A RESTful API for Access to Phylogenetic Tools via the CIPRES Science Gateway

Mark A. Miller[1], Terri Schwartz[1], Brett E. Pickett[2,*], Sherry He[3], Edward B. Klem[3], Richard H. Scheuermann[2], Maria Passarotti[4], Seth Kaufman[4] and Maureen A. O'Leary[5]

[1]San Diego Supercomputer Center, University of California, San Diego, La Jolla, CA, USA. [2]J. Craig Venter Institute, La Jolla, CA, USA. [3]Northrop Grumman Health Solutions, Rockville, MD, USA. [4]Whirl-i-gig, Brooklyn, NY, USA. [5]Department of Anatomical Sciences, Stony Brook University, Stony Brook, NY, USA. *Current address: Thomson Reuters Intellectual Property and Science; Carlsbad, CA, USA.

ABSTRACT: The CIPRES Science Gateway is a community web application that provides public access to a set of parallel tree inference and multiple sequence alignment codes run on large computational resources. These resources are made available at no charge to users by the NSF Extreme Science and Engineering Discovery Environment (XSEDE) project. Here we describe the CIPRES RESTful application programmer interface (CRA), a web service that provides programmatic access to all resources and services currently offered by the CIPRES Science Gateway. Software developers can use the CRA to extend their web or desktop applications to include the ability to run MrBayes, BEAST, RAxML, MAFFT, and other computationally intensive algorithms on XSEDE. The CRA also makes it possible for individuals with modest scripting skills to access the same tools from the command line using curl, or through any scripting language. This report describes the CRA and its use in three web applications (Influenza Research Database – www.fludb.org, Virus Pathogen Resource – www.viprbrc.org, and MorphoBank – www.morphobank.org). The CRA is freely accessible to registered users at https://cipresrest.sdsc.edu/cipresrest/v1; supporting documentation and registration tools are available at https://www.phylo.org/restusers.

KEYWORDS: CIPRES, RESTful API, MrBayes, BEAST, RAxML, MAFFT, jMODELTEST2, Science Gateway, phylogenetics, computational biology, web portal

CITATION: Miller et al. A RESTful API for Access to Phylogenetic Tools via the CIPRES Science Gateway. *Evolutionary Bioinformatics* 2015:11 43–48 doi: 10.4137/EBO.S21501.

RECEIVED: November 05, 2014. **RESUBMITTED:** January 19, 2015. **ACCEPTED FOR PUBLICATION:** January 24, 2015.

ACADEMIC EDITOR: Jike Cui, Associate Editor

TYPE: Software or Database Review

FUNDING: The work described here was supported by NSF DBI-1262628 and a generous award (TG-DEB090011) of computer time and development support from the XSEDE project (also sponsored by the U.S. National Science Foundation). MorphoBank development work was sponsored by NSF DEB−1208270. The IRD and ViPR development work was funded by NIAID contracts HHSN272200900041C and HHSN272201400028C. Initial software development for the workbench framework was also supported by NIH GM73931-01. The authors confirm that the funder had no influence over the study design, content of the article, or selection of this journal.

COMPETING INTERESTS: Authors disclose no potential conflicts of interest.

CORRESPONDENCE: mmiller@sdsc.edu

Introduction

The decreasing cost and increasing rate of DNA sequence acquisition, together with improved tools for phylogenetic analysis,[1,2] have made it possible to analyze evolutionary relationships in unprecedented detail. The massive amount of sequence data available, whether used alone or in concert with phenomic data,[3] offers great potential for new insights into the evolutionary history of life on earth. However, many of the best methods currently available for sequence alignment and phylogenetic tree inference are computationally intensive. This fact makes access to computational power an increasingly important factor in determining which analyses are tractable for individual researchers. Individuals who do not have access to large computer clusters, the knowledge of how to operate them, and/or the attendant resources to maintain them can be severely limited in the kinds of analyses they can perform.

The CIPRES Science Gateway (CIPRES)[4] and the University of Oslo Bioportal (http://www.bioportal.uio.no) were created as first-generation web portals to enable computationally demanding phylogenetic inference analyses for the broad scientific community. Both these web applications offered public access to parallel phylogenetic codes run on powerful computational resources through a browser interface. Both web applications provided users with the ability to upload a dataset, configure a job run using a particular community code (eg, MrBayes or RAxML), run that job efficiently on a multicore compute cluster, and retrieve the results. The need for this type of access is clear: over the past 4 years, phylogenetic analyses conducted using CIPRES and the Oslo Bioportal have been part of more than 1700 publications. Although the Oslo Bioportal ceased public operations in 2013, CIPRES continues to provide public access to large Linux clusters made available through the US National Science Foundation's Extreme Science and Engineering Discovery Environment (XSEDE) project, and its rate of usage continues to grow.

The access to computational resources provided by CIPRES can dramatically decrease the amount of time required to complete a phylogenetic analysis: the parallel codes available at CIPRES are 5–60-fold faster than comparable codes on a single-core desktop machine, and users can submit and run multiple jobs simultaneously. For example, an analysis involving 10 datasets, each of which requires 24 hours to analyze on a local desktop resource, will take at least 10 days to complete, but the same 10 datasets can be submitted to CIPRES simultaneously and the analysis will be completed in a few hours. For individuals with no alternative access to multicore compute resources, CIPRES represents a significant enabling technology.

To date, CIPRES has provided access to computational resources only via a web browser interface. In this mode, each job must be configured and submitted sequentially through a set of point-and-click operations. The results must then be manually downloaded into a local work environment and imported into additional software packages for further analysis. While browser-based access is clearly useful for many kinds of analyses (in 2014, CIPRES ran more than 160,000 jobs for 4600 users), it can also be cumbersome and restrictive. To make public access to HPC resources more flexible and dynamic, we created a new set of CIPRES web services so that CIPRES capabilities can be accessed by developers and individuals with basic scripting skills outside of a web browser interface. These web services are accessible through a public CIPRES RESTful application programmer interface (CRA). The CRA provides registered users with access to all services currently supported by the browser-based CIPRES web application (www.phylo.org/portal2).

The CRA was created to benefit projects and individual scientists in three specific situations. First, for users who find the current browser interface cumbersome, the CRA makes it possible to run analyses outside the confines of the browser interface. Any person with basic knowledge of a scripting language can create a simple script that will submit jobs to CIPRES resources and retrieve the results. For example, an individual who wishes to deploy 20 analyses using the current browser interface must do so by repeating a tedious set of point-and-click operations 20 times. Using the CRA, however, an individual with basic knowledge of a scripting language can construct an appropriate script and submit the same 20 jobs from their desktop with much less effort. Moreover, with appropriate scripting, it is possible to create a workflow that chains multiple analyses, with the output of one being submitted as input to the next. The current CIPRES interface does not support this type of chained workflow.

Second, a number of desktop software applications support sequence alignment and tree inference (eg, raxmlGUI,[5] siMBa,[6] and Mesquite[7]), but these applications can access only the computing power available on the desktop computer where they are installed. Desktop computing is often inadequate for modern sequence alignment and tree inference problems, and large datasets may cause the desktop resource to become unresponsive for long periods while the job completes. The CRA makes it possible for developers to incorporate the ability to submit jobs to the powerful CIPRES compute resources into desktop applications with only minor changes to the existing code. Once implemented, users of tools like raxmlGUI, Mesquite, etc can run analyses on CIPRES resources, and use their desktop resource for other activities. The result will be a dramatic increase in the size and number of datasets that can be analyzed conveniently using popular desktop software packages.

Finally, the CRA allows developers of web portals to incorporate compute-intensive sequence alignment and tree inference capabilities into their portal by adding a code that contacts the CRA, submits jobs, and retrieves results. The overhead for doing this is modest, and involves much less effort than creating and maintaining these capabilities themselves. CRA services have already been incorporated into three production web sites: the Influenza Research Database[8] (IRD; www.fludb.org), Virus Pathogen Resource[9] (ViPR; www.viprbrc.org), and MorphoBank[10,11] (www.morphobank.org). The use of CRA by these web sites is described further below.

In short, the CRA provides a new, flexible mechanism to run analyses on large compute clusters. Currently, parallel codes for BEAST (v1.8 and 2),[12,13] FastTree2,[14] GARLI,[15] jModelTest2,[16] MAFFT,[17] MrBayes,[18] PhyloBayes,[19] and RAxML[20] can be accessed through the CRA. It is important to note that while the CRA provides additional flexibility by allowing access outside the CIPRES browser interface, these services can only be accessed by third-party client software or scripts. The basic design of the CRA, information on registration, some practical examples, and future directions are provided below.

CRA Structure and Design

CIPRES was constructed from a software package that consists of two components: a software development kit called the Workbench Framework (WF), which performs the executive functions of the Gateway (eg, job submission and tracking, user authentication)[21] and a web application that creates the CIPRES browser interface (CBI). The CBI allows users to create jobs and download results through a web browser. The CRA uses the WF for its executive functions, but since no graphical user interface (GUI) is required for programmatic access, the CRA uses a much simpler web application to process calls from remote applications. The CRA was constructed using Jersey (https://jersey.java.net),[23] an extensible open-source JAX-RS (JSR 311) reference implementation for building RESTful Web services. JAX-RS is a natural fit for the Java/Struts2 platform used to construct the CBI/WF. Because the WF is shared by both the CBI and the CRA, the CRA provides programmatic access to all services available through the CBI. Improvements to the WF will benefit both the CRA and CBI applications.

Using the CRA

Use of the CRA requires some form of client application to access the services. Programmatic access must be initiated from a registered client application by a registered user. Client applications are typically developed by users outside the

CIPRES project group, and can be anything from a simple set of command line instructions to a complex Java web application. We have identified three specific use cases for the CRA, and these are described briefly below. Complete user documentation is provided at the CRA web site.

Command line access. Individuals with scripting skills can access the CRA from the command line via their preferred scripting language. A simple example of this type of access is via a bash shell script that uses the Linux curl or wget commands to submit jobs to the CRA and then retrieve the results. Scripting languages can also be used in combination with an HTTP library. For example, with Python, the Requests library can be used, and with PERL the LWP library can be used. With command line submissions, the user is responsible for the organization, storage, and persistence of all results produced. This use case expands the capabilities of individuals who are doing research where rapid, repetitive job submissions are required, a mode where the CIPRES browser interface can be unwieldy. Documentation and examples to support individuals who wish to access CRA via scripts are available on the CRA web site. Scripts that use CRA services must be registered at the CRA web site by their creators, and job submissions must include the CRA username and password, as well as the application ID provided by CIPRES when the script is registered.

Access through desktop applications. The CRA can also be accessed by phylogenetic applications that have a GUI and run on a user's desktop. There are a number of community-created applications that provide a GUI for a locally installed copy of command line multiple sequence alignment and tree inference programs. Some (eg, raxmlGUI,[5] and siMBa[6]) support only tree inference. There are other graphical desktop applications, such as Mesquite,[7] that support many forms of pre-tree data preparation and post-tree analysis, as well as alignment and tree inference. Some of these applications also provide data management, provenance, and persistence. Desktop applications like these can be adapted to make remote calls to CRA services for some or all of their sequence alignment and tree inference functions, enabling the application to handle more computationally intensive analysis than is available on the local resource where it is installed.

Desktop applications that use CRA services must also be registered at the CRA web site by their developers. On registration, application developers will receive a CRA application ID that must be included in distributions of their software. Each request that an application sends to the CRA must include the application ID, together with the CRA username and password of the person running the application. This use case expands the computing power available to applications in a desktop environment.

Access through web applications. A number of web applications currently provide registered users with access to phylogenetic tools. These applications give their users access to some of the same phylogenetic tree-building algorithms as the CRA (eg, MrBayes, PAUP, and RAxML) or to pre-tree data preparation tools (eg, MorphoBank) and provide the infrastructure to store, organize, share, persist, and integrate results. These applications can be integrated with the CRA to give users the ability to run much larger, more computationally intensive analysis on XSEDE resources. Users gain access to phylogenetic algorithms on HPC resources without having to leave a software environment where they are already comfortable and productive, and where they have access to other integrated data and metadata for downstream visualization and analysis of the resulting alignments and phylogenetic trees.

The developer of a web application such as this must register the application with the CRA and obtain an application ID, username, and password to include in its requests to the CRA. The application will authenticate to the CRA with this single username and password. Each job submission request that the application makes to the CRA will use custom request headers to identify the end user on whose behalf the submission is being made. Registered users of this type of web application do not need to register with the CRA. In fact, the use of the CRA can be as transparent to the users as the developer desires. This use case expands the capabilities of other web applications that benefit from access to powerful tree inference capabilities in their supported workflow.

Registration/Authentication. A key requirement of the CRA is that it must be possible to assign each submission to a specific, unique individual. Accordingly, individual users of scripted/command line and desktop clients must register for CRA access at the CRA web site. Any registered user may use *and* develop applications/scripts that use the CRA services.

Registered applications use either "DIRECT" or "UMBRELLA" authentication. Most applications will use DIRECT authentication. With DIRECT authentication, the application uses HTTP basic authentication over https to send the username and password of the person running the application, and jobs are submitted on behalf of this user only. DIRECT authentication is appropriate for individuals creating ad hoc scripts and for installed desktop applications, where each user has his or her own copy of the client script or application. On the other hand, web applications are often designed to submit jobs on behalf of a set of registered users (eg, ViPR,[9] see below), but it is disruptive to ask each user to provide additional (CRA) authentication for each submission. In this situation, UMBRELLA authentication is used: the application sends the username and password *of the application's administrator* in HTTP basic authentication headers, and each job submission includes information that identifies the application user who submitted the job (typically by providing an email address). The UMBRELLA mechanism allows CIPRES to map each submission from a web application to a particular individual, but uses a single CRA account for authentication. Because UMBRELLA authentication allows web applications to submit jobs for numerous users, CIPRES requires an additional vetting process before enabling job submissions from such applications. The goal of the vetting process is simply to establish trust in the UMBRELLA web application's user management strategy.

Restrictions/Controls. Although access to the CRA is provided at no cost to the user, the project has an established set of policies governing fair use of CRA resources. These are implemented as a series of controls enforced by the CRA application. Users at U.S. institutions are allowed to use up to 50,000 core hours of computational time from the CIPRES community allocation in each calendar year, and those at non-U.S. institutions are allowed up to 30,000 core hours. Individuals at institutions in the U.S. and those with collaborators in the U.S. are eligible to apply for additional computational time from the XSEDE project through a competitive allocation process. We are currently working to make it possible for users who do not have collaborators in the U.S. to purchase additional computational time at cost.

Submissions made by individual users are controlled so that users do not exceed their available allocation of compute time. New submissions by a given user will be blocked whenever the number of core hours that could potentially be consumed by their currently running jobs exceeds the amount of time remaining in their allocation. For example, a user with an allocation of 30,000 core hours would not be allowed to submit more than five jobs that are configured to run on 32 cores for a maximum of 168 hours, since six such jobs would consume more than their allocation limit (32,256 core hours) if they ran for the full 168 hours. However, if these runs each completed in 1 hour, the unused time becomes available to the user again immediately.

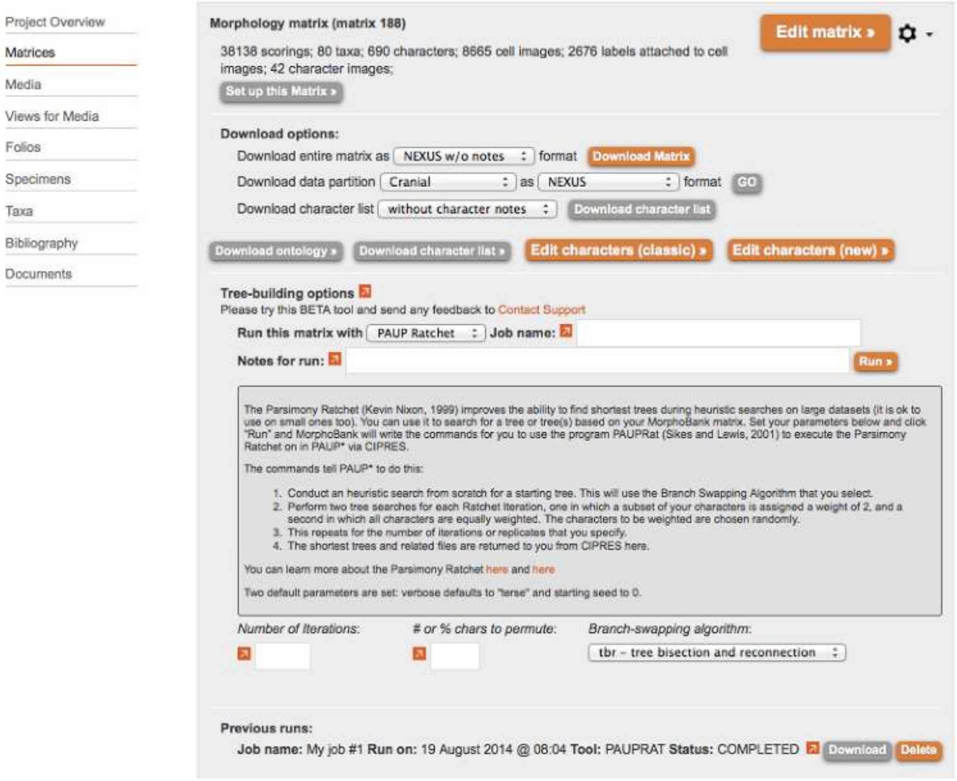
Usage Examples

Three web applications currently are using the CRA to provide access to CIPRES resources: MorphoBank,[10,11] IRD,[8] and ViPR.[9] MorphoBank supports scoring morphological characters, particularly by collaborative teams to create matrices for inferring phylogenetic trees. Matrices are the raw data that are used to build phylogenetic trees, and they consist of scientific observations about the distribution of heritable traits ("characters") such as molecular sequence data, anatomy, physiology, or behavior. The creation of "supermatrices" is the well-established practice of "combining all systematic characters into a single, giant phylogenetic matrix and then analyzing all the characters simultaneously"[2] so that all traits may impact tree structure. The organization and databasing of traits such as anatomy, physiology, and behavior that form parts of supermatrices is relatively challenging and complex, often requiring image display and metadata tools, and the web application MorphoBank was built to assist scientists who are collecting and databasing such information. MorphoBank has a highly developed interface for scoring characters, organizing and archiving metadata about characters (eg, notes, citations), and adding media that can be zoomed and labeled to illustrate what is meant by a particular homology. It automatically archives these matrices and metadata, as well as other supporting materials (eg, trees) associated with peer-reviewed phylogenetic work. Prior to adopting the CRA, it had no support for running tree inference on user-created matrices. Developers at MorphoBank incorporated tree inference via the CRA by creating a web form that allows users to configure runs for the parsimony tool PAUPrat (Fig. 1) seamlessly without leaving the MorphoBank web site. On submission, MorphoBank creates the necessary instruction set, forwards the job submission to the CRA, retrieves the results, and displays them in the user area for download. The implementation of PAUPrat represents a proof-of-concept exercise to test the integration of CIPRES tools into the MorphoBank workflow. Integration of additional CRA tools into the MorphoBank site is planned.

The IRD and ViPR are public database and analysis resources for the study of influenza viruses and Category A–C priority pathogens, respectively. Both resources provide a workbench with integrated database and computational support for virus research. The IRD and ViPR web applications natively support tree inference and other computational biology methods using several community tools. IRD and ViPR have historically provided web forms and required infrastructure to manage job submissions and results retrieval. However, with the continued growth of available data, IRD/ViPR developers sought a mechanism to support analysis tasks that exceeded the capacity of their available compute resources. Using the CRA, IRD/ViPR now facilitates direct access to the RAxML tool on XSEDE (RAXMLHPC2_TGB) through their existing phylogenetic tree analysis portal. The tree analysis input page in IRD/ViPR allows users to configure parameters for RAxML runs. On submission, when the number and size of the input sequences exceeds a prespecified threshold, a lightbox appears to recommend that the user submit their job to CIPRES through the existing IRD/ViPR interface (Fig. 2). A REST client using the Jersey framework implemented in IRD/ViPR then communicates with the RESTful services at CIPRES. After jobs are successfully submitted to CIPRES, IRD and ViPR periodically poll the job status and retrieve the results once the jobs are finished. The results are then saved to the user's private workbench area in IRD or ViPR for further use. Thus, the results obtained either from local IRD and VIPR or CIPRES resources can be accessed, visualized, and analyzed seamlessly by the user without ever leaving the IRD and ViPR web sites.

Future Directions

Our ongoing development plans include incorporating CRA services into the desktop applications such as raxmlGUI,[5] siMBa,[6] and Mesquite.[7] Each of these applications can run one or more tree inference codes on the desktop. In all cases, we are working to implement the ability to submit large tree inference calculations to the CRA from within these desktop applications. The highly developed graphical interfaces for



Morphology matrix (matrix 188)

38138 scorings; 80 taxa; 690 characters; 8665 cell images; 2676 labels attached to cell images; 42 character images;

Download options:

Download entire matrix as NEXUS w/o notes format

Download data partition Cranial as NEXUS format

Download character list without character notes

Tree-building options

Please try this BETA tool and send any feedback to Contact Support

Run this matrix with PAUP Ratchet **Job name:**

Notes for run:

The Parsimony Ratchet (Kevin Nixon, 1999) improves the ability to find shortest trees during heuristic searches on large datasets (it is ok to use on small ones too). You can use it to search for a tree or tree(s) based on your MorphoBank matrix. Set your parameters below and click "Run" and MorphoBank will write the commands for you to use the program PAUPRat (Sikes and Lewis, 2001) to execute the Parsimony Ratchet on in PAUP* via CIPRES.

The commands tell PAUP* to do this:

1. Conduct an heuristic search from scratch for a starting tree. This will use the Branch Swapping Algorithm that you select.
2. Perform two tree searches for each Ratchet iteration, one in which a subset of your characters is assigned a weight of 2, and a second in which all characters are equally weighted. The characters to be weighted are chosen randomly.
3. This repeats for the number of iterations or replicates that you specify.
4. The shortest trees and related files are returned to you from CIPRES here.

You can learn more about the Parsimony Ratchet here and here

Two default parameters are set: verbose defaults to "terse" and starting seed to 0.

Number of iterations:

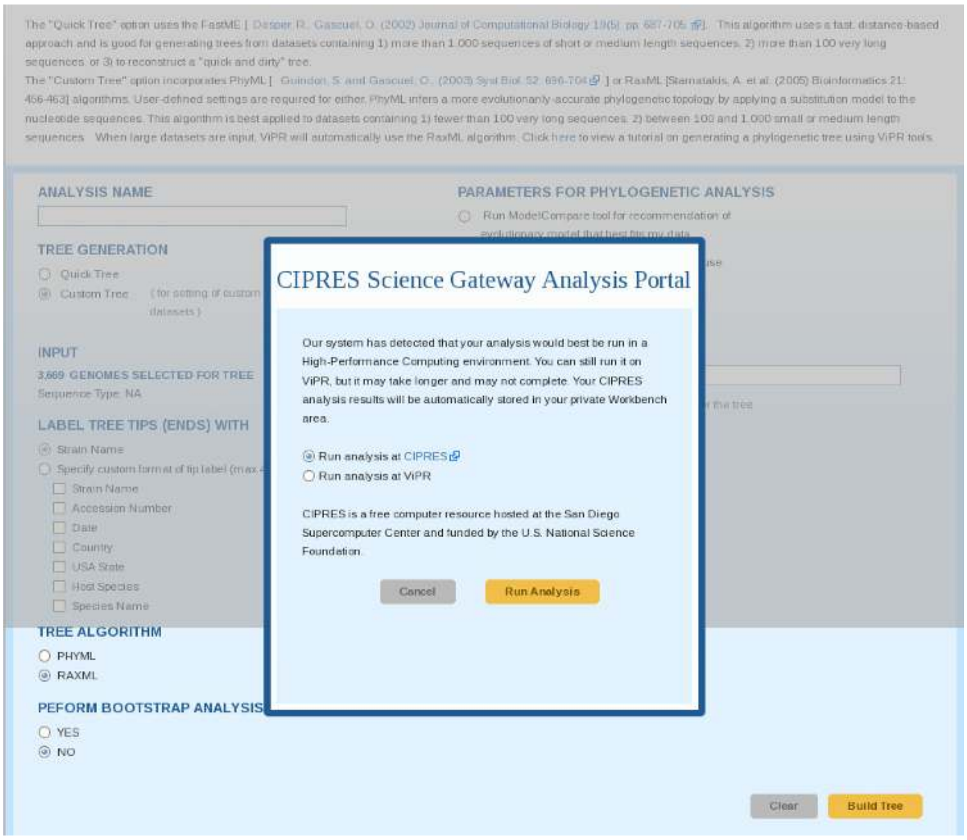
# or % chars to permute:

Branch-swapping algorithm: tbr – tree bisection and reconnection

Previous runs:

Job name: My job #1 **Run on:** 19 August 2014 @ 08:04 **Tool:** PAUPRAT **Status:** COMPLETED

Figure 1. Screen shot showing web form for access to PAUPRat in MorphoBank.



The "Quick Tree" option uses the FastME [Desper R., Gascuel, O. (2002) Journal of Computational Biology 19(5), pp. 687-705]. This algorithm uses a fast, distance-based approach and is good for generating trees from datasets containing 1) more than 1,000 sequences of short or medium length sequences, 2) more than 100 very long sequences, or 3) to reconstruct a "quick and dirty" tree.

The "Custom Tree" option incorporates PhyML [Guindon, S. and Gascuel, O., (2003) Syst Biol. 52: 696-704] or RaxML [Stamatakis, A. et al. (2005) Bioinformatics 21: 456-463] algorithms. User-defined settings are required for either. PhyML infers a more evolutionarily-accurate phylogenetic topology by applying a substitution model to the nucleotide sequences. This algorithm is best applied to datasets containing 1) fewer than 100 very long sequences, 2) between 100 and 1,000 small or medium length sequences. When large datasets are input, ViPR will automatically use the RaxML algorithm. Click here to view a tutorial on generating a phylogenetic tree using ViPR tools.

ANALYSIS NAME

PARAMETERS FOR PHYLOGENETIC ANALYSIS

○ Run ModelCompare tool for recommendation of evolutionary model that best fits my data

TREE GENERATION

○ Quick Tree

● Custom Tree (for setting of custom datasets)

INPUT

3,669 GENOMES SELECTED FOR TREE

Sequence Type: NA

LABEL TREE TIPS (ENDS) WITH

● Strain Name

○ Specify custom format of tip label (max 4

□ Strain Name

□ Accession Number

□ Date

□ Country

□ USA State

□ Host Species

□ Species Name

TREE ALGORITHM

○ PHYML

● RAXML

PEFORM BOOTSTRAP ANALYSIS

○ YES

● NO

CIPRES Science Gateway Analysis Portal

Our system has detected that your analysis would best be run in a High-Performance Computing environment. You can still run it on ViPR, but it may take longer and may not complete. Your CIPRES analysis results will be automatically stored in your private Workbench area.

● Run analysis at CIPRES

○ Run analysis at ViPR

CIPRES is a free computer resource hosted at the San Diego Supercomputer Center and funded by the U.S. National Science Foundation.

Figure 2. Screen shot showing IRD/ViPR lightbox pop up, prompting user to choose the option of running the analysis task using the CIPRES resource.

each of these tools will make it easy for users to configure their analyses on the desktop, and submission of the jobs to CIPRES will free up the desktop machine for other tasks. CRA services will also be made available through the web application offered by the National Center for Genome Analysis Support (NCGAS[22]). Plans are also under way to incorporate CRA services into the bioKepler workflow tool,[23] so that CRA tree inference services can be incorporated into more complex workflows in a visual programming environment.

The CIPRES group is developing worked examples/code samples to help individual scientists who wish to access the CRA using command line/scripting tools and will support users who wish to incorporate the CRA services into their applications. Java and Python examples are available from the CRA web site https://www.phylo.org/restusers. Perl and Javascript examples will be available in the near future.

Acknowledgments

The authors wish also to thank the support staff of the XSEDE project and system administrators of the Trestles, Gordon, and TSCC computers at SDSC for their help in managing issues as they arose.

Author Contributions

Conceived and designed the experiments: TS, SK, SH. Analyzed the data: TS, SK, SH, MP. Wrote the first draft of the manuscript: MAM. Contributed to the writing of the manuscript: TS, EBK, MO. Agree with manuscript results and conclusions: MAM, TS, BEP, SH, EBK, RHS, MP, SK, MO. Jointly developed the structure and arguments for the paper: MAM, TS, MO, EBK. Made critical revisions and approved final version: MAM, TS, EBK, RHS, MO. All authors reviewed and approved of the final manuscript.

REFERENCES

1. Burleigh JG, Alphonse K, Alverson AJ, et al. Next-generation phenomics for the tree of life. *PLoS Curr.* 2013;5, 1–6.
2. de Queiroz A, Gatesy J. The supermatrix approach to systematics. *Trends Ecol Evol.* 2007;22(1):34–41.
3. O'Leary MA, Bloch JI, Flynn JJ. The placental mammal ancestor and the post–K-Pg radiation of placentals. *Science.* 2013;339(6120):662–7.
4. Miller MA, Pfeiffer W, Schwartz T. Creating the CIPRES Science Gateway for Inference of Large Phylogenetic Trees. In: SC10 Workshop on Gateway Computing Environments (GCE10). 2010.
5. Silvestro D, Michalak I. raxmlGUI: a graphical front-end for RAxML. *Org Divers Evol.* 2011;12(4):335–7.
6. Mishra B, Thines M. siMBa—a simple graphical user interface for the Bayesian phylogenetic inference program MrBayes. *Mycol Prog.* 2014;13(4):1255–8.
7. Maddison WP, Maddison DR. *Mesquite: A Modular System for Evolutionary Analysis. Version* 2.5. 2008. Available at: http://mesquiteproject.org.
8. Squires RB, Noronha J, Hunt V, et al. Influenza research database: an integrated bioinformatics resource for influenza research and surveillance. *Influenza Other Respi Viruses.* 2012;6(6):404–16.
9. Pickett BE, Sadat EL, Zhang Y, et al. ViPR: an open bioinformatics database and analysis resource for virology research. *Nucleic Acids Res.* 2012;40:D593-8.
10. O'Leary MA, Kaufman SG. *MorphoBank* 3.0*: Web Application for Morphological Phylogenetics and Taxonomy.* 2012. Available at: http://www.morphobank.org.
11. O'Leary MA, Kaufman S. MorphoBank: phylophenomics in the "cloud". *Cladistics.* 2011;27(5):529–37.
12. Drummond AJ, Rambaut A. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evol Biol.* 2007;7:214.
13. Bouckaert R, Heled J, Kühnert D, et al. BEAST 2: a software platform for bayesian evolutionary analysis. *PLoS Comput Biol.* 2014;10(4):e1003537.
14. Price MN, Dehal PS, Arkin AP. FastTree 2 – approximately maximum-likelihood trees for large alignments. *PLoS One.* 2010;5(3):e9490.
15. Zwickl DJ. *Genetic Algorithm Approaches for the Phylogenetic Analysis of Large Biological Sequence Datasets Under the Maximum Likelihood Criterion.* Austin, TX: The University of Texas at Austin; 2006.
16. Darriba D, Taboada GL, Doallo R, Posada D. jModelTest 2: more models, new heuristics and parallel computing. *Nat Methods.* 2012;9(8):772. Available at: http://www.nature.com/nmeth/journal/v9/n8/abs/nmeth.2109.html.
17. Katoh K, Toh H. Parallelization of the MAFFT multiple sequence alignment program. *Bioinformatics.* 2010;26(15):1899–900.
18. Huelsenbeck JP, Ronquist F. MrBayes: Bayesian inference of phylogeny. *Bioinformatics.* 2001;17(8):754–5.
19. Lartillot N, Lepage T, Blanquart S. PhyloBayes 3: a Bayesian software package for phylogenetic reconstruction and molecular dating. *Bioinformatics.* 2009; 25(17):2286–8.
20. Stamatakis A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics.* 2014;30(9):1312–3.
21. Rifaieh R, Unwin R, Carver J, Miller MA. SWAMI: Integrating biological databases and analysis tools within user friendly environment. In: Cohen-Boulakia S, Tannen V, eds. *Data Integration in Life Sciences.* Vol. 4544. Berlin Heidelberg: Springer; 2007:48–58.
22. Stewart C. *National Center for Genome Analysis Support.* 2013. Available at: http://ncgas.org/index.php.
23. I. Altintas, J. Wang, D. Crawl, W. Li, "Challenges and approaches for distributed workflow-driven analysis of large-scale biological data", in: Proceedings of the Workshop on Data analytics in the Cloud at EDBT/ICDT 2012 Conference, DanaC2012, 2012, pp 73–78.