User, data, and job submission patterns for a highly accessed science gateway

Mark A. Miller
San Diego Supercomputer Center
9500 Gilman Drive
La Jolla, CA 92093
mmiller@sdsc.edu

Terri Schwartz
San Diego Supercomputer Center
9500 Gilman Drive
La Jolla, CA 92093
terri@sdsc.edu

Wayne Pfeiffer
San Diego Supercomputer Center
9500 Gilman Drive
La Jolla, CA 92093
pfeiffer@sdsc.edu

ABSTRACT
The CIPRES Science Gateway (CSG) is a public resource created to provide access to community phylogenetics codes on high performance computing resources. The CSG has been in operation since 2009, and has a large and growing user base. As a popular resource, the CSG provides an opportunity to study user behavior and job submissions in a Gateway environment. Here we examine CSG user and data turnover, jobs submissions success rates, and causes for job failures. The results of our investigation provide a better understanding of the populations that use the CSG, and point to areas where improvements can be made in meeting user needs and using resources more efficiently.

CCS Concepts
• D.2.2 [Software Engineering]: Design Tools and Techniques – Modules and interfaces

Keywords
CIPRES, Phylogenetics, Science Gateway, Open Source, Usage patterns, User behavior.

1. INTRODUCTION
Science gateways are web-based portals created to simplify access to important data and HPC resources by users in a specific domain community (1). As the concept of Gateways and their creation is a relatively new development, the science and best practices for creating Gateways is an evolving field. There are many examples of successful web resource that support scientists, but the key factors that make a Gateways successful, and the nature of the interaction of users with Gateways are areas that require considerable study.

The CIPRES Science Gateway (CSG) was created in 2009 (2) to provide the phylogenetics community with access to a set of community sequence alignment and tree inference codes run on powerful HPC resources. Since its inception, the CIPRES Science Gateway has experienced a steady, approximately linear growth in users per month and job submissions per month (Figure 1). The number of core hours consumed per month is noisier, and seems to show a slight increase over the past 12 months. The number of core hours consumed are denominated in Service Units (SU); where 1 core hour at unit priority= 1 SU. The apparent upward trend in SU consumed is due in part to problems with the initial configuration

of the code RAxML (3) on the Comet cluster in May-July 2015, and to the addition and subsequent removal of two computationally

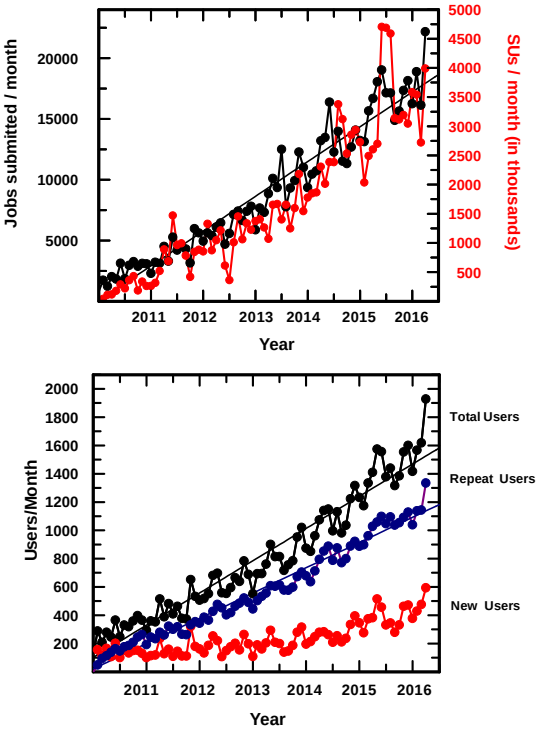

Figure 1. Usage statistics for CSG since its inception. Panel A. Job submissions shown in black, SUs consumed shown in red. SUs are calculated for Lonestar, Abe, Trestles, Gordon, and Comet, and are normalized using one SU on the Trestles cluster = 1. Panel B. Number of unique users per month who submitted one or more jobs to the CSG.

intensive codes, PhyloBayes (4) and Migrate-N (5). The PhyloBayes code was removed from the public interface in late 2014 because consumption by this code was excessive for the CIPRES community allocation.

As of April 2016, the CSG has served more than 17,000 users, and supported more than 2300 publications in all areas of modern biology. At present approximately 1700 users submit jobs to the

Conference'10, Month 1–2, 2010, City, State, Country.

DOI: http://dx.doi.org/10.1145/12345.67890

CSG from around the world. As such, the CSG is not only a valuable resource for its users but it provides a testbed for following usage patterns and user behaviors. The information gained from such studies can be used to improve our understanding of how Gateways can better serve their user populations. Our goal was to investigate usage of the CSG in detail to better understand how the CSG can be improved, and by inference, we hope to advance the science of Gateway creation. Here we analyze some of the basic characteristics of CIPRES usage, including user registration and attrition, job submission, job failures to see what could be learned that may be of use for gateway creators.

2. User registration.

The relationship between user registration and job submission can be informative in understanding what subgroups exist within the user population. For example, based on survey responses, we know that the CSG is used for curriculum delivery by at least 97 instructors, and a small fraction of their students respond to our annual survey. But we have little quantitative information about this population. Also, we allow individuals to register as Guests, and use the resource, with no way to easily re-access their data once the web session expires (after 30 minutes of inactivity).

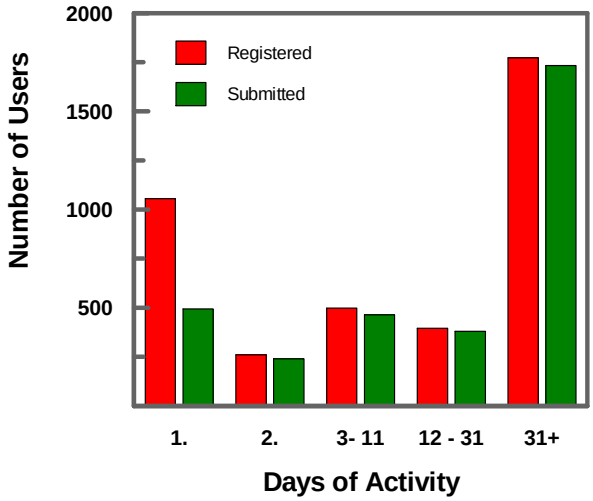

Figure 2. Number of registered user accounts (red) and number of user accounts that submitted jobs (green) as a function of the number of days the account was active after registration.

We examined registrations and job submissions during the calendar year 2015. Between Jan 1, 2015 and Dec 31, 2015, 7,365 new accounts were created through the CSG web site. Of these, 3,385 were Guest accounts, intended to last a single day so users can explore the functionality of the site, and decide if it may be useful for them. A single individual can create multiple Guest accounts, so it is uncertain exactly how many unique users created Guest accounts. However, jobs submitted from Guest accounts use many tools, command lines, and have many different input files, suggesting that these accounts are not created by a small group of users. All Guest accounts were active for only a single day, in other words, the date created and date last accessed were the same.

In addition, 3980 users created CSG accounts by completing the CSG user registration form. This involves choosing a username, identifying an email address and providing institutional information. We examined the length of time each user was actively using their account. To do this, we compared the date the account was created and the date the account was last accessed. Figure 2 shows all user-registered accounts created in 2015 binned according to the number of days they were active. 34% of all registrations were for accounts that were active for only 1-2 days. We presumed that these accounts represent short term classroom usage, users who found the CSG did not meet their needs, and users who could not figure out how to use the site effectively. 55% of all users who registered in 2015 used their accounts for one month or less. 45% of CSG users who registered in 2015 used the CSG more than 30 days after their initial registration. Our supposition was CSG accounts that were active for 1-30 days were primarily for classroom use; our expectation was that curriculum using the CSG would involve a one day session or perhaps several day session after which the account would not be used.

To investigate this, we looked at individual bins for evidence of classroom use. We defined classroom use as an event where three or more accounts from a single institution submitted a similar (low) number of jobs in a similar time period. We find evidence for classroom use that follows this definition. For example, among accounts that were active for only 1 day, 19 accounts submitted jobs from the University of South Florida, Manatee, while 27 accounts submitted jobs from the University of Nebraska, Omaha. This usage is clearly for curriculum delivery. When we scored events where three or more users submitted jobs from the same institution, 125/480 (26%) of accounts active for a single day were linked to classroom use. Expanding the analysis revealed examples of possible classroom use in all bins. For accounts active for 1 – 30 days, possible classroom use accounted for 25% of all users. This amount increased to 45% of all accounts among accounts that were active for more than one month. Apparently, classroom use does not mean that accounts are used for a short time.

We were surprised to find that 75% of users who are active for less than one month seem to be working independently. An important question is how many of these users have a sporadic use pattern, and return to use their account after a long period of inactivity, and how many abandon their accounts in less than one month. We note that between March 1 and April 21 2016, 79 users who were active for less than 1 month returned to use their CSG accounts again. The majority of these (72) had been active for 3-30 days but two users who had been active for only one day returned to use their account. We plan to continue our study to better understand usage by individuals who use the resource infrequently, with time spans of several months or more between sessions. This will help us establish what fraction of users use the resource in more than one month, and what fraction do not return.

3. Job Submission

As noted above, the registered CSG user population included individuals who created accounts and never submitted a single job (17%). It also included seven users who have submitted more than 500 jobs in 2015 alone. The user population is bimodal in terms of job submissions. Over the lifetime of the CSG, 38% of all users have submitted more than 20 jobs, and 38% have submitted less than 5; the remaining 24% submitted between 6 and 19. We have historically presumed that individuals with less than 5 submissions represent students taking classes, but as noted above, at least 75% of these users seem to be working independently, based on institutional affiliation and time stamp of submission. It remains to be determined if these users only needed a few job runs, or if the resource did not meet their needs.

Of the 3,980 users who created a registered account in 2015, 3,311 (83%) actually submitted a job. To understand this better, we examined job submission as a function of the length of time these accounts were active (Figure 2). Accounts that were active for a single day had the lowest rate of submission. Only 47% of users who used their account for a single day actually made a submission. Similarly, of the 3,385 Guest sessions that were created during this time period, 1,493 (44%) submitted one or more jobs. Apparently many users are unable to figure out how to make a submission, or find that the tools available do not meet their needs. In an attempt to understand the low submissions rates, we e-mailed surveys to the 561 users who registered but did not submit a job. The response rate to these surveys was low (9 responses were received), but 6/9 users reported they found the interface too difficult to use, 1/9 reported the tool they required was not available. On the other hand, submissions were received from greater than 94% of accounts that were active for more than 1 day. Apparently users who are able to overcome usability issues and/or have more experience/commitment are able to make job submissions. It is interesting to note that although job submission rates are very high in accounts that have more than one day of activity, in each bin there are some users who have never submitted a job. Among users who used their accounts for more than one month, 39/1734 (about 2%) have not submitted a job at this time. Since our routine accounting scores only users who have submitted jobs, none of these non-submitting users appear in our routine user statistics (e.g. Figure 1).

4. User Turnover.

Once a CSG user registers, they are forever counted among the population of users, because there is no mechanism to unregister. As a result, there is always growth in user number over time. But the more important metric, from our perspective, is what fraction of users remain active over time. On 11/3/2014 we began to store the date of each user's last login as part of a policy to sunset inactive accounts. A single login in a calendar year is scored as account activity. The last_login value is set to the current date anytime a user logs in (it can also be set by the admins at user request). After one calendar year of inactivity, the user receives an email warning them of the imminent deletion of their account and all associated data. If the user does not log in within 30 days of the one year anniversary, the account and its contents are deleted. Statistics about the jobs run from a deleted account are retained in the database, however. We can follow user attrition by monitoring the last_login date, and by monitoring whether their data remains in the database.

As of 4/27/2016, 13,268 users have submitted one or more jobs from a registered CSG account; of these 5,561 user accounts have been deleted due to user inactivity (the last login date was more than 396 days since the initial registration date). We did not consider Guest accounts, which are active for only one day. In the population of users that have logged in within the past 396 days (those inactive for longer periods would have their data deleted automatically), 264 users (2%) have deleted all data in their accounts, although their accounts are still technically active. In all, about 57% of users who have registered since the first day of operations still have active CIPRES accounts.

The departure of CSG users has a pronounced time dependence. Table 1 shows user loss as a function of calendar year. User departure was scored as the absence of a user's job results in the database for jobs submitted in the reported year. This can reflect either proactive deletion or automatic deletion due to inactivity. The rates of user retention are artificially high in 2015 and 2016,

Table 1. Loss of non-Guest users with time.

Year	Total Users	Users w/ Data	Fraction Remaining	Proactive	Fraction Proactive
2012	3133	1053	0.34	57	0.02
2013	3733	1615	0.43	77	0.02
2014	4739	2604	0.55	88	0.02
2015	6026	5469	0.91	108	0.02
2016	3773	3732	0.99	41	0.01
2015A	2237	1763	0.79	26	0.01
2015B	5341	5249	0.98	90	0.02

because most (2015) or all (2016) submissions are less than 396 days ago, and so inactive accounts have not yet been eliminated. We can show the effect of automatic deletion of inactive accounts by separating the 2015 year into usage before 3/25/2015, when inactive accounts have been deleted, and usage after 3/25/2015, where there is no automatic deletion because all jobs were less than 396 days ago. The effect of automatic deletion on eliminating inactive users is clear: 79% of users who were active before 3/25/2015 remain, whereas 98% of users after that time remain.

The trend in user loss fits well to an exponential decay function where the average registered user "half-life" is approximately 2.5 years. The number of users who proactively delete their data is about 2% of registered users independent of the year of the year of operation.

5. Data Storage/Turnover

In addition to considering users who delete all of their data, or have their data deleted, another question is how much of the data created using CIPRES is retained within the database. The amount of data stored at present is 39.6 TB uncompressed/ 10 TB compressed. Table 2 shows the amount of data remaining in the CIPRES database for each year when the job was run. The amount of data stored per user is modest reaching a maximum of 450 MB/user for data created in 2014, then decreasing in 2015 and 2016. When users are binned according to the storage size their data requires, it is evident that the lower data per user values in 2015 and 2016 are caused by selective removal of users with smaller amounts of data when inactive accounts are deleted. The fraction of users with less than 1 MB in storage drops from 18% in 2015 and to 3% in 2014, where automatic account deletion has been performed on all submitted jobs.

Table 2. Data in the CSG database by year

Year	# Users	Data (TB)*	GB/ user	1-999 GB	1-999 MB	< 1 MB
2012	620	1.94	3.13	193	403	24
2013	1045	4.08	3.90	392	623	30
2014	2036	9.14	4.49	820	1152	64
2015	6220	16.5	2.65	1450	3635	1135
2016	4397	6.92	1.67	880	2868	649

*Uncompressed. The compression reduces the size by 75% on disc.

We next asked what percentage of all jobs run by the CSG still remain in the database. The results of this analysis are shown in

Table 3. As expected based on user turnover (Table 1), the fraction of jobs remaining in the database decreases with time. However, the fraction of job results remaining in the database is consistently lower than the fraction of users remaining, indicating that users proactively delete a significant number of jobs. Approximately 40% of user jobs were deleted proactively by the user in 2015 and 2016, since automatic account deletion has affected not yet been implemented on most of these submissions. Approximately 5% of submitted jobs are deleted in each year when the user aborts the job run (not shown). The remaining jobs that are deleted are a mixture of completed jobs and jobs that failed for any of a number of reasons, including user error, misconfiguration, system errors, etc.

Table 3. Fraction of Jobs remaining in Database.

Year	Jobs Run	Jobs in DB	Fraction
2012	86,288	17,729	0.21
2013	126,826	33,592	0.26
2014	164,383	63,855	0.39
2015	214,137	117,567	0.55
2016	84,305	51,196	0.61
Total	676,627	284,286	0.42

The total number of active users (users that submitted one or more jobs top XSEDE) binned according to use is shown in Table 4. Over the past three years there is a trend toward decreasing numbers of users in the upper three bins. There is also a significant increase in the number of users in the lowest (1-100 SU) bin, and in the percentage of these users as part of the overall population. This may represent an increase in the use of CSG for curriculum delivery, since these users consume very little resources (the number of people reporting they use CIPRES for curriculum delivery has increased from 57 to 94 on the past 2 years. The number of users requiring more than 10,000 SUs is relatively constant, and this group typically consumes 65-70% of all resources distributed by the CSG. The remaining 25-30% of resources was consumed by 90% of users in the groups between 0 and 10,000 SUs per year. The number of users requiring 100-10,000 SUs also increased in number, but not as a fraction of the overall population. Overall, the data in Table 5 show clearly that the CSG is distributing XSEDE resources across a very broad and growing user community. Interestingly, the growth is primarily at the lower end of the usage spectrum

6. Job Outcomes

The CSG offers several codes run on XSEDE resources. The most accessed codes are shown in Table 5. Other codes amount to less than 1% of total CSG resource consumption. The maximum run time allowed for a given job is 168 hours for jobs run on XSEDE, and 72 hours for serial jobs run on the TSCC.

In a previous work (2), we investigated job outcomes from CIPRES job submissions to determine what the major impediments to job submissions were for the resource. This information serves as a guide to future development to improve efficiency and/or the user experience. Here we report an investigation of job outcomes in the current version of the CSG, to determine where the weak points in job submission lie, and what can be done about them. We investigated all job submissions in a two week period for the major codes supported by the CSG: RAxML, BEAST, BEAST2, MrBayes, and Migrate-N. The log files created by individual job runs were parsed using MacroScheduler14, using Regex searches for key diagnostic phrases within the log files. The output was analyzed to score job successes and failures, and to identify the reasons for individual job failures. The results of the work are presented in Table 6. The success rates of CIPRES job codes range from 56-74% for all codes except MrBayes. Successful job completion for MrBayes was only 27%. The lower success rate for MrBayes is explained by the very high rate of jobs that reach their maximum configured wall limit, and the very high rate of user input errors. More than half of MrBayes submissions fail for one of these two reasons. Although timeouts often reflect runs that have gone for a full 168 hours, these runs can be continued using a restart interface, and so do not represent a total loss of invested resource. Both RAxML and MrBayes have relatively high failure rates from user input errors. These codes differ from BEAST, BEAST2, and Migrate-N in that the latter each has a desktop GUI tool that creates the input files users submit. Apparently these GUI tools eliminate many user errors. The error rate for MrBayes is particularly high because input files for this code use the complex and rather demanding Nexus format, whereas RAxML use the much simpler

Table 4. Binned Usage of the CIPRES Science Gateway over the past three allocation years

Usage	Fraction of Total SUs			Total Users			Fraction of Users		
	2013-14	2014-15	2015-16*	2013-14	2014-15	2015-16‡	2013-14	2014-15	2015-16*
> 100 K	0.062	0.053	0.034	4	7	2	0.001	0.001	0.0002
50 – 100 K	0.151	0.114	0.098	35	32	29	0.008	0.005	0.003
30 - 50 K	0.230	0.207	0.144	100	102	72	0.025	0.018	0.009
10 - 30 K	0.358	0.406	0.399	302	426	439	0.074	0.075	0.055
1 - 10 K	0.201	0.194	0.288	851	995	1579	0.209	0.175	0.198
0.1 - 1K	0.020	0.023	0.031	876	1155	1679	0.215	0.204	0.199
1 – 100	0.002	0.003	0.004	1906	2946	4246	0.468	0.520	0.535
Total				4074	5663	7942			

* The figures for 2015-2016 were calculated using data from July 1, 2015 – April 1, 2016. ‡ The total numbers of users for 2015-2016 are projections based on trends through April 1, 2014.

relaxed Phylip format. Still it is clear that manually editing files in much less efficient than using a GUI-based tool.

The programs BEAST and BEAST2 experienced a much higher level of crashes than the other programs. Beast relies upon the BEAGLE library for improved parallelization of the code, and most of the crashes for this program are the result of underflow,

minimum successful run time varies widely among codes. The shortest time observed for a completed run was found with RAxML, which has a minimum time of less than 1 sec to create productive output. Similarly, MrBayes runs were found that produced useful output in 2 sec. On the other hand, BEAST, BEAST2, and Migrate-N all had significant lag times (45-524 s) to produce useful output. This information can be used to help us

Table 5. Resource consumption by Codes offered by the CSG

Code	Version	Number of jobs	Fraction of jobs	Number of SUs*	Fraction of SUs*	SUs/job
BEAST	1.8.x	21,293	0.21	2,592,435	0.29	121
BEAST2	2.3.x	7,024	0.07	560,946	0.06	79
Migrate-N	3.6.11	962	0.01	979,019	0.11	1,018
MrBayes	3.2.x	25,081	0.25	2,347,722	0.26	93
RAxML	8.x.x	39,551	0.39	2,057,737	0.23	52

Table 6. Job outcomes for CIPRES job runs involving major codes.

	Beast	Beast2	Migrate-N	RAxML	MrBayes
Completed	0.59	0.56	0.59	0.74	0.27
User Input error	0.04	0.07	0.06	0.15	0.29
Crash	0.15	0.24	0.04	0.04	0.01
Time out	0.12	0.07	0.21	0.03	0.25
Killed by user	0.08	0.04	0.08	0.03	0.09
System error	0.001	-	-	0.001	0.001
No Information	0.01	0.01	0.02	0.01	0.09
Avg. run time (h)	26.2	11.9	25.6	0.6	7.7
Avg. run time success (h)	31.9	13.3	29.5	0.7	10.8
Min success run time (s)	45	524	78	1	2

which causes the program to crash to do numerical instability between BEAST and the BEAGLE library. BEAST and BEAST2 also suffer from crashes when the initial model has zero probability. For BEAST2, 69% of job crashes were the result of out of memory errors. Another 20% were from unknown causes.

With the exception of timeouts, the crashes noted here (with rare exceptions) all occur very early in the job runs, and so are an annoyance, but are not costly in terms of compute resources. We also note that failures due to system errors are very rare once a job has been submitted successfully. The only job failure we found that were costly in resources were Migrate-N jobs that the time out. Migrate-N currently does not have a restart function, and creating one would clearly have benefits.

To help in identifying job runs that fail more easily, we tried to determine if there is a minimum job run time required for success. To help in identifying job runs that fail more easily, we tried to determine if there is a minimum job run time required for success. If so, any jobs shorter than the minimum necessary time can be flagged as failures, and studied further. As shown in Table 6, the quickly identify failed jobs in the future.

7. Conclusions

The analysis reported here has some important implications for the CSG, and raises some questions that require further investigation:

1. We have long assumed that users who register and use their accounts for less than one month are using the CSG as part of a classroom exercise, or as part of instruction. Our results challenge that notion. While we certainly see clear use patterns that correspond to many users in a classroom using the resource, the vast majority of short-term users use the resource at diverse times, and from many different institutions. Perhaps these users have short term needs, or perhaps they find the CSG hard to use, or lacking some feature they require. We can gain more insight into this by studying the behavior of the short term users, how many jobs they run, and how much compute resources they use.

2. We found that only half of the users who come to visit the CSG through a guest session ever submit a job. Moreover, 17% of users who take the time to create an account also never submit a job.

Preliminary feedback suggests that users may not be able to figure out how to use the resource when they arrive. It is important for us to think about how we might provision the initial landing spot for new CSG users to make it more usable for new visitors. Together with the evidence about short term usage patterns, the results raise some concerns about whether or not usability is a serious problem for the CSG. In both cases, we can devise automated tools that attempt to communicate with users within 2-3 days of their creating an account if they have not submitted a job yet, and offer assistance.

3. Despite the loss of many users after only a few days, there is also a stable population of active CIPRES users. The "half-life" of approximately 2.5 years for active accounts seems consistent with the expected lifetime of graduate student and postdoctoral positions.

4. We see that users actively manage their data; and that perhaps as much as 40% of data produced by submitted jobs were deleted by users in 2015. We also found that our policy of deleting data in accounts that are inactive for more than one year is an important mechanism for controlling storage costs. The current solution still permits users to store data indefinitely for the cost of logging in once per year, but does not retain data the user is no longer interested in for more than one year.

5. Although the CSG was designed with the intention of serving higher end computational needs of phylogenetics communities, the fastest growing user population is at the low end. We need to investigate this further to understand the intentions and the needs of this population.

6. The success rates among the various codes offered by CIPRES vary dramatically. The variation and the failures we observe are indigenous to the codes, and cannot be attributed to or addressed by CIPRES for the most part. It is clear that codes that provide in input file generated by a local desktop application have much higher success rates than those created by direct editing. CIPRES can help the use of code like MrBayes and RAxML by providing tools to check input files, and edit errors as they are discovered. Creating this functionality is in our future plans.

8. Acknowledgements

The work described here was supported by NSF DBI-1262628, NSG DBI-1146949, and NSF ACI-1339856. Development support was also received from allocation award TG-DEB090011 from the XSEDE project, which is sponsored by the National Science Foundation. Initial software development for the workbench framework was also supported by NIH GM73931-01.

9. REFERENCES

1. Wilkins-Diehr, N., and Lawrence, K. A. (2010) Opening science gateways to future success: The challenges of gateway sustainability. in Gateway Computing Environments Workshop (GCE), 2010

2. Miller, M., Pfeiffer, W., and Schwartz, T. (2010) Creating the CIPRES Science Gateway for inference of large phylogenetic trees. Proceedings of the Gateway Computing Environments Workshop (GCE), 2010, 1 - 8

3. Stamatakis, A. (2014) RAxML Version 8: A tool for Phylogenetic Analysis and Post-Analysis of Large Phylogenies. Bioinformatics

4. Lartillot, N., Lepage, T., and Blanquart, S. (2009) PhyloBayes 3: a Bayesian software package for phylogenetic reconstruction and molecular dating. Bioinformatics 25, 2286-2288

5. Beerli, P., and Palczewski, B. (2010) Unified Framework to Evaluate Panmixia and Migration Direction Among Multiple Sampling Locations. . Genetics 185, 313-326