Embedding CIPRES Science Gateway Capabilities in Phylogenetics Software Environments

Mark A. Miller
San Diego Supercomputer Center
Phylogenetics is the study of the diversification of life on the planet Earth, both past and present, and the relationships among living things through time.
Phylogenetic relationships are inferred by comparing characteristics of living organisms, and grouping them according to shared traits.
The Cladisticules

Species 1  Species 2  Species 3
Species 4  Species 5  Species 6
Species 7  Species 8
The Cladisticules

Fused head/thorax

Species 1
Species 2
Species 3
Species 4
Species 5
Species 6
Species 7
Species 8
The Cladisticules

Separate head/thorax

Species 1
Species 2
Species 3
Species 4
Species 5
Species 6
Species 7
Species 8
The Cladisticules

Species 1
Species 2
Species 3
Species 4
Species 5
Species 6
Species 7
Species 8

“Head Gear”
The Cladisticules

Species 1
Species 2
Species 3
Species 4
Species 5
Species 6
Species 7
Species 8

Antennae
The Cladisticules

- Species 1
- Species 2
- Species 3
- Species 4
- Species 5
- Species 6
- Species 7
- Species 8

Horns
Score traits, create a matrix

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Species 2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Species 3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Species 4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Species 5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Species 6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Species 7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Species 8</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Group according to character traits

Sp. 1

- Sp. 1
- Sp. 5
  - Sp. 2
  - Sp. 8
- Sp. 3
- Sp. 4
- Sp. 7
- Sp. 6
Now, algorithmically, we want to search for the “best” tree, the one that gives us the most satisfactory explanation of the data.
Evolutionary relationships can be inferred from DNA sequence comparisons:

1. Align sequences to determine evolutionary equivalence:

<table>
<thead>
<tr>
<th>Species</th>
<th>DNA Sequence</th>
<th>DNA Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species 1</td>
<td>TCAAGATTAAAG -CATTTACTAAATT -ACAAAGTTTCTTC -TTAGCTATGA -AACCATAGGTCA -TCA</td>
<td>Species 2</td>
</tr>
<tr>
<td>Species 3</td>
<td>TCAAGATTAAAG -CATTTACTAAATT -ACAAAGTTTCTTC -TTAGCTATGA -AACCATAGGTCA -TCA</td>
<td>Species 4</td>
</tr>
<tr>
<td>Species 5</td>
<td>TCAAGATTAAAG -CATTTACTAAATT -ACAAAGTTTCTTC -TTAGCTATGA -AACCATAGGTCA -TCA</td>
<td>Species 6</td>
</tr>
<tr>
<td>Species 7</td>
<td>TCAAGATTAAAG -CATTTACTAAATT -ACAAAGTTTCTTC -TTAGCTATGA -AACCATAGGTCA -TCA</td>
<td>Species 8</td>
</tr>
<tr>
<td>Species 9</td>
<td>TCAAGATTAAAG -CATTTACTAAATT -ACAAAGTTTCTTC -TTAGCTATGA -AACCATAGGTCA -TCA</td>
<td>Species 10</td>
</tr>
<tr>
<td>Species 11</td>
<td>TCAAGATTAAAG -CATTTACTAAATT -ACAAAGTTTCTTC -TTAGCTATGA -AACCATAGGTCA -TCA</td>
<td>Species 12</td>
</tr>
<tr>
<td>Species 13</td>
<td>TCAAGATTAAAG -CATTTACTAAATT -ACAAAGTTTCTTC -TTAGCTATGA -AACCATAGGTCA -TCA</td>
<td>Species 14</td>
</tr>
<tr>
<td>Species 15</td>
<td>TCAAGATTAAAG -CATTTACTAAATT -ACAAAGTTTCTTC -TTAGCTATGA -AACCATAGGTCA -TCA</td>
<td>Species 16</td>
</tr>
<tr>
<td>Species 17</td>
<td>TCAAGATTAAAG -CATTTACTAAATT -ACAAAGTTTCTTC -TTAGCTATGA -AACCATAGGTCA -TCA</td>
<td>Species 18</td>
</tr>
<tr>
<td>Species 19</td>
<td>TCAAGATTAAAG -CATTTACTAAATT -ACAAAGTTTCTTC -TTAGCTATGA -AACCATAGGTCA -TCA</td>
<td>Species 20</td>
</tr>
<tr>
<td>Species 21</td>
<td>TCAAGATTAAAG -CATTTACTAAATT -ACAAAGTTTCTTC -TTAGCTATGA -AACCATAGGTCA -TCA</td>
<td>Species 22</td>
</tr>
<tr>
<td>Species 23</td>
<td>TCAAGATTAAAG -CATTTACTAAATT -ACAAAGTTTCTTC -TTAGCTATGA -AACCATAGGTCA -TCA</td>
<td>Species 24</td>
</tr>
</tbody>
</table>

2. Infer evolutionary relationships based on some set of assumptions:

[Diagram showing evolutionary relationships between different species of whales.]
Sequence alignment algorithms determine which nucleotides in each species are most probably “evolutionarily equivalent.”

<table>
<thead>
<tr>
<th></th>
<th>263</th>
<th>*</th>
<th>260</th>
<th>*</th>
<th>300</th>
<th>*</th>
<th>320</th>
</tr>
</thead>
<tbody>
<tr>
<td>species 1</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCAAAGT--ACAAGCGGCTATTAA--AG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 2</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAATCCCTCTCTAGG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 3</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ATAGTCATTACATTAA--AG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 4</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 5</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 6</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 7</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 8</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 9</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 10</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 11</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 12</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 13</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 14</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 15</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 16</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 17</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 18</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 19</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 20</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 21</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 22</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>species 23</td>
<td>TCAAAAGATTAAAGC-CAT3CATGTCCTAGTG--ACAGGCGA--ATCT--AAG--GGA--AACC3C3AGTGGCTCAATTAA--TCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
We can all agree on that legs, heads, etc. are evolutionarily equivalent.
We can all agree on that legs, heads, etc. are evolutionarily equivalent.

Sequence alignment shows us which sequence letters are evolutionarily equivalent.
Tree inference algorithms look for the best tree based on some set of assumptions about the evolutionary process:
Inferring Evolutionary relationships from DNA sequence comparisons is powerful:

DNA sequences are determined by fully automated procedures.

Sequence data can be gathered from many species at scales from gene to whole genome.

The high speed and low cost of NexGen Sequencing means new levels of sensitivity and resolution can be obtained.

The speed of sequencing is still increasing, while the cost of sequencing is decreasing.
Inferring Evolutionary relationships from DNA sequence comparisons is powerful, **BUT:**

Current analyses often involve 1000’s of species and/or 1000’s of characters, creating very large matrices.

Even with heuristics, Sequence alignment and Tree inference algorithms are computationally intensive, so computational power often limits the analyses (already).

The run times for tree search analysis scales exponentially with number of taxa and number of characters for codes in current use.

There are at least $10^7$ species, each with 3000 - 30,000 genes, so the need for computational power and new approaches will continue to grow.
Inferring Evolutionary relationships from DNA sequence comparisons is powerful, **BUT:**

Current analyses often involve 1000’s of species and/or 1000’s of characters, creating very large matrices.

Even with heuristics, Sequence alignment and Tree inference algorithms are computationally intensive, so computational power often limits the analyses (already).

The run times for tree search analysis scales exponentially with number of taxa and number of characters for codes in current use.

There are at least $10^7$ species, each with 3000 - 30,000 genes, so the need for computational power and new approaches will continue to grow.
Inferring Evolutionary relationships from DNA sequence comparisons is powerful, BUT:

Current analyses often involve 1000’s of species and/or 1000’s of characters, creating very large matrices.

Even with heuristics, Sequence alignment and Tree inference algorithms are computationally intensive, so computational power often limits the analyses (already).

The run times for tree search analysis scale exponentially with number of taxa and number of characters for codes in current use.

There are at least $10^7$ species, each with 3000 - 30,000 genes, so the need for computational power and new approaches will continue to grow.
Inferring Evolutionary relationships from DNA sequence comparisons is powerful, **BUT:**

Current analyses **often involve 1000’s of species and/or 1000’s of characters**, creating very large matrices.

Even with heuristics, **Sequence alignment and Tree inference algorithms are computationally intensive**, so computational power often limits the analyses (already).

The run times for tree search analysis **scale exponentially with number of taxa and number of characters** for codes in current use.

There are at least $10^7$ species, each with 3000 - 30,000 genes, so the need for computational power and new approaches will continue to grow.
Inferring Evolutionary relationships from DNA sequence comparisons is powerful, **BUT:**

Current analyses often involve 1000’s of species and/or 1000’s of characters, creating very large matrices.

Even with heuristics, Sequence alignment and Tree inference algorithms are computationally intensive, so computational power often limits the analyses (already).

The run times for tree search analysis scale exponentially with number of taxa and number of characters for codes in current use.

There are at least $10^7$ species, each with 3000 - 30,000 genes, so the need for computational power and new approaches will continue to grow.
Biology in the new world of abundant DNA sequence data requires a new kind of cyberinfrastructure!

- Phylogenetics codes that were historically run in desktop environments must be moved to high performance computing resources.

- The need for access to HPC resources will increase for the foreseeable future.

- Scientists who do not have HPC access will have to tailor their questions to available resources, and risk being left out of the discovery process.
Step 1. Democratizing access
The CIPRES Science Gateway was designed to allow users to analyze large sequence data sets using community codes on significant computational resources.

The CSG provides

- Login-protected personal user space for storing results indefinitely.
- Access to most/all native command line options for several codes.
- Support for adding new codes and upgrading to new versions as needed.
Step 1. Democratizing access
The CIPRES Science Gateway was designed to allow users to analyze large sequence data sets using community codes on significant computational resources.

The CSG provides

- Login-protected personal user space for storing results indefinitely.
- Access to most/all native command line options for several codes.
- Support for adding new codes and upgrading to new versions as needed.
Step 1. Democratizing access
The CIPRES Science Gateway was designed to allow users to analyze large sequence data sets using community codes on significant computational resources.

The CSG provides

- Login-protected personal user space for storing results indefinitely.
- Access to most/all native command line options for several codes.
- Support for adding new codes and upgrading to new versions as needed.
Step 1. Democratizing access
The CIPRES Science Gateway was designed to allow users to analyze large sequence data sets using community codes on significant computational resources.

The CSG provides

- Login-protected personal user space for storing results indefinitely.
- Access to most/all native command line options for several codes.
- Support for adding new codes and upgrading to new versions as needed.
The Science Gateway Program provides scalable, sustainable resources

XSEDE

Parallel codes

TSCC

Serial codes

Workbench Framework

Web Interface

XSEDE

Serial codes

Web Interface
The Science Gateway Program provides scalable, sustainable resources

Awarded by competitive allocation

- XSEDE
- TSCC

Parallel codes

Workbench Framework

Web Interface

Serial codes
The Science Gateway Program provides scalable, sustainable resources

- **XSEDE**
  - Parallel codes
- **TSCC**
  - Serial codes

**Workbench Framework**

**Web Interface**

Fee-for-service at SDSC

- SDSC

- NSF

- XSEDE
Workflow for the CIPRES Gateway:

Assemble Sequences

Upload to Portal

Store

Download

Run Alignment

Run Tree Inference

Post-Tree Analysis
Take away message: CIPRES success is unrelated to its interface....
Our app is relatively simple, and has been driven by community requirements alone….

“Developers may address new research topics in the course of gateway design in order to further their academic goals. Resulting gateways may be more complex than necessary, less reliable, and may not meet the goals of the domain science community for whom they were designed.

Focus group participants noted that sometimes simple tools are all that is needed to enable cutting edge science, but [Gateway developers] ‘make the easy things hard.’”

Submissions and SU* usage are increasing linearly.

29,000 more SU*s requested each month.

Projected use for 2013 - 2014 is 20 million SU*s

*1 SU = 1 core hour at unit priority
Usage of the CIPRES Science Gateway Dec 2009 – July 2013

12 more users submit 160 more jobs each month

Growth in usage is driven by new users
The CIPRES use case is different from the typical XSEDE resource request:

- Most tree inference codes scale to no more than 64 cores.
- 20% of CSG users are students in classes, so queue time matters.
- 88% of CSG jobs complete within 12 hours, so queue time matters.
- 3% of CSG jobs run for more than 1 week and most codes have no restart capability, so run times of up to 334 hours are required.
- These jobs are not a good fit for the intent of the large XSEDE machines.
Important Policy Moment:

Based (in part) on our use case, the US NSF created the Trestles cluster to provide “On demand” computing (Thanks, NSF!):

- Trestles is managed and allocated to keep queue depth near zero
- Administrators allow CSG to run jobs for 334 hours
- The machine is significant in size, but small jobs (64 cores or less) are welcomed
Impact on Science:

Publications enabled by the CIPRES Science Gateway/CIPRES Portal:

<table>
<thead>
<tr>
<th>Year</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013*</td>
<td>191</td>
</tr>
<tr>
<td>2012</td>
<td>229</td>
</tr>
<tr>
<td>2011</td>
<td>143</td>
</tr>
<tr>
<td>2010</td>
<td>92</td>
</tr>
<tr>
<td>2009</td>
<td>60</td>
</tr>
<tr>
<td>2008</td>
<td>4</td>
</tr>
</tbody>
</table>

*As of September 1, 2013

Publications in the pipeline:

<table>
<thead>
<tr>
<th>Status</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>In preparation</td>
<td>91</td>
</tr>
<tr>
<td>In review</td>
<td>25</td>
</tr>
</tbody>
</table>
Impact on Science:

- In Q2 2013, 29% of all XSEDE users who ran jobs ran them from the CSG
- 50% of users said they had no access to local resources, nor funds to purchase access on cloud computing resources
- Used for curriculum delivery by at least 68 instructors.
- Jobs run for researchers in 23/29 EPSCOR states.
- Routine submissions from Harvard, Berkeley, Stanford and from non-PhD granting institutions
- Jobs submitted from 6 continents; 50% US, 32% Europe; 11% South America; 4% Asia; 3% Australia; > 1% Africa
Step 2: If a little access makes science go faster, can we do even better?
Workflow for the CIPRES Gateway:

1. Assemble Sequences
2. Upload to Portal
3. Run Alignment
4. Run Tree Inference
5. Store
6. Download
7. Post-Tree Analysis
There are highly-evolved desktop/browser applications that help with matrix assembly, but have no tree inference tools or are under powered:

- Morphobank
- Mesquite
- RaxmlGUI
- Vipr
- Influenza Research Database
There are projects that offer powerful and distinct user experiences, and are interested in incorporating powerful tree inference tools into an existing application:
Many advanced developers find the workflow supported by the CIPRES browser too restrictive.
A Public CIPRES RESTful API (CRA) will help these use cases
Use Cases: Mesquite and REST Services

Mesquite provides powerful visual tools for pre- and post tree tasks on the desktop……
Use Cases: Mesquite and REST Services

Mesquite

Sequence Assembly | Sequence Editing | Tree Display | Tree Reconciliation | Tree Editing | Tree Analysis

Desktop

But its tree inference is limited by the desktop hardware……
Use Cases: Mesquite and REST Services

RESTful CIPRES API can provide the needed compute power without leaving the app......
Use Cases: MorphoBank and REST Services

MorphoBank provides powerful visual tools for creating and sharing data matrices among large teams……
Use Cases: MorphoBank and REST Services

- Character Visualization
- Character Quantification
- Character Recording
- Team Data Sharing
- Character Matrix Assembly
- Character Matrix Publication

But it has no concept of trees or tree inference……
Use Cases: MorphoBank and REST Services

CIPRES RESTful API will allow users to proceed with their workflow within the MorphoBank environment……
Use Cases: Individual developers and REST Services

Advanced phylogenetic researchers want:

• to run many jobs simultaneously
• create ad hoc workflows

Advanced phylogenetic researchers don’t want:

• to assemble and click each job one at a time
• to manually port the output of one job to the subsequent job in their workflow
Use Cases: Individual developers and REST Services

Assuming modest scripting skills, an advanced researcher can accomplish this goal using the CIPRES RESTful API to avoid the clumsy browser interface.
OK, the use cases seem appealing, even compelling.

How to go about implementing this?
Design changes for implementing RESTful services:
The CSG Web Application (WA) provides browser access. It is based on Java Struts2.
The Workbench Framework (WF) provides backend functions
The WF deploys generic “tasks”....
Servlets
JSP
Struts

....and queries
generic DBs
Specific information is coded in a Central Registry
User information, data, and job runs are stored in a MySQL database.
Tasks and queries are sent to remote machines and DBs
The CRA replaces the Presentation Layer with a simple web server.
It uses the same WF package.
Design Challenges

The CRA will provide access by an open group of developers (of unknown number and skill level) with tools to access significant computational resources.
There are several immediate requirements for providing this kind of access:

- The interface between “outside” developers and the CRA software must be versatile and simple.
- Changes in phylogenetic codes accessed by the CRA must be easy to propagate to client applications.
- As responsibility for the end-user interface is shifted from the CIPRES development group to outside developers, error management is key.
- Resources must be protected from unintentional (and intentional) abuse.
There are several immediate requirements for providing this kind of access:

• The interface between “outside” developers and the CRA software must be versatile and simple.

• Changes in phylogenetic codes accessed by the CRA must be easy to propagate to client applications.

• As responsibility for the end-user interface is shifted from the CIPRES development group to outside developers, error management is key.

• Resources must be protected from unintentional (and intentional) abuse.
There are several immediate requirements for providing this kind of access:

• The interface between “outside” developers and the CRA software must be versatile and simple.

• Changes in phylogenetic codes accessed by the CRA must be easy to propagate to client applications.

• As responsibility for the end-user interface is shifted from the CIPRES development group to outside developers, error management is key.

• Resources must be protected from unintentional (and intentional) abuse.
There are several immediate requirements for providing this kind of access:

• The interface between “outside” developers and the CRA software must be versatile and simple.

• Changes in phylogenetic codes accessed by the CRA must be easy to propagate to client applications.

• As responsibility for the end-user interface is shifted from the CIPRES development group to outside developers, error management is key.

• Resources must be protected from unintentional (and intentional) abuse.
How the current application manages job submissions:

1. User configures, WA submits form
2. WA generates browser form; Javascript controls
3. Submitted form populates parameter map in WF
4. WF generates scheduler files, does JAVA backend checking of parameter map

Submit Job

Code XML Documents
Submitted form populates parameter map in WF

WF Generates scheduler files, does JAVA backend checking of parameter map

Submit Job

How will the CRA manage job submissions?
How will the CRA manage job submissions?

?  

REST Client must populate the parameter map, BUT

No automatically generated forms

No control over submissions

Submitted form populates parameter map in WF

WF Generates scheduler files, does JAVA backend checking of parameter map

Submit Job

Code XML Documents
GOAL: Create code that allows clients to generate a form from CodeXML

- REST client submits form
- Submitted form populates parameter map in WF
- WF generates scheduler files, does JAVA backend checking of parameter map
- Code XML Documents

- REST Client generates GUI from Code XML
- Submit Job
GOAL: Create code that allows clients to generate a form from CodeXML

- REST client submits form
- Submitted form populates parameter map in WF
- REST Client generates GUI from Code XML
- WF Generates scheduler files, does JAVA backend checking of parameter map
- Submit Job

Requires participation by the REST client developer
GOAL: Create code that allows clients to generate a form from CodeXML

REST client submits form

Submitted form populates parameter map in WF

WF Generates scheduler files, does JAVA backend checking of parameter map

REST Client generates GUI from Code XML

Submit Job

Automating this means new changes to Code XML can be rolled out quickly
GOAL: Provide robust “backend” input checking

REST client submits form

Submitted form populates parameter map in WF

WF Generates scheduler files, does JAVA backend checking of parameter map

Submit Job

Code XML Documents
GOAL: Provide robust backend checking

- generate error checking code from the tool XML document
- reject submissions that violate constraints in the tool xml file
- input file format checking/transformation
- return an informative numeric and human readable error message

Submitted form populates parameter map in WF

WF Generates scheduler files, does JAVA backend checking of parameter map

Submit Job

Code XML Documents
How the current application reports job status/completion:

1. WF moves results to CSG DB
2. WA posts links to results, notification of completion
3. WF notifies WA
4. WF sends e-mail to user
5. Completed Job
6. WF moves results to CSG DB
How will the CRA report job status/completion?

- REST client submits form
- WF sends e-mail to user
- WF notifies WA
- WF moves results to CSG DB
- Completed Job

WF notifies WA?

WF moves results to CSG DB
How the current application reports job status/completion:

Client application:

- Specifies how their application should be notified of job completion or job status change via a set of submission parameters.

- provides either an email address, a callback URL, both or neither.

- will be allowed to poll the callback urn up to a specified frequency.
Methods for Access to CRA:

Scripter/Developer: via Registered Application

End User: via Registered Desktop Application

via Registered Web Application
Registration of Client Applications:

Only registered applications can submit jobs.

Applications will be reviewed and approved by a CIPRES staff member.

Developer receives an application key to include in all CRA requests.

The key will be used to monitor (and if necessary, throttle) use of the CRA from all client applications.
Registration of End Users:

User registers with Web App.

Registered Web Application (stores user info)

Application provides key and User info

CRA

User enters credentials*

Registered Desktop application (stores user info)

*User must register once
Registration of End Users:

User registers with Web App.

Web App Users

Registered Web Application (stores user info)

Application provides key and User info

CRA

User enters credentials*

Desktop App Users

Registered Desktop application (stores user info)

Application Provides Key and User credentials

*User must register once

Per-User accounting information is required by XSEDE
With a RESTful API, a script can be used to deploy thousands of jobs.

Additional controls that will be implemented:

• Limit of $x$ jobs submitted by a single application

• Limit of $y$ jobs sent to the queue simultaneously by user

• Place “reserves” on each user’s account by debiting projects use by job in progress the user account.

• Track and disable submissions from any client application that is highly problematic.

• Provide a testbed for client application and script developers.
The 907,180 kg gorilla in the room.

Will we be able to control usage sufficiently? Is providing programmatic access to these kinds of resources crazy?
Expected Release mid-2014

Stay Tuned....
Acknowledgements:

CIPRES Science Gateway
Terri Schwartz
Bryan Lunt
Paul Hoover
Wayne Pfeiffer

XSEDE Implementation Support
Nancy Wilkins-Diehr
Doru Marcusiu
Leo Carson

Workbench Framework:
Terri Schwartz
Paul Hoover
Lucie Chan
Jeremy Carver