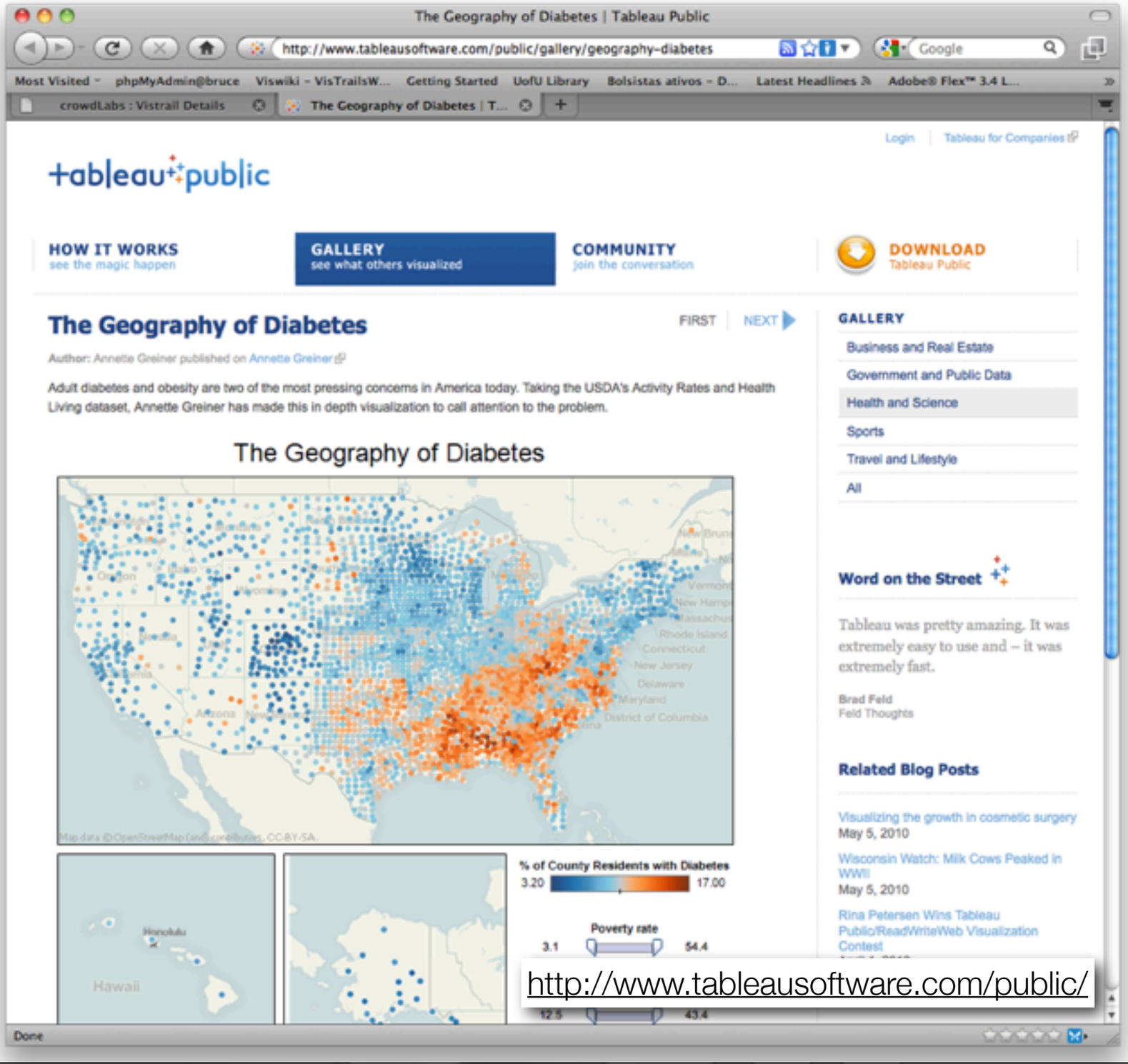


Tableau Public

Many Eyes

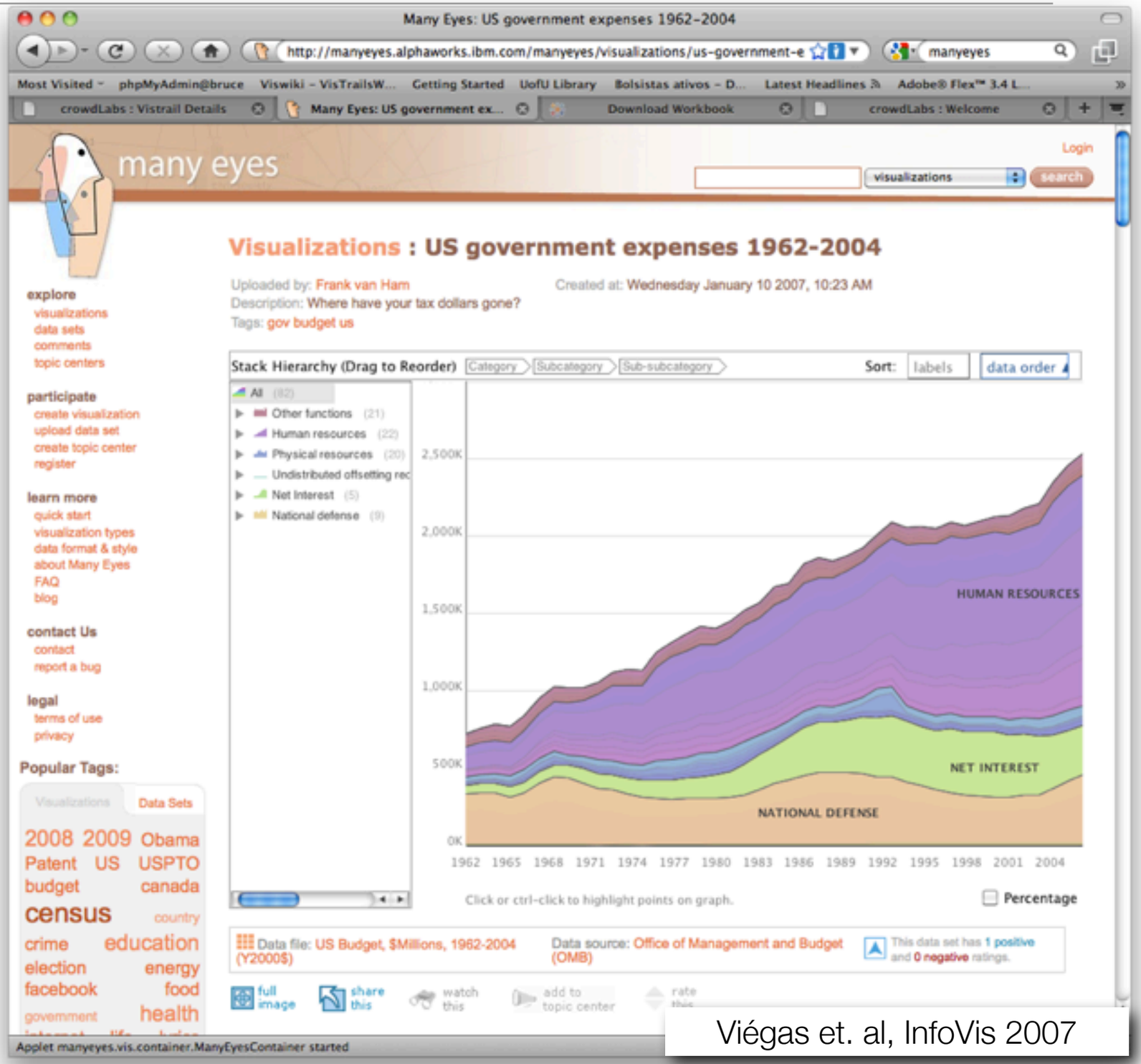
Supporting social analysis and visualization of data

- Tableau Public
  - Limited set of visualizations
  - Visualizations are built using Tableau and saved on server
  - Public data only
  - Processing on server side



Supporting social analysis and visualization of data

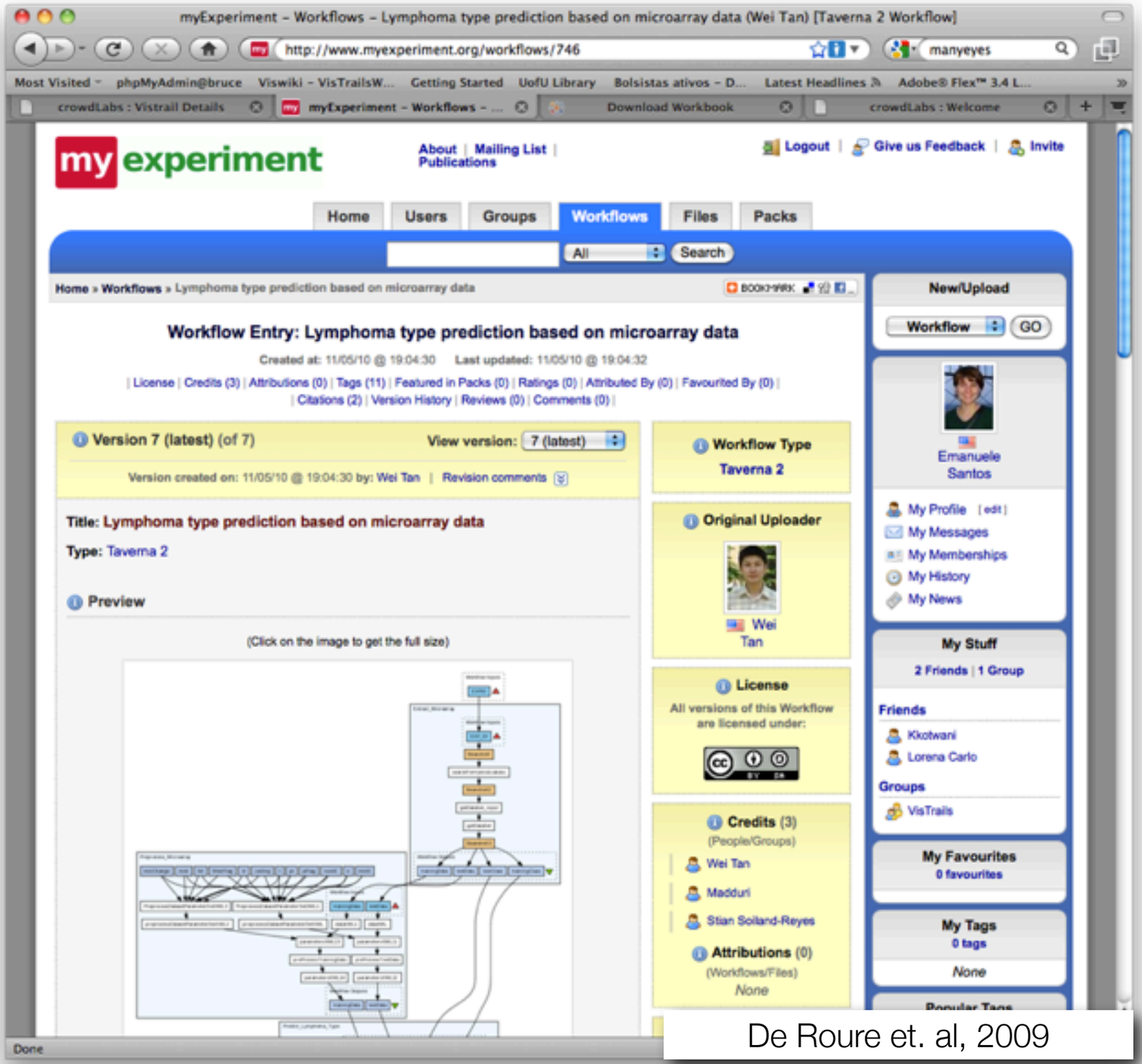
- Many Eyes
  - Limited set of visualizations
  - Limited data size
  - Visualizations are built online
  - Public data only
  - Processing on client side



Viégas et. al, InfoVis 2007

Supporting social analysis and workflows

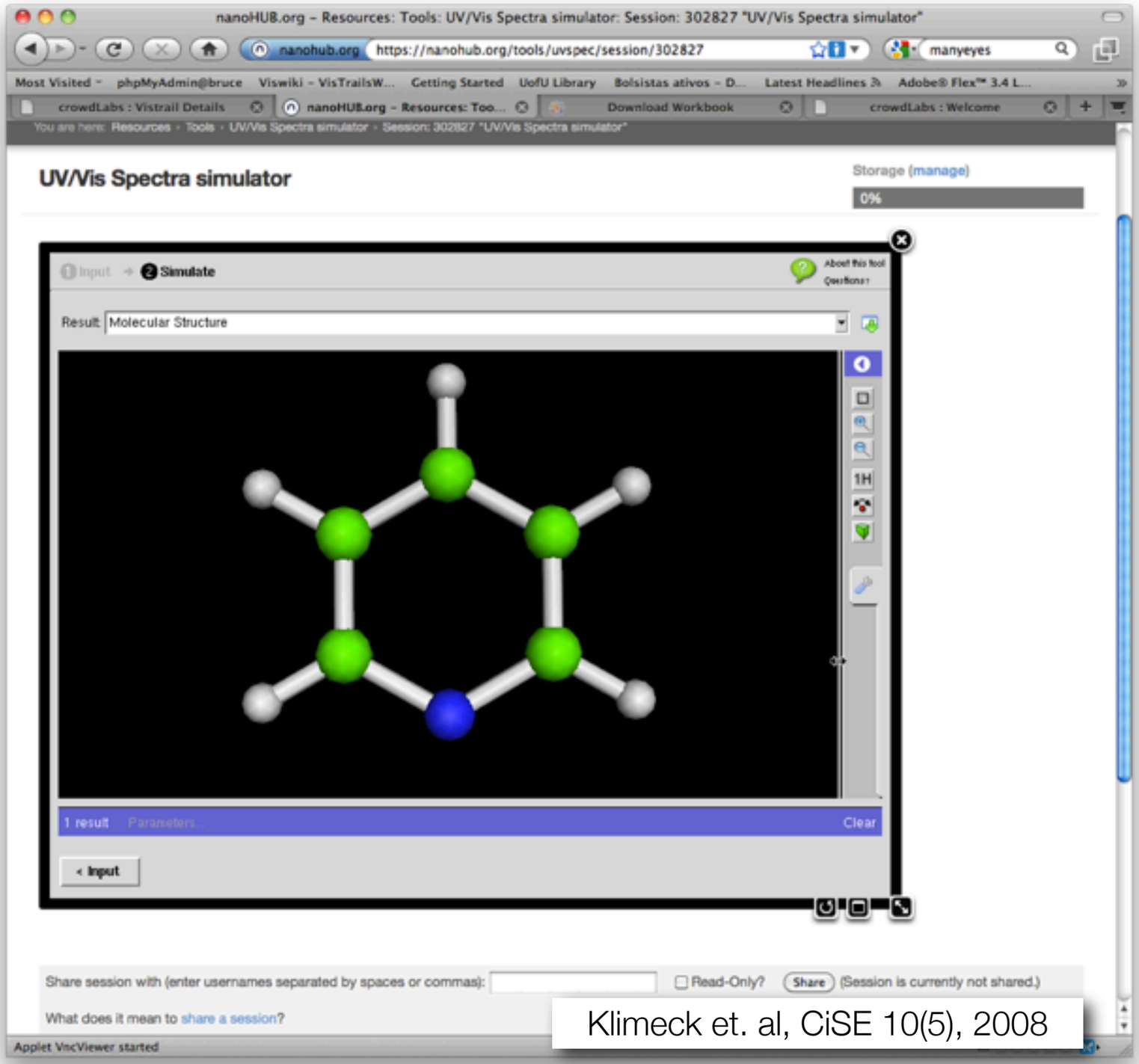
- myExperiment
  - supports workflow versions
  - focus on bioinformatics-related Web services



De Roure et. al, 2009

Online compute-intensive simulation

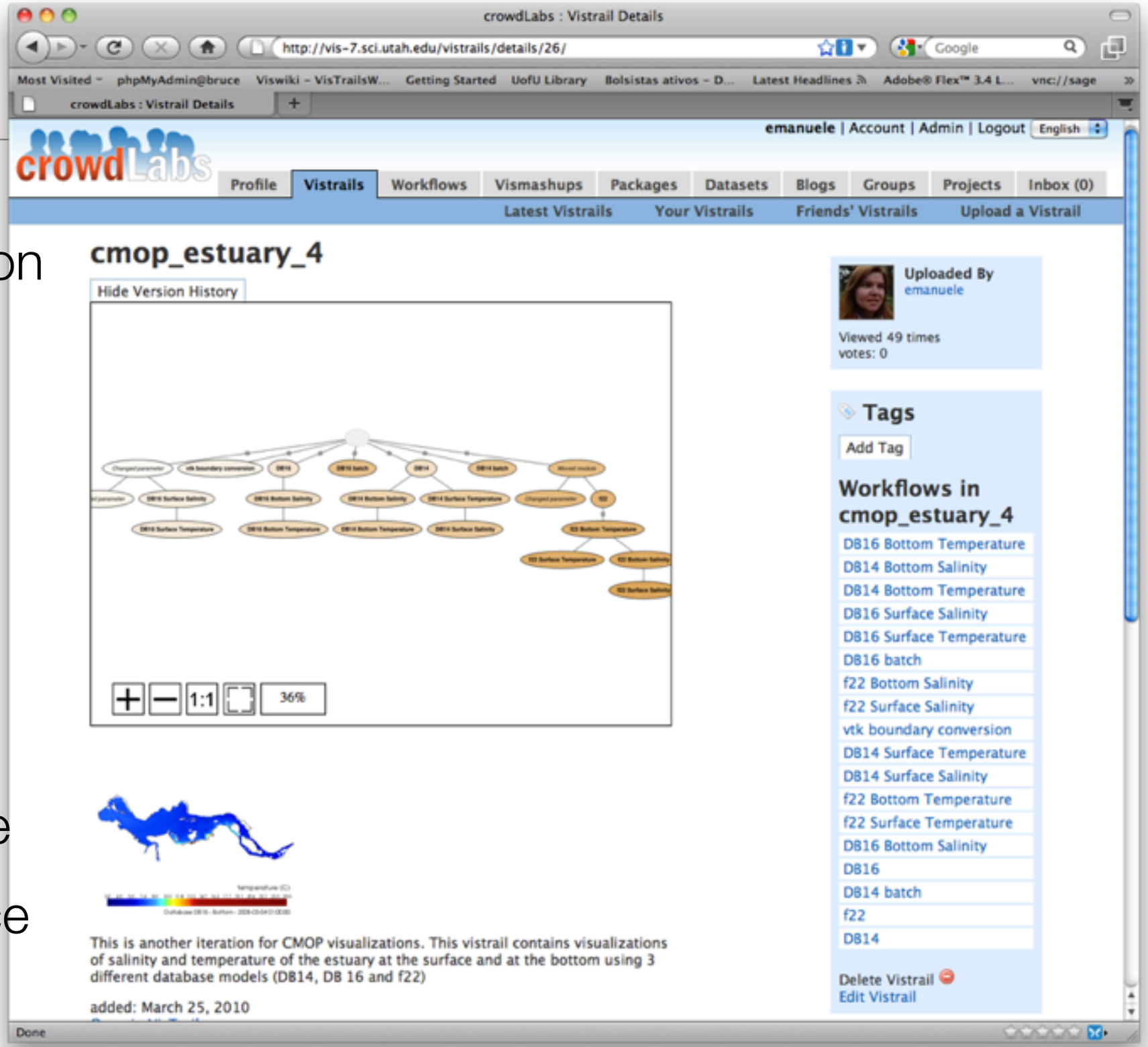
- nanoHub.org
  - Access to HPC resources (grid, cloud)
  - process on server side (using virtual machines)
  - visualization support through VNC
  - tools compiled on a cluster node



Klimeck et. al, CiSE 10(5), 2008

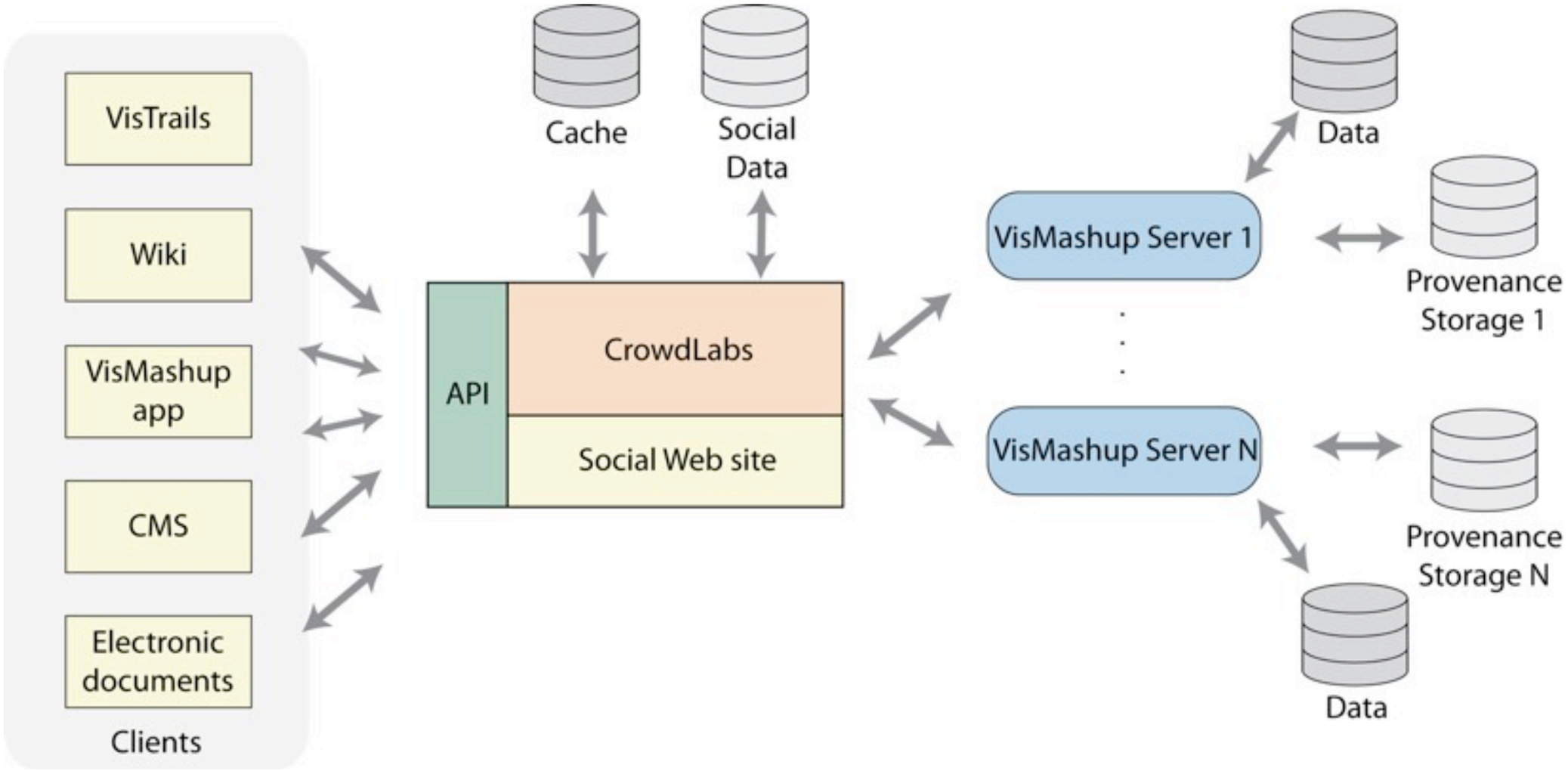
crowdlabs.org

- Facilitates collaboration and sharing among scientists and visualization experts
- Adopts Social Web Sites Model
- Integrates a set of usable tools and a scalable infrastructure
- Leverages provenance information



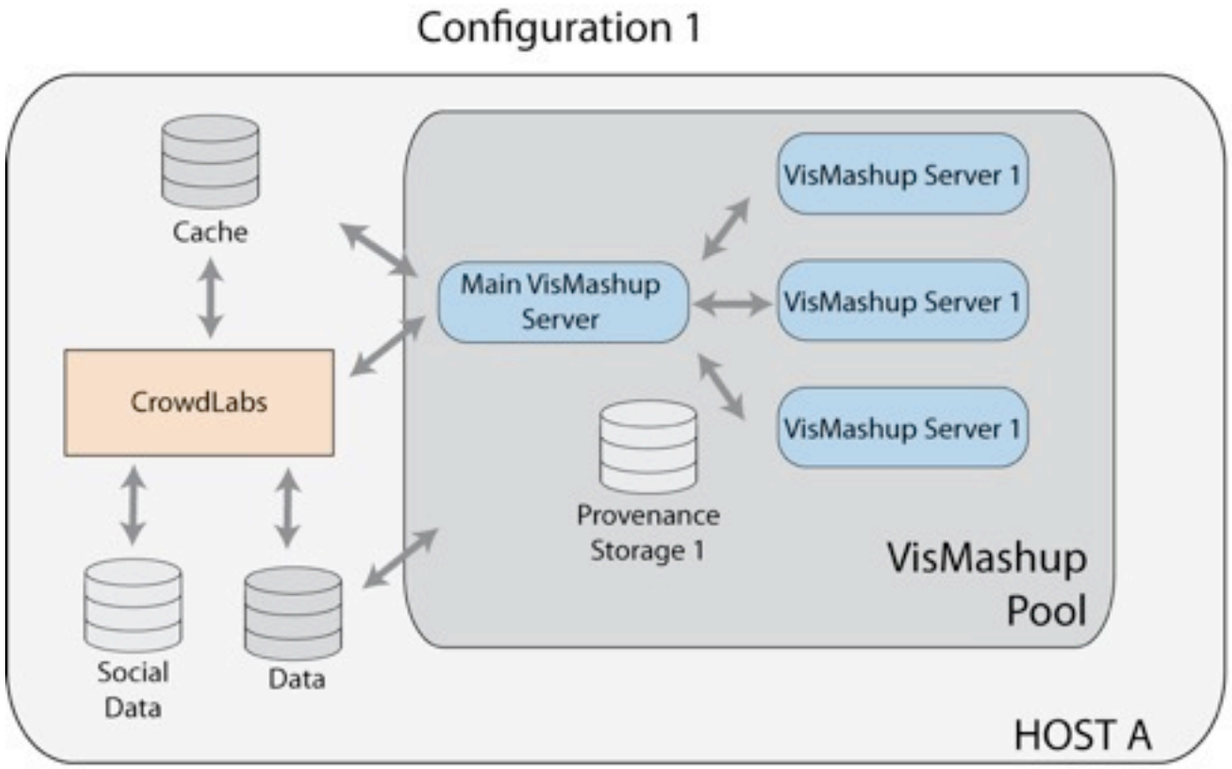
(joint work with Phillip Mates, Juliana Freire and Cláudio Silva)

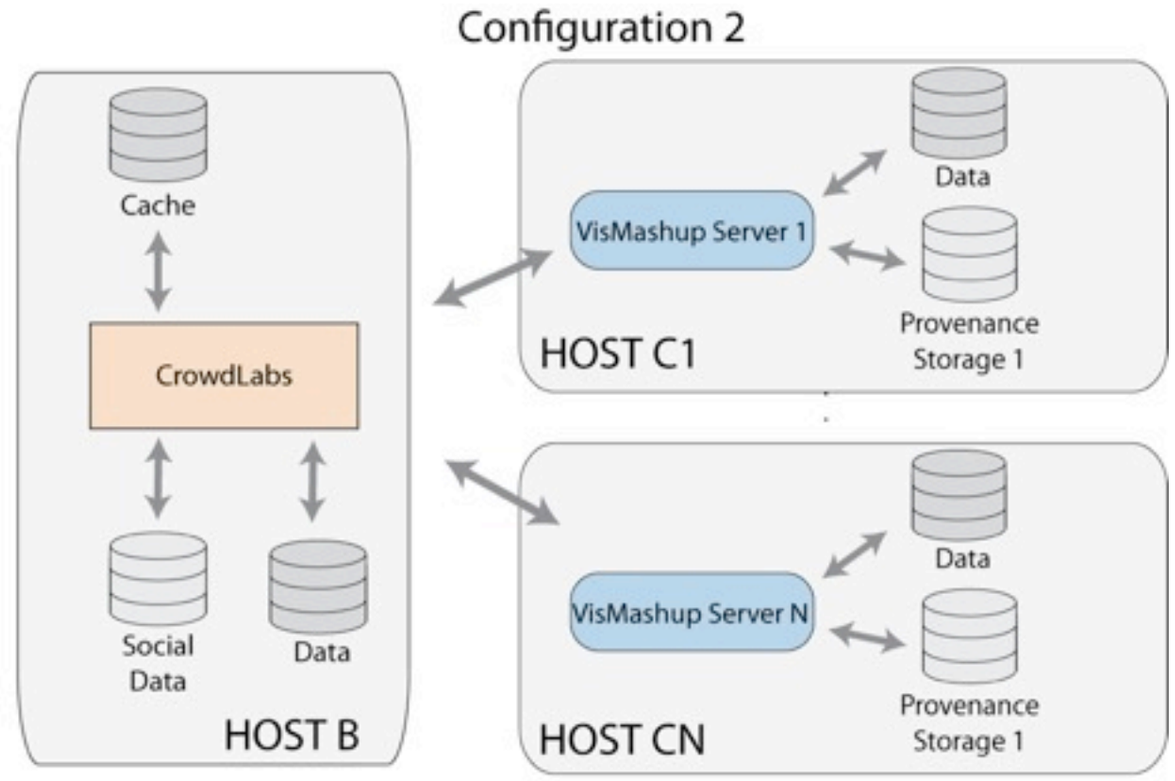
CrowdLabs System Configuration



Provenance is made accessible through the Web site and an API

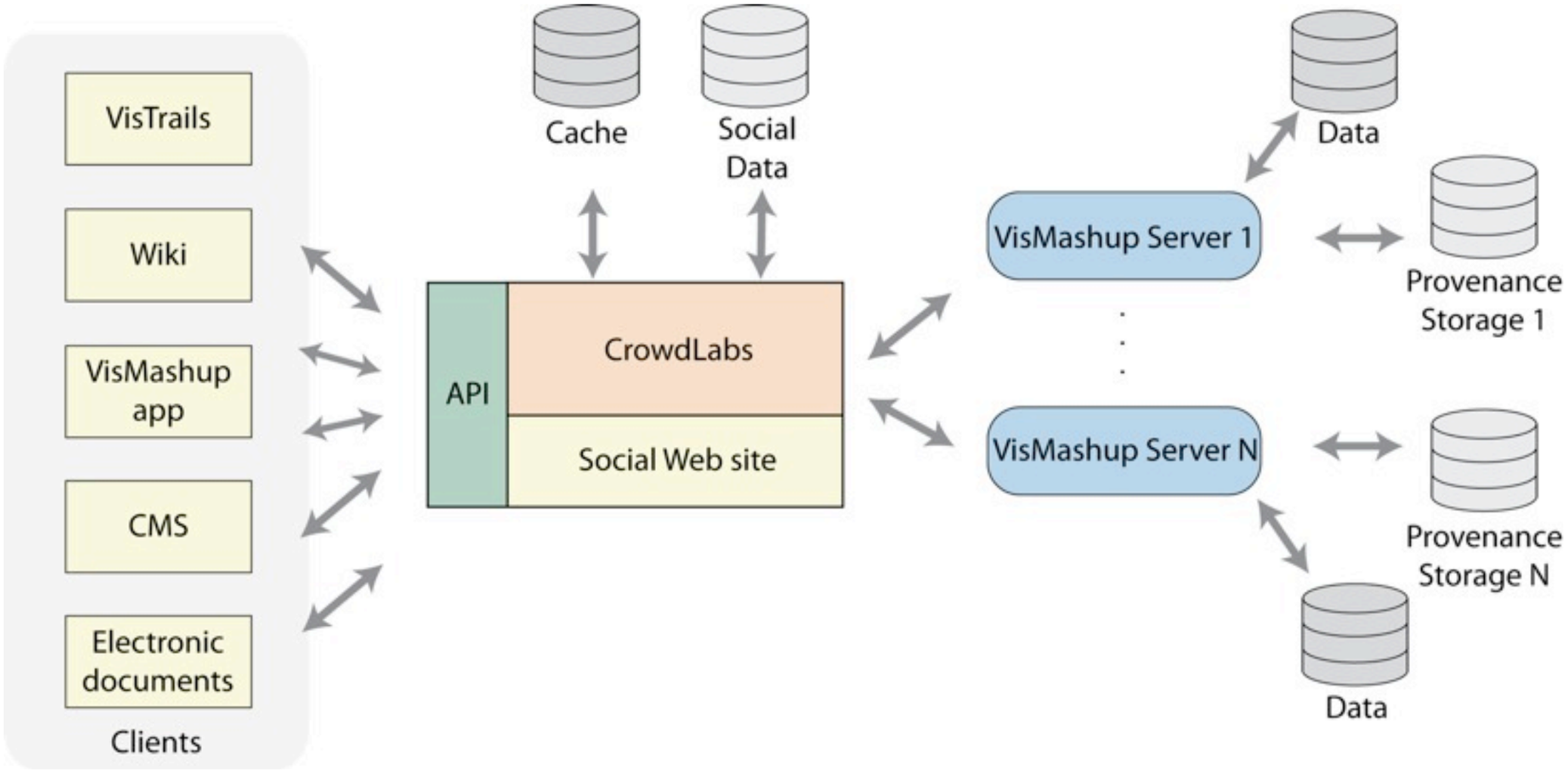
CrowdLabs System Configuration





Multiple configurations are possible, depending on available hardware

CrowdLabs System Configuration



Users can connect results published in an article or wiki page to the pipelines and data served by CrowdLabs

Outline

- Motivation
  - Scientific Research Scenario
- Framework
  - VisMashup
  - CrowdLabs
  - **Provenance Rich Publications**
- Demonstration
- Summary and Conclusions
- Future Work

Reproducible Documents
Other Approaches

Reproducible Documents
Other Approaches

Schwab et al., 2000

Reproducible Documents
Other Approaches

WaveLab and Reproducible Research

Jonathan B. Buckheit and David L. Donoho

Stanford University, Stanford CA 94305, USA

Abstract

WaveLab is a library of Matlab routines for wavelet analysis, wavelet-packet analysis, cosine-packet analysis and matching pursuit. The library is available free of charge over the Internet. Versions are provided for Macintosh, UNIX and Windows machines.

WaveLab makes available, in one package, all the code to reproduce all the figures in our published wavelet articles. The interested reader can inspect the source code to see exactly what algorithms were used, how parameters were set in producing our figures, and can then modify the source to produce variations on our results. WaveLab has been developed, in part, because of exhortations by Jon Claerbout of Stanford that computational scientists should engage in "really reproducible" research.

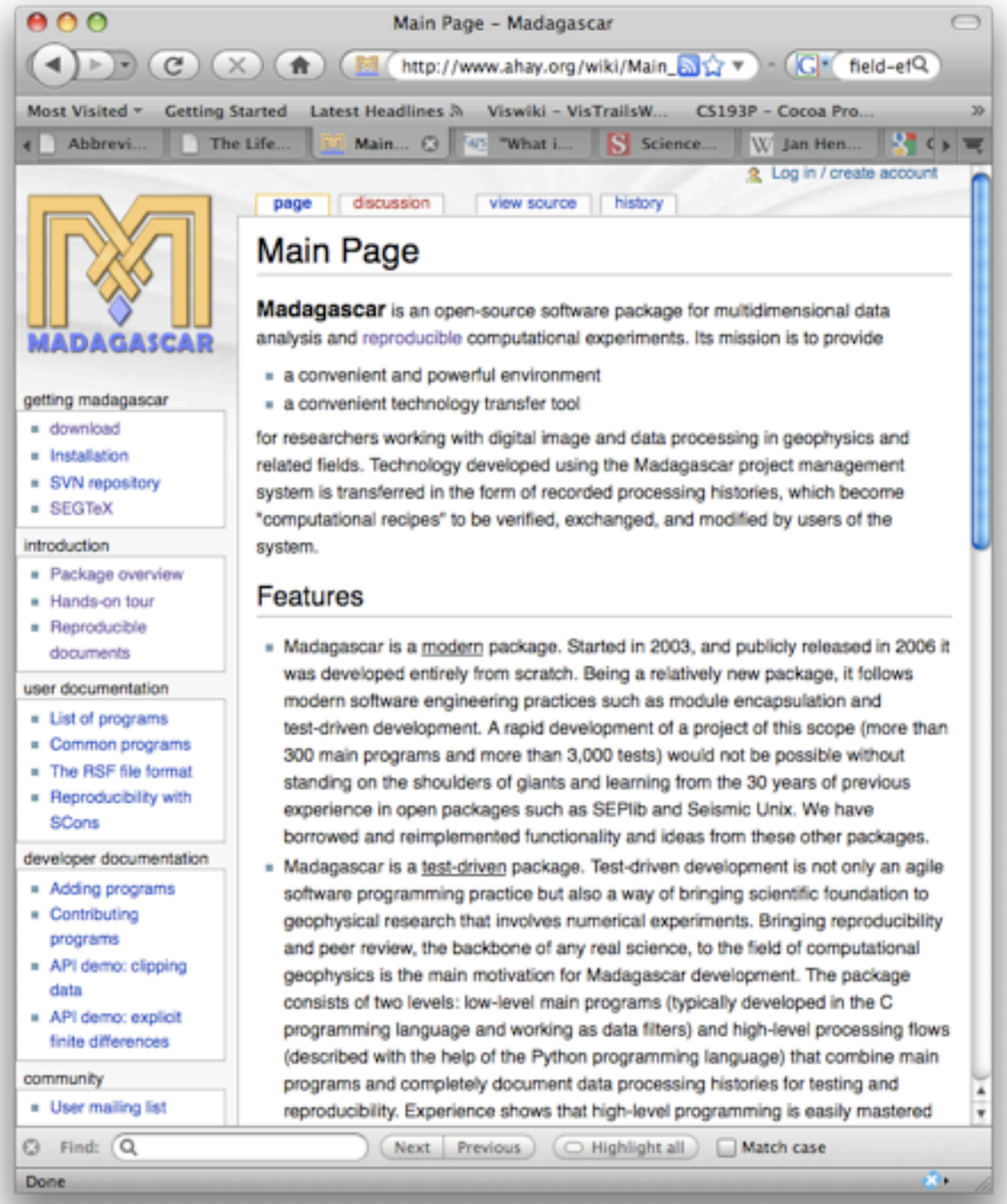
1 WaveLab – Reproducible Research via the Internet

A remarkable aspect of "the wavelet community" is the wide span of intellectual activities that it makes contact with. At one extreme, wavelets are interesting to mathematicians who are interested in functional spaces, their properties and decompositions – while at the other extreme wavelets are interesting in certain commercial software development efforts, where engineers craft computer programs applying wavelets to specific problems in high technology.

Work at Stanford on statistical applications of wavelets has, over the last five years, reflected a great deal of this range of wavelet activity. Dave Donoho and Iain Johnstone have written a number of theoretical papers; but also, a team involving Donoho, Johnstone, students Jon Buckheit, Shaobing Chen and Eric Kolaczyk, as well as Jeff Scargle of NASA Ames, have developed a collection of software tools known as WaveLab.

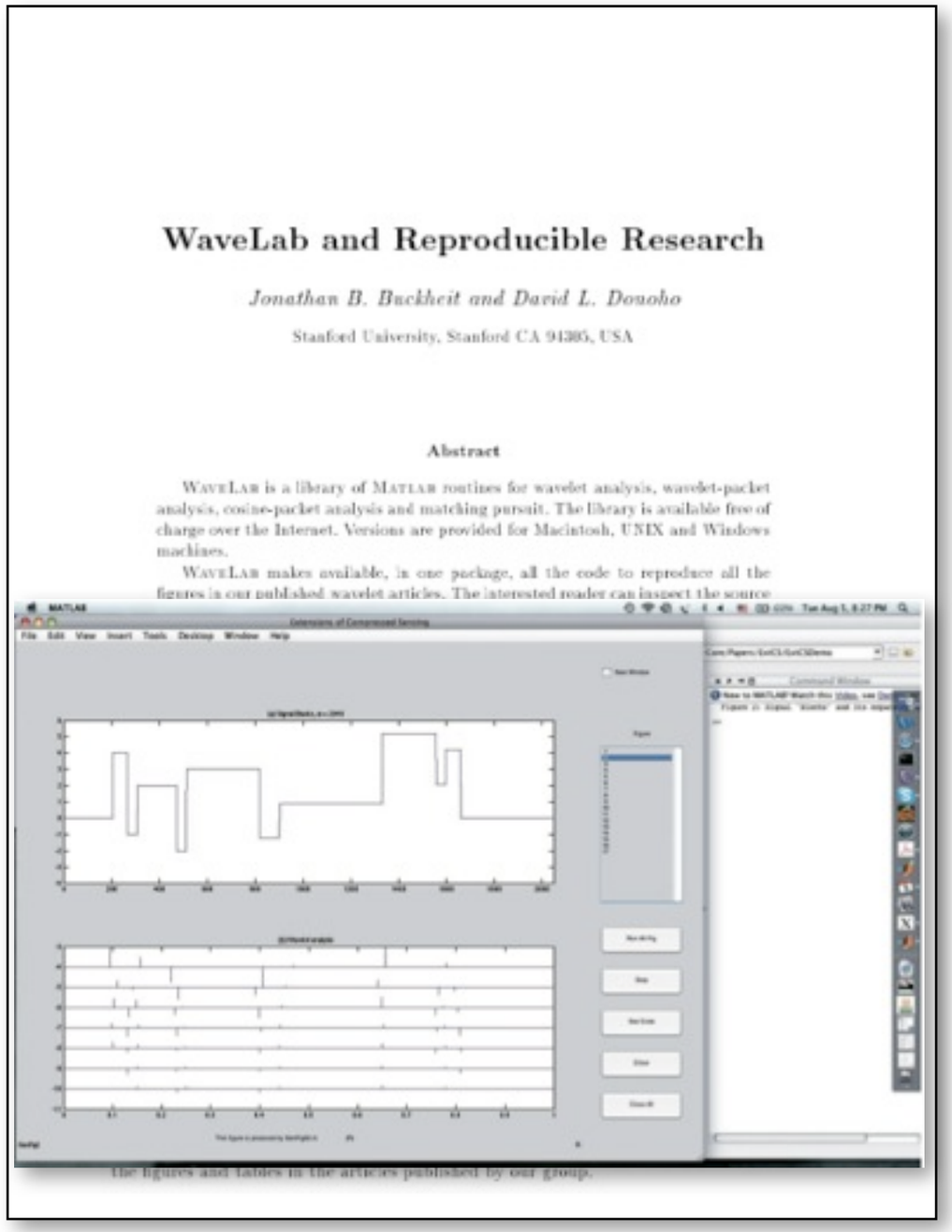
The WaveLab package contains a wide range of tools for wavelet and related time-frequency transforms. As this was written, version .700 was almost complete, consisting of over 700 files – programs, data, documentation and scripts. At the moment, the package requires over two megabytes of storage in compressed form. The package is available free of charge over the Internet, using standard interfaces like FTP and WWW. The stated goal of the package, and the stated reason for its distribution, is to allow others to reproduce the figures and tables in the articles published by our group.

[Bucheit and Donoho, 1995]



Schwab et al., 2000

Reproducible Documents
Other Approaches



WaveLab and Reproducible Research

Jonathan B. Buckheit and David L. Donoho

Stanford University, Stanford CA 94305, USA

Abstract

WaveLab is a library of Matlab routines for wavelet analysis, wavelet-packet analysis, cosine-packet analysis and matching pursuit. The library is available free of charge over the Internet. Versions are provided for Macintosh, UNIX and Windows machines.

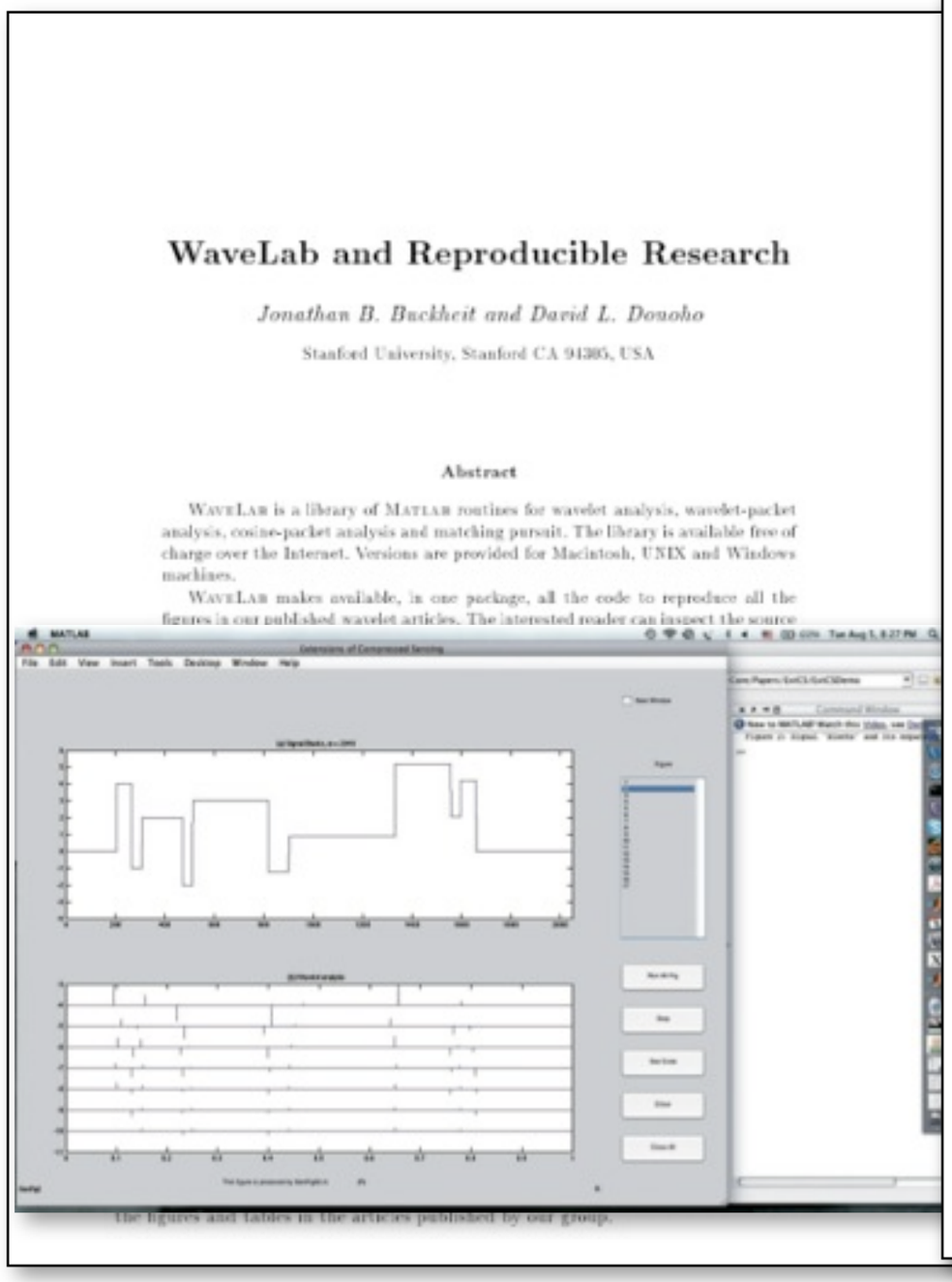
WaveLab makes available, in one package, all the code to reproduce all the figures in our published wavelet articles. The interested reader can inspect the source

the figures and tables in the articles published by our group.

[Bucheit and Donoho, 1995]

Schwab et al., 2000

Reproducible Documents
Other Approaches



WaveLab and Reproducible Research

Jonathan B. Buckheit and David L. Donoho

Stanford University, Stanford CA 94305, USA

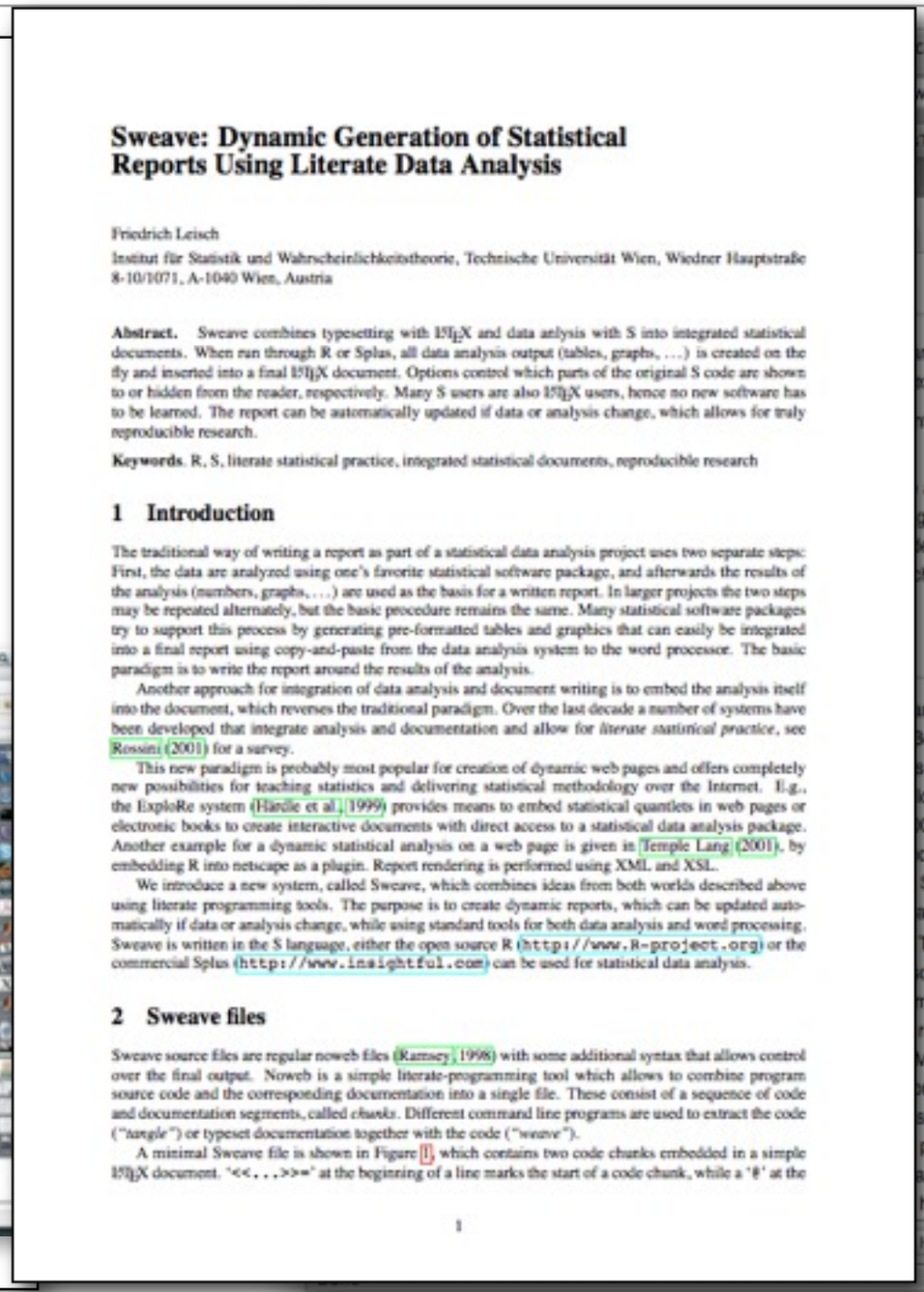
Abstract

WaveLab is a library of Matlab routines for wavelet analysis, wavelet-packet analysis, cosine-packet analysis and matching pursuit. The library is available free of charge over the Internet. Versions are provided for Macintosh, UNIX and Windows machines.

WaveLab makes available, in one package, all the code to reproduce all the figures in our published wavelet articles. The interested reader can inspect the source

The figures and tables in the articles published by our group.

[Bucheit and Donoho, 1995]



Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis

Friedrich Leisch

Institut für Statistik und Wahrscheinlichkeitstheorie, Technische Universität Wien, Wiedner Hauptstraße 8-10/1071, A-1040 Wien, Austria

Abstract. Sweave combines typesetting with LaTeX and data anlysis with S into integrated statistical documents. When run through R or Splus, all data analysis output (tables, graphs, ...) is created on the fly and inserted into a final LaTeX document. Options control which parts of the original S code are shown to or hidden from the reader, respectively. Many S users are also LaTeX users, hence no new software has to be learned. The report can be automatically updated if data or analysis change, which allows for truly reproducible research.

Keywords. R, S, literate statistical practice, integrated statistical documents, reproducible research

1 Introduction

The traditional way of writing a report as part of a statistical data analysis project uses two separate steps: First, the data are analyzed using one's favorite statistical software package, and afterwards the results of the analysis (numbers, graphs, ...) are used as the basis for a written report. In larger projects the two steps may be repeated alternately, but the basic procedure remains the same. Many statistical software packages try to support this process by generating pre-formatted tables and graphics that can easily be integrated into a final report using copy-and-paste from the data analysis system to the word processor. The basic paradigm is to write the report around the results of the analysis.

Another approach for integration of data analysis and document writing is to embed the analysis itself into the document, which reverses the traditional paradigm. Over the last decade a number of systems have been developed that integrate analysis and documentation and allow for *literate statistical practice*, see Rossini (2001) for a survey.

This new paradigm is probably most popular for creation of dynamic web pages and offers completely new possibilities for teaching statistics and delivering statistical methodology over the Internet. E.g., the ExploRe system (Härdle et al., 1999) provides means to embed statistical quantlets in web pages or electronic books to create interactive documents with direct access to a statistical data analysis package. Another example for a dynamic statistical analysis on a web page is given in Temple Lang (2001), by embedding R into netscape as a plugin. Report rendering is performed using XML and XSL.

We introduce a new system, called Sweave, which combines ideas from both worlds described above using literate programming tools. The purpose is to create dynamic reports, which can be updated automatically if data or analysis change, while using standard tools for both data analysis and word processing. Sweave is written in the S language, either the open source R (`http://www.R-project.org`) or the commercial Splus (`http://www.insightful.com`) can be used for statistical data analysis.

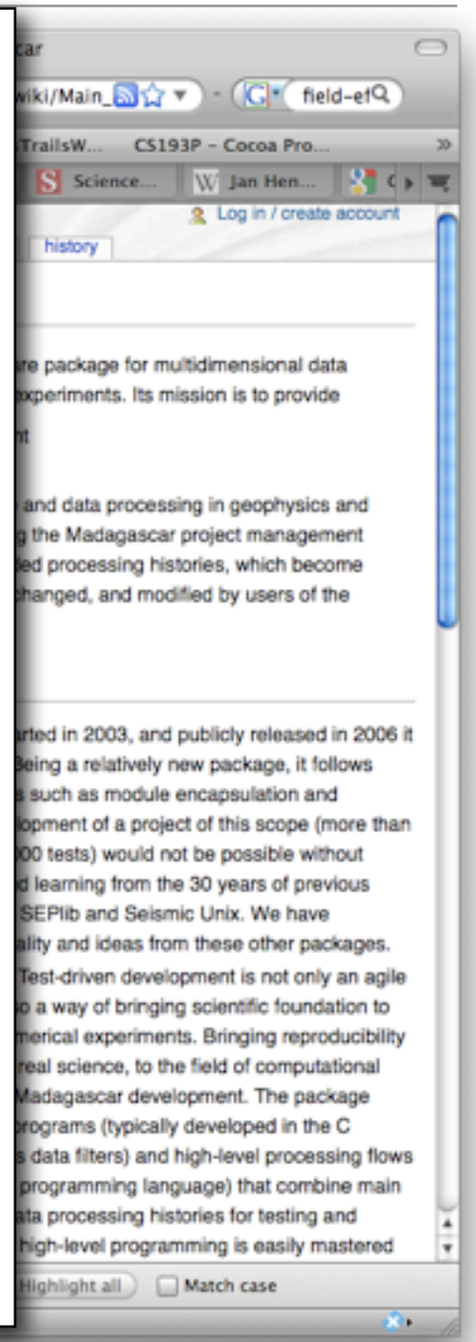
2 Sweave files

Sweave source files are regular noweb files (Ramsey, 1998) with some additional syntax that allows control over the final output. Noweb is a simple literate-programming tool which allows to combine program source code and the corresponding documentation into a single file. These consist of a sequence of code and documentation segments, called *chunks*. Different command line programs are used to extract the code ("*tangle*") or typeset documentation together with the code ("*weave*").

A minimal Sweave file is shown in Figure 1, which contains two code chunks embedded in a simple LaTeX document. '`<<...>>=`' at the beginning of a line marks the start of a code chunk, while a '`@`' at the

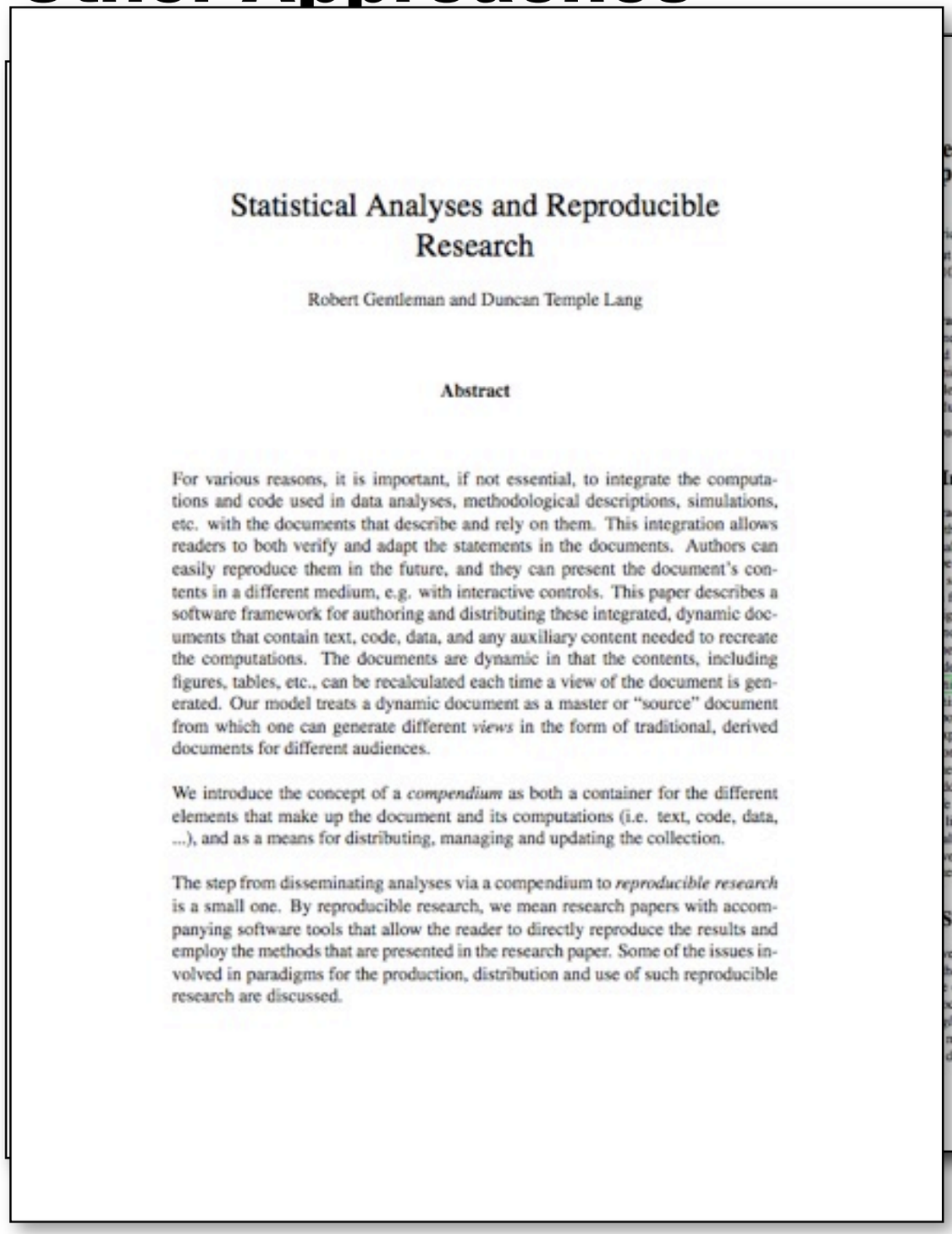
1

Leisch, 2002



Schwab et al., 2000

Reproducible Documents
Other Approaches



Statistical Analyses and Reproducible Research

Robert Gentleman and Duncan Temple Lang

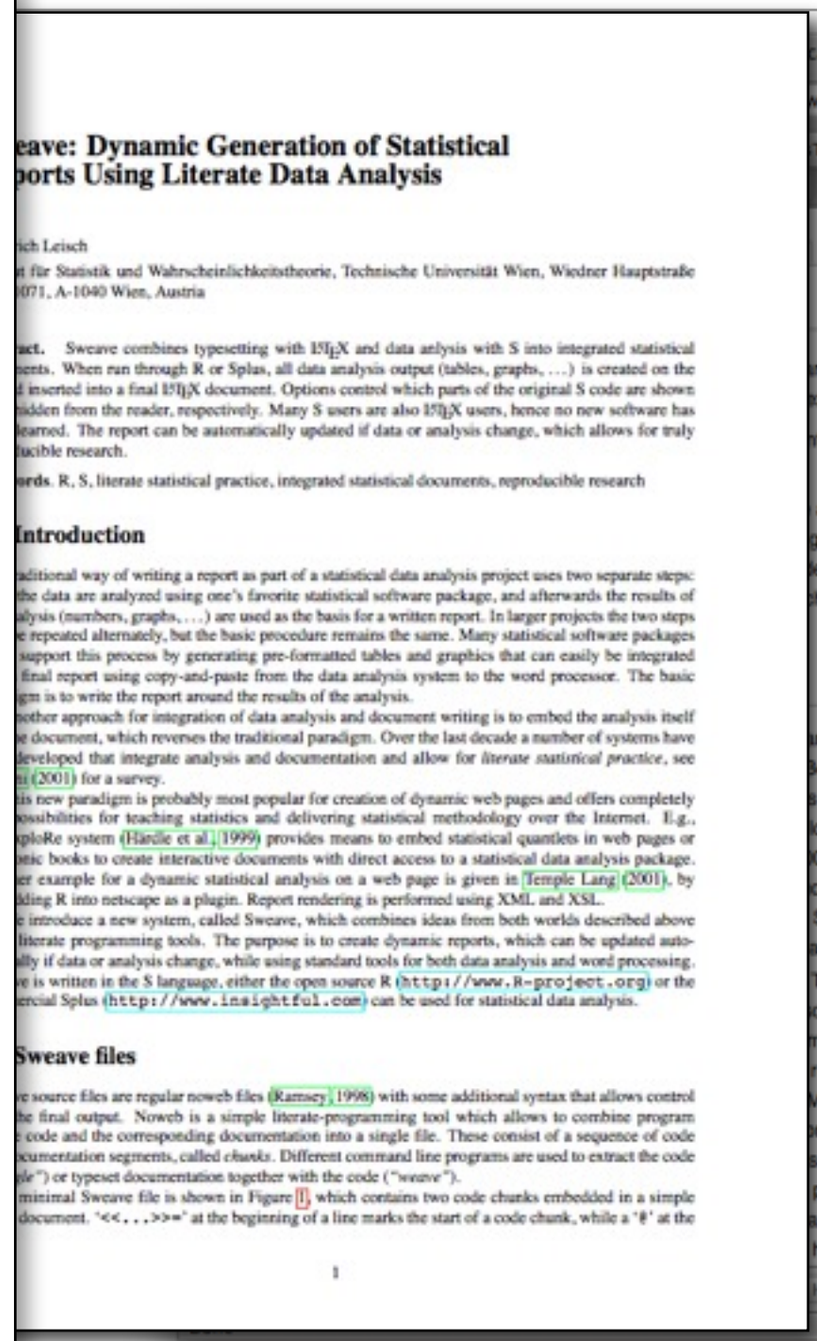
Abstract

For various reasons, it is important, if not essential, to integrate the computations and code used in data analyses, methodological descriptions, simulations, etc. with the documents that describe and rely on them. This integration allows readers to both verify and adapt the statements in the documents. Authors can easily reproduce them in the future, and they can present the document's contents in a different medium, e.g. with interactive controls. This paper describes a software framework for authoring and distributing these integrated, dynamic documents that contain text, code, data, and any auxiliary content needed to recreate the computations. The documents are dynamic in that the contents, including figures, tables, etc., can be recalculated each time a view of the document is generated. Our model treats a dynamic document as a master or "source" document from which one can generate different *views* in the form of traditional, derived documents for different audiences.

We introduce the concept of a *compendium* as both a container for the different elements that make up the document and its computations (i.e. text, code, data, ...), and as a means for distributing, managing and updating the collection.

The step from disseminating analyses via a compendium to *reproducible research* is a small one. By reproducible research, we mean research papers with accompanying software tools that allow the reader to directly reproduce the results and employ the methods that are presented in the research paper. Some of the issues involved in paradigms for the production, distribution and use of such reproducible research are discussed.

[Gentleman and Lang, 2004]



eave: Dynamic Generation of Statistical
orts Using Literate Data Analysis

ich Leisch
t für Statistik und Wahrscheinlichkeitstheorie, Technische Universität Wien, Wiedner Hauptstraße
071, A-1040 Wien, Austria

act. Sweave combines typesetting with LaTeX and data anlysis with S into integrated statistical
ents. When run through R or Splus, all data analysis output (tables, graphs, ...) is created on the
inserted into a final LaTeX document. Options control which parts of the original S code are shown
idden from the reader, respectively. Many S users are also LaTeX users, hence no new software has
earned. The report can be automatically updated if data or analysis change, which allows for truly
ucible research.

ords. R, S, literate statistical practice, integrated statistical documents, reproducible research

Introduction

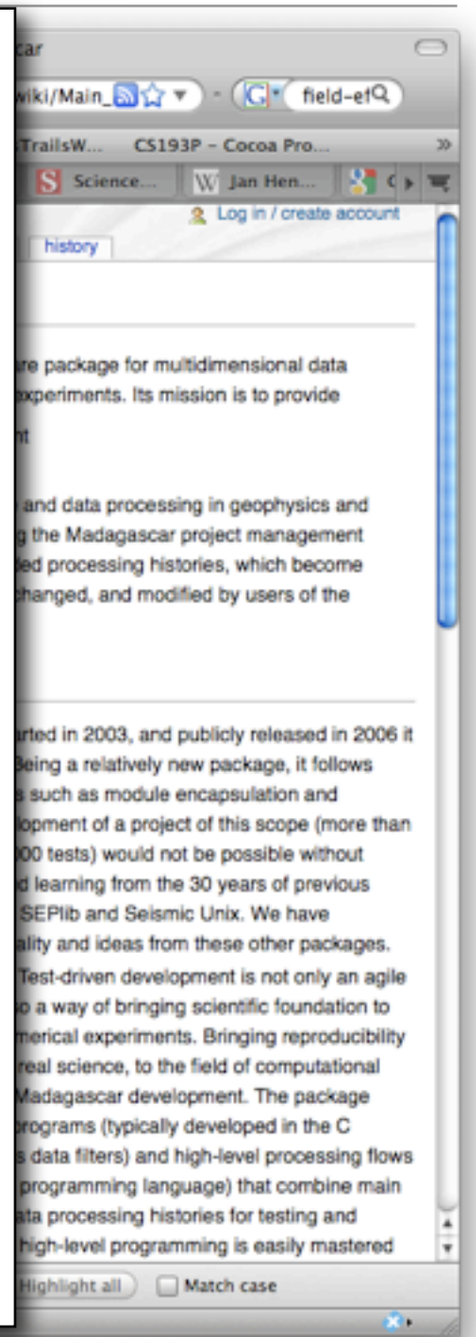
aditional way of writing a report as part of a statistical data analysis project uses two separate steps:
he data are analyzed using one's favorite statistical software package, and afterwards the results of
alysis (numbers, graphs, ...) are used as the basis for a written report. In larger projects the two steps
e repeated alternately, but the basic procedure remains the same. Many statistical software packages
support this process by generating pre-formatted tables and graphics that can easily be integrated
final report using copy-and-paste from the data analysis system to the word processor. The basic
gm is to write the report around the results of the analysis.
nother approach for integration of data analysis and document writing is to embed the analysis itself
e document, which reverses the traditional paradigm. Over the last decade a number of systems have
developed that integrate analysis and documentation and allow for *literate statistical practice*, see
(2001) for a survey.
is new paradigm is probably most popular for creation of dynamic web pages and offers completely
ossibilities for teaching statistics and delivering statistical methodology over the Internet. E.g.,
ploRe system (Härdle et al., 1999) provides means to embed statistical quantlets in web pages or
nic books to create interactive documents with direct access to a statistical data analysis package.
er example for a dynamic statistical analysis on a web page is given in Temple Lang (2001), by
ding R into netscape as a plugin. Report rendering is performed using XML and XSL.
e introduce a new system, called Sweave, which combines ideas from both worlds described above
literate programming tools. The purpose is to create dynamic reports, which can be updated auto-
ally if data or analysis change, while using standard tools for both data analysis and word processing.
e is written in the S language, either the open source R (`http://www.R-project.org`) or the
ercial Splus (`http://www.insightful.com`) can be used for statistical data analysis.

Sweave files

e source files are regular noweb files (Ramsey, 1998) with some additional syntax that allows control
he final output. Noweb is a simple literate-programming tool which allows to combine program
code and the corresponding documentation into a single file. These consist of a sequence of code
cumentation segments, called *chunks*. Different command line programs are used to extract the code
le") or typeset documentation together with the code ("*weave*").
minimal Sweave file is shown in Figure 1, which contains two code chunks embedded in a simple
document. '`<<...>>=`' at the beginning of a line marks the start of a code chunk, while a '`@`' at the

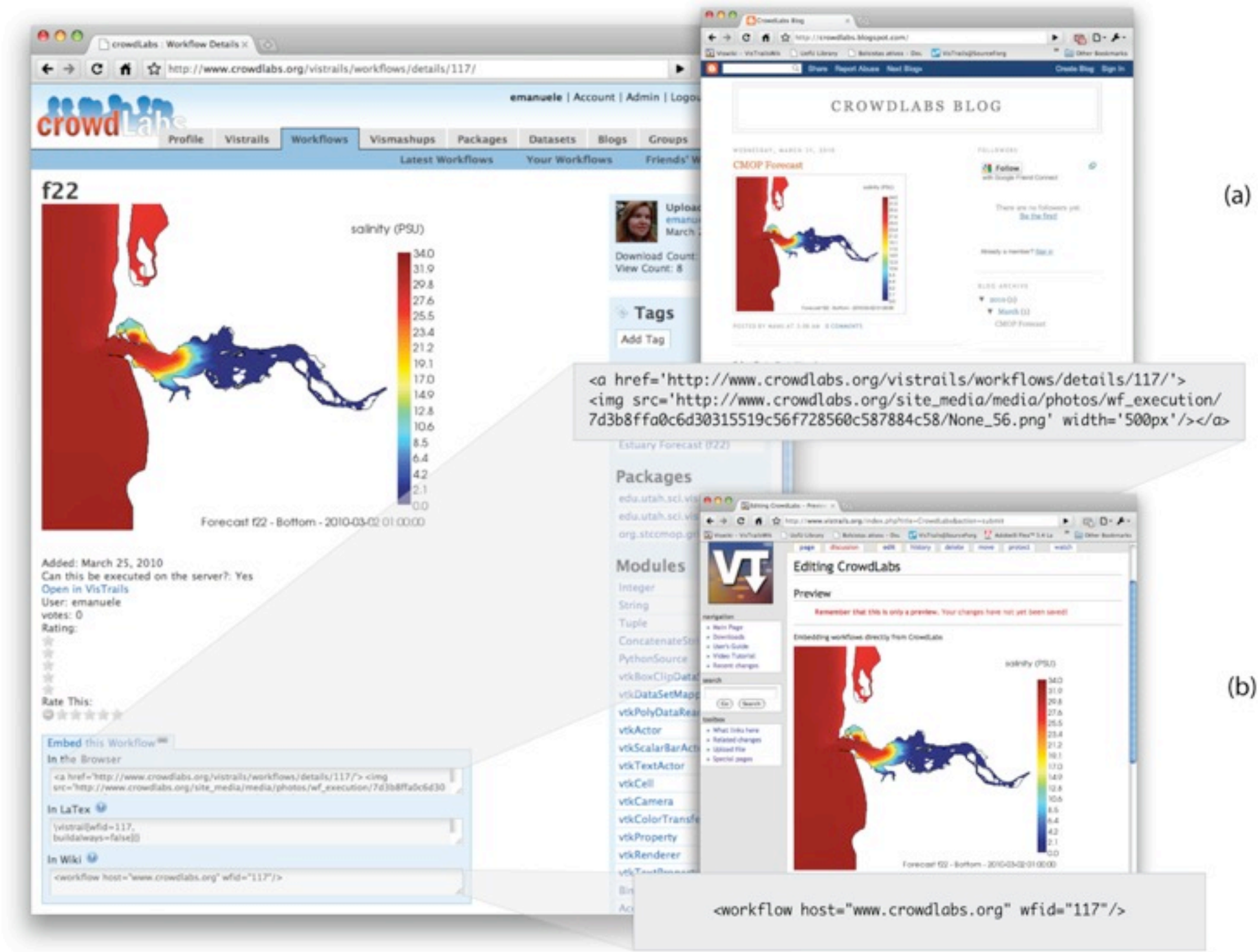
1

Leisch, 2002



Schwab et al., 2000

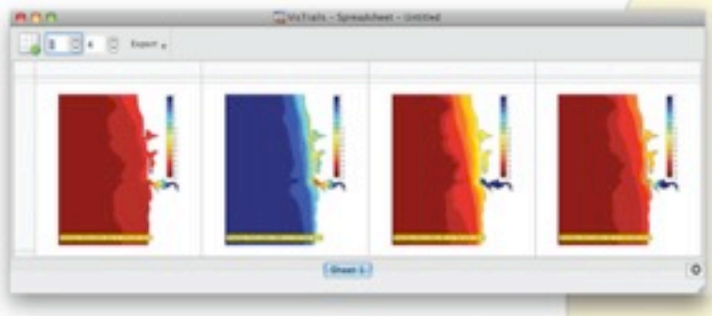
Provenance-Rich Publications



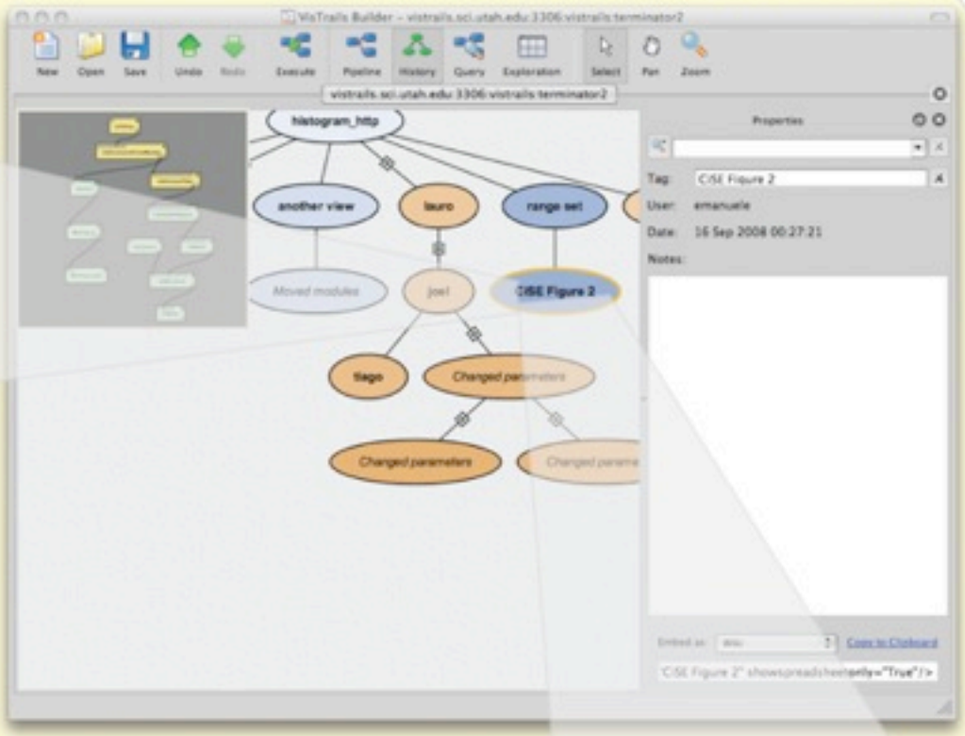
(a)

(b)

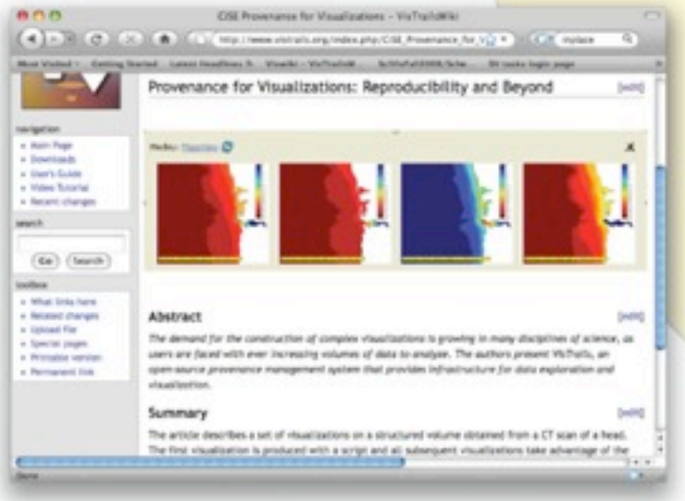
Provenance-Rich Publications



Interactive visualization



Workflow evolution



wiki page

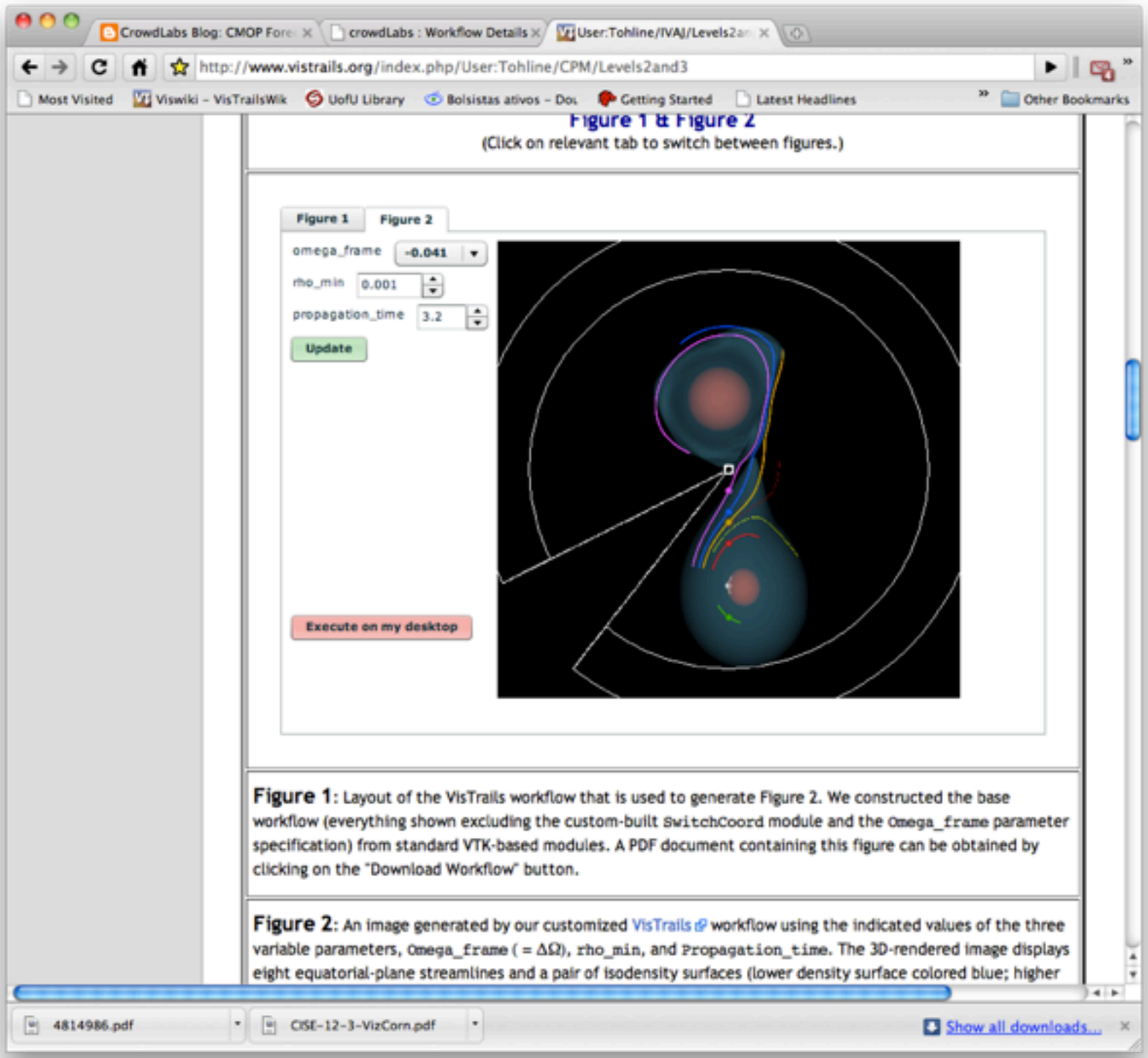
pdf document

powerpoint presentation

Scientist

(joint work with Huy Vo, Juliana Freire and Cláudio Silva)

The routine of an Astrophysicist
Provenance-Rich Documents



(joint work with Joel Tohline, CiSE 12(3), 2010)

Demonstration

Crowdlabs
Provenance-Rich Document
VisMashup

Our users

Open Wildland Fire Modeling E community
http://www.openwfm.org/wiki/index.php5/Main_Page
Most Visited
Viswiki - VisTrailsWiki
Latest Headlines (2199)
Bolsistas at...- Doutorado
vnc://sage
WFM
Open Wildland Fire Modeling E community
page
discussion
view source
history
Log in
Main Page
Welcome to the Open Wildland Fire Modeling E community Wiki
How to get an account
We encourage participation on this wiki by members WFM Community members. Content on this wiki is public to view, however an account is required to edit or add content. To get an account, please contact the site webmaster webmaster[at]sci.utah.edu. Include your full name and your affiliations within the WFM Community. Please also suggest a desired username for your account. We will set it up for you and email your login info as soon as possible.
Getting Started
Help Editing
Welcome to CMOP | Center for Coastal Margin Observation & Prediction (CMOP)
http://www.stccmop.org/
cmop
HOME
ABOUT CMOP
NEWS
EDUCATION
RESEARCH
KNOWLEDGE TRANSFER
DATA
CONTACT US
CMOP
Center for Coastal Margin Observation & Prediction
Transforming Ocean Exploration
QUICK LINKS
Research
Seminars
Blogs
Directory
Map & Directions
Employment
Site Map
EVENTS
« October 2009 »
Welcome to the Center for Coastal Margin Observation & Prediction
Hypoxia
Quinault Indian Nation gains better understanding of hypoxia in ocean with the help of Phoebe
Read More
OUR VISION
SEARCH CM
RESEARCH
SATURN
Lean more
last modified on 18 September 2009, at 23:13.
This page has been accessed 142 times.
Privacy policy
About
Open Wildland Fire Modeling E community
Disclaimers
The ALPS project
Algorithms and Libraries for Physics Simulations

Summary and Conclusions

- This is a new framework for simplifying the creation and deployment of collaborative data analysis and visualization tools
- VisMashup:
  - Simplifies the creation of custom visualization applications
  - Developers can quickly assemble custom applications, leveraging an existing collection of visualization pipelines and their provenance
  - Generated apps are flexible and simple enough for end-users
- CrowdLabs:
  - Facilitates collaboration and sharing among scientists and visualization experts
  - Scientists can create and publish new data products with minimal intervention of expert programmers and IT personnel

Summary and Conclusions

- Provenance-Rich Publications
  - Users can create documents whose digital artifacts (e.g., figures) include a deep caption: detailed provenance information which contains the specification of the computational process (or workflow) and associated parameters used to produce the artifact
  - Supports the publication on wikis, other Web sites, or scientific documents

Future Work

- Evaluate effectiveness and usability of VisMashup and CrowdLabs
- Use the provenance captured by VisMashup to improve GUI generation
- Use distributed model for workflow execution that depends on the client configuration (for portable devices)
- Advanced Provenance Analytics
  - Workflow search engine
  - Recommendations based on the types of file the user wants to visualize
  - The ability to connect people based on the types of workflows they are interested in, etc.
- Improve Web-enabled interfaces and graphics (with 3D support)
- Add streaming and multi-resolution techniques for supporting remote data

Acknowledgements

Acknowledgements

Publication List
(Related to the Dissertation)

Journals

- VisMashup: Streamlining the Creation of Custom Visualization Applications. E. Santos, L. Lins, J. Ahrens, J. Freire and C. Silva. IEEE Trans. Vis. Comp. Graph (Proceedings of IEEE Vis 2009).
- Visualizing a Journal that Serves the Computational Sciences Community, J. Tohline and E. Santos. Computing in Science & Engineering (CiSE), Visualization Corner, vol. 12, no. 3, 2010.

Conferences

- Using Workflow Medleys to Streamline Exploratory Tasks. E. Santos, D. Koop, H. Vo, E. Anderson, J. Freire and C. Silva. In Proceedings of SSDBM 2009, vol. 5566/2009, 2009, pp. 292-301.

Workshops

- Information Sharing in Science 2.0: Challenges and Opportunities. E. Santos, J. Freire, and C. Silva. CHI 2009 Workshop on The Changing Face of Digital Science: Workshop on New Practices in Scientific Collaborations, 2009.
- Towards Supporting Collaborative Data Analysis and Visualization in a Coastal Margin Observatory. E. Santos, P. Mates, E. Anderson, B. Grimm, J. Freire and C. Silva, CSCW 2010 Workshop on The Changing Dynamics of Scientific Collaboration, 2010.

Submitted/In Manuscript

- Interacting with Scientific Applications Using Portable Devices. E. Santos, W. Silva, P. Hendricks, J. Freire and C. Silva.
- CrowdLabs: Supporting Collaborative Data Analysis and Visualization. E. Santos, P. Mates, J. Freire and C. Silva.

Publication List
(Others)

- Bridging Workflow and Data Provenance using Strong Links, D. Koop, E. Santos, B. Bauer, M. Troyer, J. Freire and C. Silva, In proceedings of the 22nd International Conference on Scientific and Statistical Database Management (SSDBM), 2010, (to appear).
- Collaborative Monitoring and Analysis for Simulation Scientists, R. Tchoua, S. Klasky, N. Podhorszki, B. Grimm, A. Khan, E. Santos, C. Silva, P. Mouallem, M. Vouk. In Proceedings of The 2010 International Symposium on Collaborative Technologies and Systems (CTS 2010), pp. 235 - 244, 2010.
- Using VisTrails and Provenance for Teaching Scientific Visualization, C. Silva, E. Anderson, E. Santos and J. Freire, In Proceedings of Eurographics 2010 Education program, 2010.
- Enabling Advanced Visualization Tools in a Web-Based Simulation Monitoring System, E. Santos, J. Tierny, A. Khan, B. Grimm, L. Lins, J. Freire, V. Pascucci, C. Silva, S. Klasky, R. Barreto, N. Podhorszki, In Proceedings of 5th IEEE International Conference on e-Science, pp.358-365, 2009.
- End-to-End eScience: Integrating Workflow, Query, Visualization, and Provenance at an Ocean Observatory, B. Howe, P. Lawson, R. Bellinger, E. Anderson, E. Santos, J. Freire, C. Scheidegger, A. Baptista, C. Silva. In Proceedings of 4th IEEE International Conference on e-Science, pp. 127-134, 2008.
- Provenance for Computational Tasks: A Survey, J. Freire, D. Koop, E. Santos, C. Silva. Computing in Science & Engineering (CiSE) Special Issue on Computational Provenance, vol. 10, no. 3, pp. 11–21, 2008.
- A First Study on Clustering Collections of Workflow Graphs, E. Santos, L. Lins, J. Ahrens, J. Freire, C. Silva. In Proceedings of Second International Provenance and Annotation Workshop (IPAW '08), vol. 5272/2008, pp. 160–173, 2008.
- The First Provenance Challenge, L. Moreau, B. Ludscher, I. Altintas, R. Barga, S. Bowers, S. Callahan, G. Chin Jr., B. Clifford, S. Cohen, S. Cohen-Boulakia, S. Davidson, E. Deelman, L. Digiampietri, I. Foster, J. Freire, J. Frew, J. Futrelle, T. Gibson, Y. Gil, C. Goble, J. Golbeck, P. Groth, D. A. Holland, S. Jiang, J. Kim, D. Koop, A. Krenek, T. McPhillips, G. Mehta, S. Miles, D. Metzger, S. Munroe, J. Myers, B. Plale, N. Podhorszki, V. Ratnakar, E. Santos, C. Scheidegger, K. Schuchardt, M. Seltzer, Y. L. Simmhan, C. Silva, P. Slaughter, E. Stephan, R. Stevens, D. Turi, H. Vo, M. Wilde, J. Zhao, and Y. Zhao. Concurrency and Computation: Practice and Experience, vol. 20, no. 5, pp. 409 – 418, 2008.
- Tackling the Provenance Challenge One Layer at a Time, C. Scheidegger, D. Koop, E. Santos, H. Vo, S. Callahan, J. Freire, C. Silva. Concurrency and Computation: Practice and Experience, vol. 20, no. 5, 2008, pp. 473–483.
- Examining Statistics of Workflow Evolution Provenance: A First Study, L. Lins, D. Koop, E. Anderson, S. Callahan, E. Santos, C. Scheidegger, J. Freire, C. Silva. In Proceedings of SSDBM 2008, vol. 5069/2008, pp. 573–579, 2008.
- VisTrails: Visualization Meets Data Management, S. Callahan, J. Freire, E. Santos, C. Scheidegger, C.T. Silva and H.T. Vo. In Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 745-747, 2006.
- Managing Rapidly-Evolving Scientific Workflows, J. Freire, C.T. Silva, S. Callahan, E. Santos, C. Scheidegger and H.T. Vo. In Proceedings of International Provenance and Annotation Workshop (IPAW), 2006.
- Managing the Evolution of Dataflows with VisTrails, S. Callahan, J. Freire, E. Santos, C. Scheidegger, C.T. Silva and H.T. Vo. In Proceedings of IEEE Workshop on Workflow and Data Flow for Scientific Applications (SciFlow), 2006.

Thank you!

Questions?