Collaborative Research: ABI Development: Cultivating a sustainable Open Tree of Life

1 Overall goals of proposal

The Open Tree of Life project synthesizes scientific knowledge about the phylogeny of all life. The project ("OpenTree" hereafter) has constructed a comprehensive, dynamic, and digitally-available tree of life across 2.5 million taxa by bringing together published phylogenetic trees and taxonomic data (see Figure 1). This synthetic tree, as well as all of the data and code used to generate this estimate are freely available online. Together these artifacts provide a valuable resource to a wide variety of users, for example: evolutionary biologists who require phylogenetic estimates for downstream analyses, ecologists who want to estimate phylogenetic diversity of study sites, and members of the public who are interested in the shared ancestry of different lineages.

Improving the quality of the synthetic tree and taxonomy is important in attracting new users, and the project depends on community curation to improve these resources. Therefore, we seek funding for several extensions to the OpenTree platform that motivate deposition of published phylogenies and curation of existing data, improve reliability of the synthetic tree and taxonomy, and lay the groundwork for long-term sustainability through subscription and software-as-a-service models.

This proposal was co-written by Karen A. Cranston (KAC, the lead PI of the original OpenTree award), Mark T. Holder (MTH, the PI of the OpenTree award to the University of Kansas, KU), and Emily Jane McTavish (EJM, a former postdoctoral researcher with the project and now an assistant professor at UC Merced), though KAC will not be funded by this proposal. First, we will review the goals and status of the OpenTree project; then propose specific tasks to be completed; provide a workplan and plans for dissemination, user engagement, and sustainability, and finally discuss the broader impacts of the proposed work.

2 Response to reviewers

In response to previous reviews, we have included statistics on usage and community engagement; reduced the mainte-

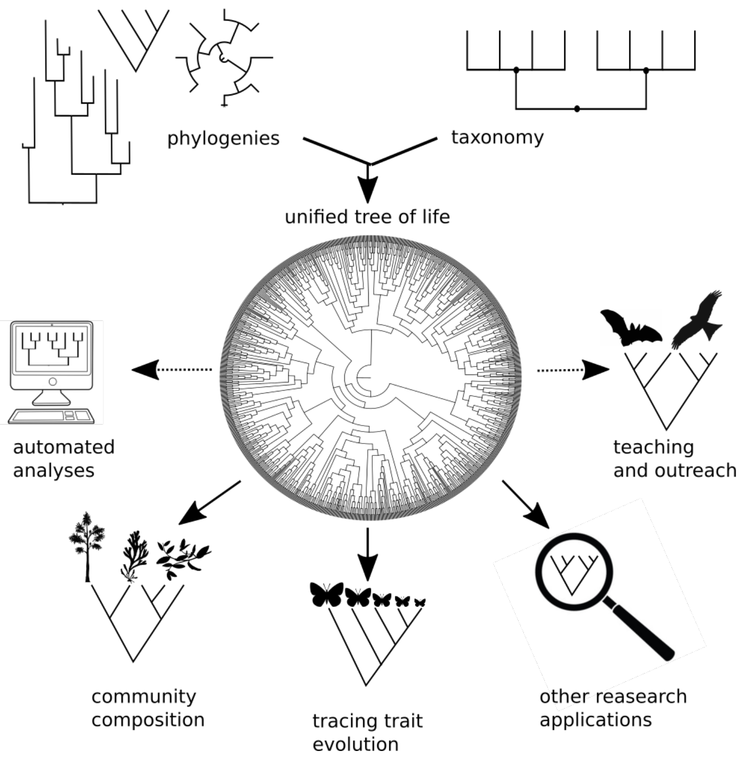


Figure 1: OpenTree data resources and applications: the collection of well-curated, published phylogenetic trees; the backbone taxonomy (built by merging multiple online taxonomies); and the synthetic tree of life (built by algorithmically combining the input phylogenies and backbone taxonomy). This synthetic tree is then programatically accessible for downstream analyses, as well as freely available online for teaching and outreach. Some examples of downstream analyses include assessing phylogenetic relatedness across ecological communities, or tracing trait evolution, among others.

nance component, taking into account work that has been completed in the past year; provided citations for our now-published summary tree and taxonomy methods; and described how our stated goals are driven by community feedback. The major critiques centered on plans to incorporate trees based on GenBank data. We have completely re-written this section to address concerns about quality of inferred trees, detail how these trees will be incorporated into the summary tree, and connect the work to prior art that makes it clear that this important work is low-risk.

3 Background and Conceptual Design

Open Tree of Life builds a synthetic tree of life by combining published phylogenetic trees with taxonomic hierarchies. Trees provide resolution and contribute newly-described taxa, while taxonomies provide completeness. The project was initially funded by a three-year collaborative AVAToL grant awarded to 11 investigators from biology, bioinformatics, and computer science. A two-year supplemental award to three bioinformatics labs (led by KAC, MTH, and Stephen Smith at U. Michigan) funded the project until May, 2017. No-cost extensions will support some limited maintenance work until May, 2018.

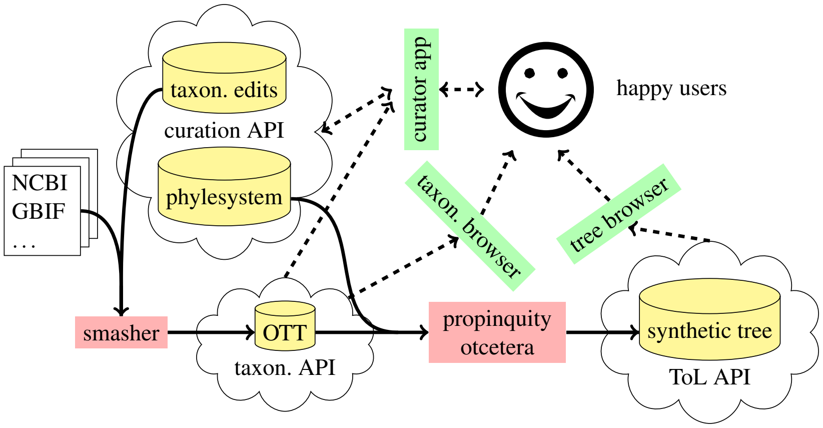


Figure 2: Overview of the software architecture of OpenTree. The main artifacts are shown in yellow. Clouds represent the API (application program interface) layers that provide access to that data. Processing tools to produce OTT and the synthetic tree are shown in pink with solid arrows indicating their inputs and outputs. The JavaScript web applications that allow users to interact with the resources via the API are shown in green.

3.1 Overview of current architecture

The current architecture (see Fig. 2) centers around three types of artifacts: 1. User-curated sets of published phylogenetic estimates (stored in the project's "phylesystem") and taxonomic corrections; 2. A comprehensive phylogenetic taxonomy called the Open Tree taxonomy ("OTT" hereafter) which is built by an OpenTree tool called `smasher` from external taxonomies modified by the user-curated taxonomic edits; and 3. the OpenTree's complete "synthetic tree". This synthetic tree is built by a pair of tools called `propinquity` and `otcetera` from the phylogenetic information in phylesystem and the taxonomic information in OTT. The phylogenetic and taxonomic information used to create this estimate is also exposed via RESTful APIs. Finally, a set of JavaScript-based web applications (the "curator app", "taxonomy browser", and "tree browser") provide user-friendly access to the data and outputs. To avoid clutter, the figure does not depict the interactions between the taxonomy and tree browser and the curation API, but links are in place to allow users browsing the tree or taxonomy to contribute expertise via curation.

The 'Open' in OpenTree is a commitment to transparency of operation and data. All of the code is available under our GitHub organization. All three primary data artifacts (OTT, the phylesystem, and the synthetic tree) are made available for download and via web services. The user-curated data (phylesystem and the taxonomic edits) are available as git repositories to make it easy for users to fetch the latest changes and to preserve provenance of changes.

3.2 OpenTree user communities

The Open Tree of Life project has already been successful in generating widespread interest and engagement among the general public, the phylogenetics community, and bioinformatic developers. In the development phase of this project we plan to maintain and strengthen our engagement with these communities through publications, workshops, feedback forms, and a range of online communication forums.

The publication of the draft tree in PNAS [1] received widespread press coverage including in the Washington Post, NBC news, Scientific American, NPR, and also as commentary on creationist web pages. The Open Tree team fielded questions on the work in a well attended "Ask me anything" discussion on reddit [2], and there were 6.5 million hits and 194,000 visits to the synthetic tree webpage over two days following release. In a typical month in 2017, our web analytics tools report over 20,000 unique visits to the user-facing site that hosts the tree with an additional 1092 visitors of the API server from user-agents that are not recognized as browsers (almost all of these are probably programmatic usage of the API's). One of the major aims of the project was to make phylogenetic information open and accessible to non-specialists, so the widespread public engagement and success is heartening.

This proposal aims to grow OpenTree user communities: researchers who use trees, researchers who produce trees, and other bioinformatics projects. Work that improves the accuracy of the synthetic tree, such as more frequent updating based on published trees (section 4.1.7) and automated incorporation of new taxa (section 4.2.3), benefit all users. Our plans are informed - now and going forward - by feedback provided at workshops, through our website and via our various communication channels.

3.2.1 "Tree consumers"

The first user group are researchers who need a phylogenetic tree for an analysis, figure, or teaching example (see Figure 1). The research use cases are often comparative analyses. The most common requests from these users are to improve the quality of the synthetic tree, and to provide branch length information for that tree. Proposed features for this community include custom synthesis (section 4.1.4) to allow for fine-tuning synthesis inputs and methods, the incorporation of fossil taxa (section 4.2.1) and inference of branch lengths (section 4.2.4). We note that access by these users OpenTree can be through other projects (see section 3.2.3)

3.2.2 "Tree builders"

The second group are primarily systematists and taxonomists. This is the most challenging group to engage. While a high-quality synthetic tree helps build trust with these users, simply improving the tree is not sufficient motivation for them to participate, as they prefer to generate their own phylogenetic estimates rather than re-use trees. Motivating this group is crucial, because the value of the OpenTree platform is closely tied to how well it represents the current state of phylogenetic knowledge, and these users have the expert knowledge. The rates of deposition of phylogenetic estimates into *any* digital repository are still quite low – less than 17% of published phylogenies are archived [3, 4, 5], and much of the data that is archived is difficult to reuse [3, 6, 7]. We have had some success building a community of curators; to date, 120 of the approximately 150 curators registered with the system are biologists who are not directly involved in the project. OpenTree has conducted multiple workshops involving systematists, including events at the annual meeting of the Society of Systematic Biologists and "clade workshops" jointly run with the Arbor Workflows and FuturePhy projects. The clade workshops brought together biologists who work on particular clades in the tree of life, but who have a wide range of expertise (e.g. taxonomists, paleontologists, functional morphologists, *etc*). The clade workshop format has been highly successful for Open Tree on four important axes: (1) the researchers contributed and curated new trees to the OpenTree database, greatly improving our inferences in the clades represented; (2) we were able to discuss directly with our target contributor/user base what improvements would encourage future data submission; and (3) participants have been willing to beta test new features and interfaces after the workshop. Many features proposed here are the direct results of workshop requests.

The most common requests from systematists are the ability to access OpenTree analysis tools before publication of their trees and the ability to update the underlying taxonomy. Thus, we propose implementation of private data stores (section 4.1.3) to make it easier for systematists to compare their unpublished results with the phylogenetic and taxonomic data in OpenTree. The custom synthesis feature (section 4.1.4) will allow expert users to build supertrees targeted to their needs. We also propose two features that will help systematists advertise the importance of their own work: gap analyses to identify understudied parts of the Tree of Life (section 4.1.6) and personalized user homepages (section 4.1.5). These profile pages will let users configure notifications about clades of interest and will also provide links to the parts of the synthetic tree that rely on that user's expertise. Finally, our proposed taxonomy editing features (section 4.2.2) will let taxonomic experts correct our taxonomy and receive credit for doing so.

3.2.3 Other bioinformatics projects The third user community are the external projects already using OpenTree products. Phylotastic [8] uses OpenTree's curator application, phylesystem, taxonomic name recognition (TNRS) to provide a user interface for finding a phylogeny given a taxon or list of taxa, and the tree APIs; Phylotastic alone has accounted for over 10,000 calls to the open tree APIs over the last two years. The Ancestor's Tale relies on the synthetic tree to trace the shared ancestry of humans and other species, and has added that information to an updated version of the book "The Ancestor's Tale" by Dawkins and Wong [9]. Arbor workflows [10] imports trees from OpenTree as the backbone for downstream comparative analyses. Members of rOpenSci have built a client package, 'rotl' [11], to help users access OpenTree APIs from the R programming language. The 'rotl' package has already been downloaded more than 36,000 times [12]. Recently, developers of Mesquite [13], a GUI environment for working with phylogenetic data, added features to improve interoperability with OpenTree. In addition, we are in discussions with developers of the CIPRES portal [14] to make it easier for CIPRES users to access OpenTree services and to directly commit their results to OpenTree. Many of the core infrastructure improvements (section 4.3) help projects that want to deploy a single OpenTree component, or want to collaborate on a feature addition.

We are in discussions with Nico Franz of the EulerX taxonomy reasoning project [15], the PIs of the phyloref phylogenetic nomenclature project [16], and developers of TaxonWorks [17] about coordinating work on taxonomies. We will work with these projects and other interested participants to design an interoperable format for specifying taxonomy amendments (see section 4.2.2). These amendments will improve OTT, but would also help us exchange information with other cybertaxonomy infrastructure projects.

3.3 Results from prior NSF-support MTH and EJM are the PIs of this collaborative proposal; KAC plans to continue to work on the project as a contractor and/ or consultant (depending on the constraints of her employment during the award). EJM is a new investigator. Thus, only results from MTH's prior award are reported here.

MTH received $799,769 over five years from NSF DEB award #1208393 " Collaborative research: Automated and community-driven synthesis of the tree of life," as part of a set of three-year awards to ten institutions (starting in May of 2012) and a set of two-year supplements to the three institutions that were responsible for most of the software written by the project. The supplemental award ends in May, 2017; a no-cost extension will be sought (due to some delays in finding postdocs, we project that approximately 20% of KU's direct costs will still be available to spend at the end of the award period). **Intellectual Merit** The KU award has resulted in four research publications thus far [1, 18, 19, 20]; 1 in review, and 1 more to be submitted in the fall of 2017. The award funded contributions to almost all of OpenTree's software efforts and was the primary funding mechanism for five new software tools which are heavily used as a part of OpenTree: 1. peyotl – a Python library that provides core infrastructure and client-side wrappers for some OpenTree web services; 2. otcetera – a suite of efficient supertree operations written in C++; 3. phylesystem-api – a server-side application for providing access to OpenTree's corpus of published tree estimates; 4. propinquity

– a make-based workflow for producing reproducible, annotated supertrees on the scale of millions of leaves (this is the tool that currently builds the OpenTree's synthetic tree); and 5. physcraper – a Python toolkit for efficiently updating gene tree estimates in light of new sequence data. **Broader Impacts** The software contributions are all available under open source licenses, and the phylesystem software also makes the set of published trees curated via OpenTree continually available as an open git repository (called phylesystem-1) in a manner that tracks the contributions of the biologists involved in curating the data. The award to MTH has trained one (female) graduate student in bioinformatics and trained 2 postdoctoral researchers, one of whom (EJM) is now in a tenure-track assistant professor position at UC Merced. MTH and EJM co-authored an encyclopedia entry in a teaching oriented Encyclopedia of Evolution on statistical testing of phylogenetic trees [21]; as part of that publication, they developed and extended a set of web-browser applications (called mephytis) for teaching about phylogenetic theory. They negotiated the open source release of these apps and their source code as a precondition for their contribution to the Encyclopedia of Evolution.

4 Improving Open Tree of Life

Our proposed extensions to the OpenTree platform fall into two major and one minor categories: (1) (major) additions to existing features that encourage curation by directly affecting how users interact with OpenTree, (2) (major) improving the reliability and usability of OTT and the synthetic tree by algorithmic improvements and establishing connections to a wider set of web resources, and (3) (minor) improving the current software architecture to make the system more efficient and maintainable and to enable long-term sustainability.

4.1 Improving users' interactions with OpenTree.

The section outlines seven features which represent "low-hanging fruit" by making OpenTree more useful without involving major coding challenges. The first will be completed using current OpenTree funds; the others would be funded by this award.

4.1.1 Tree illustrator tool In the Fall of 2017, we will integrate a tree illustrator tool (source code currently available in one of Open Tree's GitHub repositories) into the curator app. This tool will enable users to produce publication-quality tree images from the curator app as a means of motivating users to add their trees to phylesystem as they write their papers. Users who do not want to share their trees pre-publication will be able to use the tool without saving their data to phylesystem. Adding support for private data (section 4.1.3) will provide these users with another way to use the tree illustrator before publishing their data.

4.1.2 Improvements to curator UI Several features of the current infrastructure are implemented but not easily discoverable to users. A better high level design of the curator app would alert curators to any issues that block a study from contributing to the synthetic tree and would lead curators to solutions. We have already implemented conflict analyses that can provide evidence that an input tree is rooted incorrectly or has tips that are mis-mapped, but these analyses are hard for most curators to find. Our recent re-implementation of the phylesystem indexing process into a tool called `otindex` means that we can expose new search functionality in the curation UI. For example, one can now filter through the list of phylesystem studies by a particular taxon, curator, or tree size.

Feedback from curators has led to the design of a new interface that consists of a series of panels with user prompts at each stage. This interface needs to be implemented, and the existing interface will then be made available as an "expert" mode for curators who do not need step-by-step help. We anticipate devoting some project resources to the curation app through the duration of the project, in order to efficiently handle the frequent feature requests from users.

4.1.3 Private data "shards" of the phylesystem to expose OpenTree services As mentioned above, participants in our "clade workshops" expressed a desire to visualize conflict between their unpublished trees and the artifacts (e.g. previously published trees, the taxonomy and the synthetic tree) managed by OpenTree. To some extent, this can be accomplished by client-side scripting to make use of the OpenTree APIs. However, many systematists would prefer a graphical interface.

Maximizing the usefulness of OpenTree resources requires that the unpublished studies have their taxa mapped to OTT. We have a user-friendly curator app to support this operation, but that web application uses an open data store (phylesystem). We propose enabling private data by saving phylogenetic data to a private GitHub repository rather than our public phylesystem GitHub repository. We can re-use much of the same infrastructure for private data that we already do for public data. The software that manages phylesystem already supports data being spread across multiple shards – each corresponding to a different git repository. Because the git operations used by the server are agnostic to the permissions on the cloud-based server, most of the server's tasks can be made private merely by changing the configuration of the repository (*i.e.* without code changes). Some programming will be required to allow users to manage their set of studies in the private repositories. For example, we will add functionality to assign each private study with a 'secret' URL that can be shared with collaborators (similar to the link used to share pre-publication Dryad or TreeBASE data with reviewers). We will also modify the `otindex` tool that makes phylesystem information searchable so that it does not expose private data when users search for studies in the curator app.

Adopting this route for enabling pre-publication analyses using OpenTree has several advantages: (1) continued development of analytical and visualization tools for curation of phylesystem will automatically benefit the private analyses; (2) the storage format of the data will be identical to the phylesystem so it will be easy for a researcher to make their data public once a study is published; (3) using a private repository requires minimal changes to the existing architecture; and (4) this same mechanism of privacy will provide an avenue for other tools, like CIPRES, to let their users work use the OpenTree system before those users have published their data.

4.1.4 Custom synthesis controlled by users Researchers who participated in our "clade workshops" wanted to be able to have rapid feedback about the effect that their newly added trees would have on their clade of interest in the synthetic tree. To do this we can perform custom runs of synthesis on subsets of taxa. The smaller scale of these custom runs would make it easier to get feedback from taxonomic experts because these trees can be examined using a wide variety of tools (we do not know of any desktop tree visualization tool that will handle the full synthetic tree of life). Our recent migration to a more efficient `propinquity +otcetera` synthesis procedure makes it feasible to generate a synthetic tree in a few minutes. Currently, there are a few manual steps in the process of deploying the tree to the servers in the Amazon Web Services (AWS) cloud to be served via the Tree of Life APIs. While OpenTree makes old copies of the synthetic tree available for download, only one version is served via the API at any point in time.

We propose to make it easy for users to supply a list of trees and trigger the creation and deployment of a custom created version of the synthetic tree built on-demand. The `propinquity` synthesis pipeline contains some run-time options (for instance flagging properties of dubious taxa to prune from OTT); so some user interface is required to gather the desired settings from the user. The anticipated schedule for developing this feature is: 1. write a web service that provides a user interface and triggers the creation of an annotated supertree on demand but for download only (i.e. not served via the APIs); 2. finish the scripting of the deployment of a new tree so that no manual intervention is needed; 3. extend the server-side Tree of Life code to allow the server to store and serve multiple trees which are distinguished by IDs; 4. refactor the on-demand synthesis implementation so that the custom trees are browsable by the user who triggered the synthesis run.

4.1.5 Better user engagement via improvements to the homepage for each user Activity in the curator app is associated with a unique login name (currently we use GitHub logins). Similarly, the web forms that enable user feedback to our synthetic tree or OTT require users to be logged in. We use GitHub APIs to create GitHub issues with the user name of the user. This allows for public activity of our users to be tracked using GitHub's APIs and to enable users to receive notifications of responses to their issues via GitHub's notification features.

A commonly requested feature is the idea of a taxon-specific notification of OpenTree activity for each authenticated user. By storing the IDs of taxa of interest to a user, OpenTree could notify users when new trees that are relevant to them are uploaded, when the synthetic tree changes for their taxa of interest, or when conflict reports for their studies change. Implementation of this feature will require write access to a user's profile information, but the private data features implemented for the pre-publication curation will make it easy to store user info in a private repo. The software adaptors that we use to persist information in git repositories are all implemented in the general purpose `peyotl` library.

Better developed user-profile pages will also allow the project to credit users not only for their curation and activity in feedback issues, but will allow us to credit authors of papers that contribute to the synthetic tree when those authors are users of OpenTree. Because the synthetic tree is already annotated with lists of studies that support and conflict with each grouping in the tree, it will be feasible to report statistics such as "number of branches in the synthetic tree that are the result of curation and/or publications by this user." Consolidating this information into "leader boards" for different statistics could even help the project "gamify" some aspects of data curation.

4.1.6 Gap analysis to increase phylogenetic coverage Currently, only a small fraction of the tips in the synthetic tree are represented in one of the phylogenetic inputs – most tips are placed based on the taxonomy only. Tools from `otcetera` annotate the full tree with statements indicating which input trees support (or conflict) with each grouping in the synthetic tree. These tools can be easily extended to provide reports on phylogenetic coverage of different clades throughout the tree.

These reports will allow us to highlight "most wanted" taxa in the curator app and the synthetic tree browser, and to provide general coverage statistics about the tree of life. The reports can also be used to identify and highlight studies in the curator app that could be most useful if they were curated to the level necessary for inclusion into the synthesis pipeline. The automated phylogentic updating described below will highlight regions of the tree for which sequence data is available, but published phylogenetic studies have not been completed or are not available.

4.1.7 Faster inclusion of trees from phylesystem into the synthetic tree Curators frequently ask when their newly added trees will be reflected in the synthetic tree. Thus, simply increasing the frequency with which the tree is updated will help user engagement. The move to `propinquity` +`otcetera` to build the tree has made the synthetic tree easier to build and compare to previous trees. The project has released 9 official versions of the tree, with 3 being released in the last 12 months. Replacing the remaining manual steps of the synthesis deployment pipeline is a task that will enable us to release more frequently and will help us implement "custom synthesis" feature described above.

In addition to scripting all aspects of building the new tree, more frequent updates will require bolstering our system for testing synthetic trees. Currently OpenTree maintains a list of taxa that are expected to be monophyletic (a "monophyly test suite"), and has code to identify differences between successive versions of the tree. Experience has shown that a poorly curated tree (e.g. incorrect rooting) can result in dramatically less resolution in the tree or increased computational time. We also have a tree-conflict API that allows us to detect trees that strongly conflict with the taxonomy or previous tree. Integrating these conflict tools into our testing regime will allow us to flag studies that should be examined (or re-examined) by an expert before they are included as inputs to synthesis. Currently trees have to be manually flagged as inputs to synthesis, but improved testing may allow us to move to a system in which trees are automatically included in synthesis if they pass minimal tests. This change would mean that frequent updates to the synthetic tree would have a more visible effect (because the set of inputs is growing more rapidly).

4.2 New algorithms to improve the primary products of OpenTree We also envision four large scale changes to the current OpenTree system: (1) expanding the tools that build the synthetic tree to allow fossil taxa to be included; (2) a major expansion of the interoperability of our process for building OTT, (3) incorporation of gene-tree estimates produced by updating trees with newly available sequence data, and (4) adding estimates of node ages and branch lengths to the synthetic tree.

4.2.1 Permitting fossil and incompletely placed taxa into the synthetic tree This goal will be implemented in the final year of the current funding; it is discussed here because it is a prerequisite of the dating analyses mentioned below. Currently fossil taxa and taxa flagged as being incompletely placed (*incertae sedis*) are pruned from input trees and the taxonomy before creation of the synthetic tree. These taxa require the implementation of logic that allows a taxon to be infiltrated by "floating" taxa without invalidating the definition of the taxon. MTH and Dr. Ben Redelings (postdoc of MTH) extended the `propinquity` pipeline to allow *incertae sedis* and fossil taxa to be included. The algorithm is currently being tested, and we anticipate that the next release of the tree in the fall of 2017 will include these taxa. This will increase the number of tips in the synthetic tree by several thousand. The change in taxonomic placement between the "raw" OTT and the pre-processed form used in the latter parts of the synthesis pipeline will be recorded in a format that is visible to our proposed system of taxonomic amendments (see below).

4.2.2 Richer interactions between users, OTT, and external taxonomic projects A major goal of our proposed work is to expand the capabilities of our user feedback infrastructure to collect semantically rich taxonomic feedback, to use this feedback to improve OTT, and to explain OTT to users and other taxonomy-oriented bioinformatics projects. These changes will improve OTT and make it easier for others to understand how the taxonomy was built. The most common form of feedback by far (58% via online reporting tools, and a large fraction of comments from users at workshops) pertains to OTT. Taxonomic problems are a significant roadblock to data curators and users of the synthetic tree.

Construction of a taxonomy was actually not one of the original goals of OpenTree, but we found that existing taxonomic resources were either closed, not comprehensive enough, or were focused on matching as many names as possible without paying sufficient attention to the hierarchical classification. Thus, we built a tool (`smasher`) to combine several taxonomies into OTT [22]. The fully automated nature of `smasher` has been a boon to identifying the source of taxonomic problems that have been pointed out by users, and the tool does let us build user-created taxonomic amendments so that errors in source taxonomies can be overridden. However, the diagnosis of the source of the problems and the authoring of many of the taxonomic amendments require manual work by an expert in the software.

In the summer of 2016, OpenTree deployed a user-friendly method for allowing data curators to add a taxon to OTT. This solves a common problem in which recently published papers (often papers that are the basis of new classifications) use new species names that are not in any source taxonomy (and thus not in OTT). We will expand the user-interface of our online tool for gathering feedback so that our software will write a wider range of taxonomic amendments.

Currently our feedback system prompts the user for free-text feedback and creates a GitHub issue (in our feedback repository) with a structured text format. The feedback tool populates the comment with a URL that identifies the location in the synthetic tree that the user is reacting to. The issue is associated with the user's GitHub profile to allow conversation about the issue to take place via the OpenTree tree browser or GitHub interface. A trivial extension is to deploy this feedback tool on the taxonomy browser. It is also vital that we channel expert taxonomic feedback through an interface that will capture the information semantically. Taxonomic feedback can vary from comments about the form/spelling of a name, pointers to evidence that the name is a synonym or homonym, or comments about the content of a taxon (*i.e.* comments about the "taxon concept" being used). Each of these forms of commentary could be incorporated during the next iteration of OTT, but having an expert software developer reformulate a free text comment into a machine-operable taxonomic amendment clearly does not scale well to a large user base.

The OTT-building tool `smasher` currently can read a series of taxonomic amendments from experts and give those statements high weight over competing information when building OTT. It stores, for each taxon in OTT, a list of identifiers for that taxon in input taxonomies. It also writes log files documenting its decisions during a run, however these logs are only useful to developers who understand how `smasher` works. The second phase of our extension of `smasher` will be to refactor the code so that the decisions made during the combination of taxonomies are documented in a machine readable form. We already perform a similar set of operations in the `propinquity` pipeline that builds the synthetic tree; after the processing steps are completed a set of JSON files are written that describe the actions taken at each step. This makes it relatively easy to diagnose the cause of surprising groupings. We will migrate taxonomic logic from `smasher` into a `propinquity +otcetera` pipeline, and write files that explain the taxonomic alignment. This system will use the set of taxonomic amendments, but will also elicit richer feedback from users. Currently, the taxonomy browser used in OpenTree is crude and only shows the final output (the taxa and their corresponding IDs in other resources). By serving the "decision" files from an OTT run in the taxonomy browser, we will make it feasible for taxonomic experts to make more targeted, specific comments about how the system is failing.

By standardizing both our taxonomic amendment format and the logs of our software's decisions, we will make it easier for the projects that control the input taxonomies (e.g. NCBI, and GBIF) to benefit from OpenTree curation. It will be easy to filter the list of amendments and decisions that are disagreeing with each input taxonomy, and write software that will transform those OpenTree artifacts into corrections usable by the source taxonomic providers. We hope that increased communication between OpenTree and its taxonomic sources will lead to a shared system for discussing taxonomy, and allow OpenTree to have a less central role in constructing a comprehensive taxonomy. We note that our current architecture, which relies on structured GitHub issues, is a good start toward a shared forum for taxonomic information, because OpenTree does not "own" the user information and all of the feedback issues are publicly visible and writable to anyone with a GitHub account. In the second year of the grant we will hold a workshop that will try to bring together interested participants from other bioinformatics projects with the goal of standardizing a format for expressing corrections to taxonomies (the "taxonomic amendments" mentioned above). The "nomen" ontology developed by the species file group is one example of a structured format for discussing names (but not taxonomic concepts). The EulerX project and the PhyloRef project are funded to build tools around the meaning of taxonomic and phylogenetic names respectively. We will adopt any standardized formats that we can agree on to make the taxonomic expertise shared with OpenTree as widely useful and interoperable as possible.

4.2.3 Automated phylogenetic updating In the current tree, many species are represented only by taxonomy even though sequence data is available for them in GenBank. In some cases this disconnect between sequencing and phylogenetic information is due to those taxa not having been included in phylogenetic analyses. In others, these taxa have been included in trees, but those phylogenetic analyses are not digitally available. With the context of the existing trees deposited in the Open Tree database there is an opportunity to provide active, automated extension of the tree of life with new inferences made directly from sequence data. We have developed a software pipeline, Physcraper, (available through EJM's public repositories on GitHub) which uses trees and their associated metadata from the Open Tree database. The approach builds on previous work on automatically updating phylogenies with new sequence data [23, 24, 25], and the tool is integrated with the OpenTree corpus of studies and OTT. Physcraper automates the updating of phylogenetic relationships by combining these trees, the sequence alignments used to generate them, and newly available sequence data. The procedure consists of querying sequence databases, aligning sequences and inferring their phylogenetic relationships. This tree-updating procedure is currently functional as a free-standing analysis tool. Although these updated phylogenetic analyses have not been subject to the level of scrutiny of published, peer reviewed phylogenetic estimates, they can provide some evolutionary context for taxa which would otherwise only be placed by taxonomy. These automated updates will be made available through OpenTree, but clearly sequestered from the curated published phylogenies. By integrating visualization of regions of the tree where there is sequence data available but not included in digitally available phylogenetic estimates, we will be able to highlight opportunities for rapid phylogenetic progress. Using the conflict and concordance tools available in the web browser, it will be possible to visualize where these new trees refine, agree with, or conflict with taxonomy and previous phylogenetic inference. These areas of conflict may represent regions in which taxonomy and phylogenetic inferences need to be updated in the face of new data. Alternatively, conflict may be driven by labeling or metadata errors in the queried sequence data bases. By integrating this information derived from automated updating with OpenTree, we will also provide users with notifications when sequences related to their taxa of interest become available. Automating the first steps of an expanded analysis will point practitioners to areas of the tree where data is available and further investigation is warranted. In regions where automated estimates do not disagree with existing taxonomic or phylogenetic data, but refine relationships, theses phylogenies can provided better phylogenetic information than that currently available. Integrating automated tree updating will require a new phylesystem datastore shard to hold the updated phylogenies, and an alternative synthetic tree including these unpublished phylogenetic estimates.

4.2.4 Branch lengths / node ages A major research use of phylogenies is comparative biology - inferring evolutionary and ecological processes across species. Most comparative methods require some estimate of branch length in terms of time or character change. Thus, producing a synthetic tree tied to the timescale of evolution would dramatically increase the usability of the synthetic tree.

The phylesystem corpus of published trees has incomplete and heterogeneous data on branch lengths. Some trees contain no branch length information. Others contain branch lengths from different gene regions, and others divergence times and branch lengths measured in years.

EJM and MTH have developed FastDate (in collaboration with the lab of Dr. Alexandros Stamatakis; manuscript in prep) - a very fast approach to estimating credible intervals for ages from sequence data and fossil calibrations. We will couple this tool with the GenBank-scraping tools mentioned in the phylogenetic updating section to produce large matrices for the estimation of dates on large parts of the synthetic tree. This will also necessitate integration of OpenTree with databases of fossil dates (paleodb and the fossil calibrations database). We can also make use of tools being developed in Arbor to interpolate dates for nodes that are not represented in the sequence data matrices, for example by combining branch length estimates with divergence times [26]

4.3 Improvements to infrastructure and tooling The final category of proposed work is oriented around making the infrastructure of OpenTree more sustainable and robust. The project was originally funded as a high-risk project to build a supertree on a scale much larger than had ever been attempted. OpenTree uses a modular approach in which different components interact interact through the same common set of APIs that we publish to external users, allowing us to update components without disrupting other users or other parts of the architecture. The current architecture was developed across four different labs and ten developers, with a high level of independence between the development of various components, which encouraged both rapid development and testing of new technologies. The downside of this approach is multiple codebases, using different programming languages and frameworks. Making OpenTree software easy to deploy, develop, and maintain is one of the keys to long term sustainability. Simplification of the deployment, number of technologies, code repositories, and programming languages involved will reduce hosting costs, make it easier for new developers to join the project, and simplify maintenance of the code long term.

4.3.1 Simplify code and reduce number of technologies As we design and implement the new features already described in this proposal, we aim to: (a) reduce the number of production code repositories from 18 to 10 by consolidating functionality for building and serving the tree and taxonomy; (b) port software that uses web2py[27] to unify around a single Python framework, Pyramid [28]; (c) remove our dependencies on maven [29], Redis [30], Java [31], RabbitMQ [32], and neo4j [33]. .

Our current use of neo4j databases for the tree and taxonomy browsing services necessitates the use of "large" AWS instances, and introduces dependencies on Java and maven. Given that we are not dependent on unique features of neo4j, eliminating neo4j will allow us to deploy our servers on smaller, cheaper virtual machines in AWS and will make it easier for developers to quality check our results and code. The increased efficiency will save the project direct costs spent on AWS but will also make the Software-as-a-Service model discussed in our sustainability plan much cheaper and more attractive to users.

We have already begun the process of simplifying and consolidating code. The curation API has been rewritten to use the Pyramids rather than web2py framework, that new implementation is currently being tested and is expected to be deployed in the Fall of 2017. The tree API and basic parts of the taxonomy API were recently reimplemented in `propinquity`. Those implementations will also be tested in the Fall of 2017. Once the "fuzzy-matching" functionality of the taxonomy API's is implemented, then we will be able to dramatically reduce the RAM requirements of our servers by deprecating the current implementations of the tree and taxonomy APIs. We propose to implement the fuzzy taxonomic matching functionality in the `otindex` tool because there are readily available tools for indexing PostgreSQL for fuzzy matching, and that tool currently uses PostgreSQL to store the taxonomy.

4.3.2 Deployment using Ansible The current deployment process uses a series of custom bash scripts that install components on AWS servers. We propose moving our deployment to Ansible [34] playbooks because: 1. Ansible modules make it easier to guarantee idempotency, 2. errors and status reports will be much easier to find and debug, and 3. the terser, standardized syntax enabled by a tool designed for deployments will make the system much more transparent to new developers. The newly developed `otindex` software uses Ansible as a test case.

4.3.3 Improved testing and documentation OpenTree software currently has over 7000 lines of code devoted to testing (both unit-testing and integration testing), and new feature development will expand that. Richer sets of tests make it easier for developers to find bugs and problems. We use the "coverage" Python package to assess the fraction of code that is exercised by unit tests (coverage is currently ≈50% for `peyotl`, for example). We will raise the coverage proportion, and conduct analyses to verify that the untested code is non-essential. OpenTree already uses TravisCI [35] for continuous testing of branches of code that are pushed to GitHub. As part of our richer testing, we will expand the set of tests run by Travis with each push. This strategy mitigates the annoyance of having an extensive, slow-to-execute set of tests while assuring that developers do not introduce regression bugs.

5 Workplan

Figure 3 shows a graphical depiction of the timeline of the significant tasks to be performed color coded by the person expected to perform them (Jim Allman is the contractor who has written most of the JavaScript parts of OpenTree software; we anticipate that he will be one of the two developers hired on a contract basis). The figure also has a table that shows how the tasks relate to the deliverables described in the text. The timeline of tasks depicts developers as working on only one task for months. We certainly acknowledge that, in actuality, the work will be collaborative and iterative. A bug report or feature request may require revisiting a task that was already "completed" according to the timeline. However, the main goal of the figure is to ground the approximate delivery dates for the products of the grant and convey that the staffing level requested is appropriate. The workshops planned (see below) are not depicted on the timeline, but will be crucial in gathering user feedback to help refine the software.

5.1 Management Plan See Figure 3 for the approximate timeline of each task. EJM will manage the integration and testing of Physcraper, the work on adding dates and branch lengths to the tree, the undergraduates working on the project at UC Merced, and the software development workshops (described in the Broader Impacts section). MTH will manage other parts of the grant, including the work of the KU postdoc and the 2 contractors. Dr. Ben Redelings is likely to continue his role as the KU postdoc. We use weekly video conferences to stay in sync with respect to major goals and decisions, our gitter chat channel for day-to-day communication, and GitHub issues to communicate about specific bug reports or feature requests. In the first five years of the OpenTree project, these communication channels have proven more successful than our email lists or our use of Trello project management software.

6 Dissemination Plan

6.1 Release schedule and dissemination We will continue our practice of making source code available under free and open source license during development, followed by journal publications to increase visibility. phylesystem and the curator app [18], the project's original synthesis pipeline [36], and the tree browser [1] were all deployed and then discussed in publications. A publication on `peyotl` is planned for submission in the winter of 2017. In addition to deploying and using code pre-publication, the project has also presented on the tools at Evolution meetings, iEvoBio meetings, and meetings of the Society of Systematic Biologists. We plan to continue our participation in these meetings and to continue our joint hosting of hackathons or workshops with other interested projects (to date we have hosted events with NESCent's HIP group, FuturePhy, and Arbor).

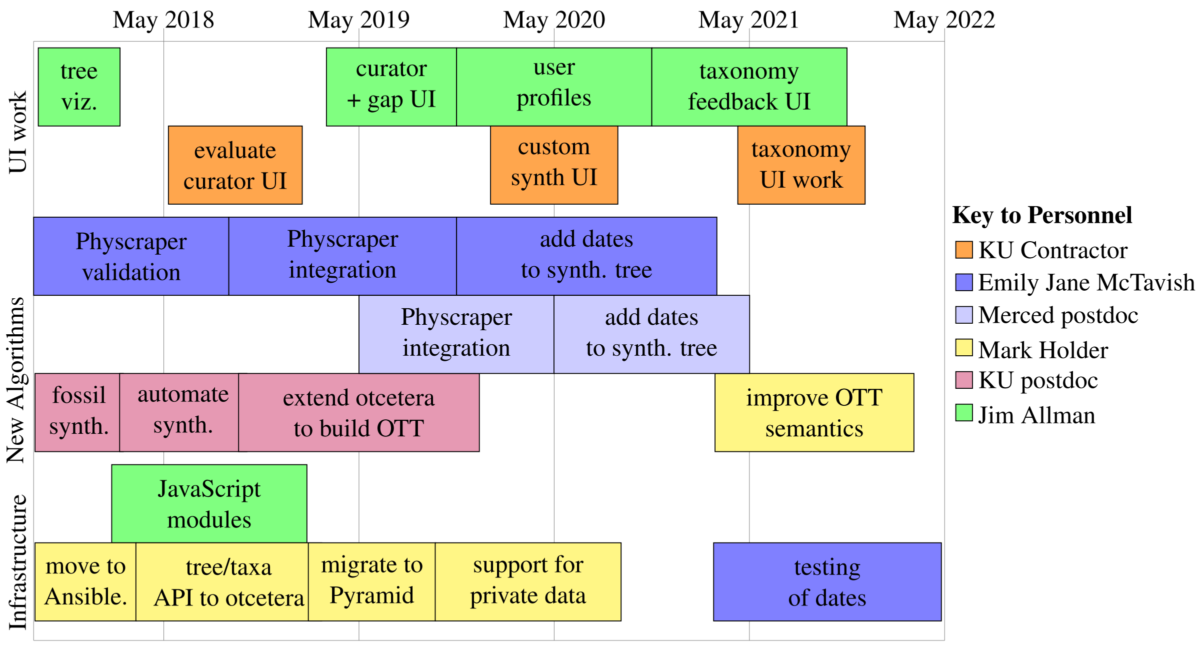
As discussed above, a major goal of this proposal is to increase the rate of releasing versions of the synthetic tree and OTT to bring the project much closer to a "live" representation of phylogenetic knowledge. We version OTT and the synthetic tree separately, and maintain a progress page to help users understand how the information on the site changes.

6.2 Intellectual property All OpenTree software is released under free and open-source licenses (GPL-2.0 if required by dependencies; BSD 2-clause otherwise). All data artifacts produced by the project (the synthetic tree and OTT) are released with Creative Commons Zero (CC0) copyright waiver. Our curation interface and phylesystem data store have fields for storing the license information associated with published trees that are uploaded to phylesystem. We encourage authors depositing to the phylesystem to use the CC0 waiver.

7 Plan for user engagement

The new and improved OpenTree features described will be successful only if communicated well, and informed by feedback from the various user communities. OpenTree uses a combination of social media, mailing lists, chat rooms, GitHub issues and wikis, and workshops to engage the community.

We have 4,000 followers of the project's Twitter account, and 1,650 "likes" on Facebook. We also have a public mailing list (134 members) for comments or questions, have most of our project conversation in a publicly accessible chat room (29 members), and all notes and documentation in public Google Docs or GitHub wikis.



Goal	Delivery date	Task(s)
4.1.1 Tree illustrator	Before grant	tree viz.
4.1.2 Improve Curator UI	Summer 2019	“evaluate curator UI” and “curator + gap UI”
4.1.3 Private data	Summer 2020	support for private data
4.1.4 Custom synth	Summer 2020	ansible, automate synth, custom synth UI
4.1.5 User homepages	Summer 2020	user profiles
4.1.6 Highlight gaps	Fall 2019	curator + gap UI
4.1.7 Frequent Synth.	Summer 2018	automate synth., increase synth. freq.
4.2.1 Include fossils	Before grant	fossil synth.
4.2.2 Collab. taxonomy	Spring 2021	extend otc. to build OTT, improve OTT semantics, taxonomy feedback UI, taxonomy UI work
4.2.3 Auto-update trees	Spring 2020	Physcraper validation, Physcraper integration
4.2.4 Dates/branch lengths	Spring 2021	add dates to synth. tree, testing of dates
4.3.1(a) Fewer repos	Spring 2020	otc. to build OTT, tree/taxa API to otc.
4.3.1(b) Avoid web2py	Spring 2019	migrate to Pyramid
4.3.1(c) Avoid Java	Fall 2019	oct. to build OTT, tree/taxa API to otc.
4.3.1(d) Fewer dependencies	Throughout	previous Java tasks, migrate to Pyramid
4.3.1(e) Lower memory needs	Fall 2019	tree/taxa API to otc.
4.3.2 Ansible deploy.	before grant	move to Ansible
4.3.3 Better testing	Throughout	JS modules, testing of dates, most infrastructure tasks

Figure 3: Top: an approximate timeline of major tasks color coded by personnel. Bottom: a table that relates the deliverables described in the text to the tasks shown in the timeline.

The synthetic tree browser allows users to comment on any node in the tree, automatically generating a GitHub issue for the comment. This provides a low activation threshold for users, while keeping comments organized. Work to expand this feature (see section 4.2.2) will better capture feedback from expert users in a machine-readable, semantically-rich framework, allowing us to incorporate such feedback much more quickly and share with other projects.

Direct engagement of the systematics community is essential to continued success and development of the project. We have used multiple types of workshops to disseminate these tools to the user community, most in collaboration with the NSF funded FuturePhy and Arbor projects. These included tutorials at the SSB meeting, online webinars and clade workshops. We have included funds for two additional workshops, which will target clades under-represented in the synthetic tree.

8 Sustainability Plan

KAC and MTH participated in a workshop at NSF (April, 2016) on sustaining infrastructure projects, completing an exercise lead by ESA, based on their Sustaining Biological Infrastructure course. We have identified two promising avenues for generating sustainable funding for OpenTree without obtaining further research grants: (1) subscription-based access to lower-latency and faster web-services, and (2) deploying and maintaining versions of the software as a service (SaaS). We note that neither of these strategies places any OpenTree data behind paywalls, but instead provide additional services on a cost-recovery basis.

OpenTree subscription model: A subscription-based plan would entail allowing free access with a governor that increases latency to unsubscribed domains that make high use of the OpenTree services. We will extend our web-services to make it possible to log usage by domain and introduce limits on total amount of data downloaded via the APIs by users from unsubscribed institutions. The TAIR project has been successful pricing subscriptions tailored to the usage by an institution and getting buy-in from companies and libraries [37]. Note that all of the artifacts of the project will still be accessible via links to archives or git repositories. So this route will not put data behind a paywall, rather it will attempt to recover the costs of running the web services.

OpenTree Software as a service: As mentioned above, this proposal would fund extensions to the current architecture to enable "on-demand" synthesis, and provide private sandboxes for groups of users. All of the software needed to use this system will be open source, but many labs may prefer to sub-contract to OpenTree developers to deploy and maintain instances of these sandboxing systems rather than install the tools themselves. Applying a small surcharge to the service can help fund extended maintenance of the OpenTree platform.

We will use external resources in development of sustainability plans. EJM and MTH will attend the ESA Sustaining Biological Infrastructure course in order to further develop and obtain feedback on sustainability. The UCMerced Office of Business Development has developed a 'Venture Lab' program that focuses on helping academics develop their ideas into commercially viable businesses. The Venture Lab is working with OpenTree to develop potential long term sustainability plans. If appropriate, based feedback from these collaborations, we will seek funds through programs such as SBIR or I-corps.

Cutting the base operating costs is important for sustainability. We have recently started discussion with CyVerse about using the CyVerse Atmosphere instead of AWS for our development/testing servers. Our technical requirements are well within the capabilities offered. As discussed in section 4.3.1 we anticipate that the move away from neo4j will dramatically reduce AWS costs associated with the production servers.

9 Broader Impacts of the Proposed Work

OpenTree is already providing a valuable resource to the broader biological community. Researchers can access and download the evolutionary relationships for any set of taxa of interest from the synthetic tree.

The phylesystem database provides open access to curated and reusable phylogenetic data, simplifying data sharing in phylogenetics. The services OpenTree provides to the biological research community are discussed in depth in the 'User communities' section. We are also providing easy web access to estimates of evolutionary relationships to non-academic users. The improvements described in this development grant will allow us to better serve users based on feedback we have received on the project thus far.

In addition, we are well-poised to use the OpenTree platform to teach computational skills to biologists. MTH and EJM have experience teaching basic computational skills in Software Carpentry (SWC) workshops. SWC is a volunteer organization whose goal is to make scientists more productive, and their work more reliable, by teaching them basic computing skills through a collaborative curriculum development process. We will organize and hold two participant workshops focused on training systematists on use of tools to access the OpenTree web resources in a programmatic manner. Based on experience from previous participant workshops, the community of systematists could greatly benefit from some basic applied training in use of command line interfaces, scripting and API's. We will use teaching techniques developed by Software Carpentry to present this material, and develop an OpenTree focused curriculum. We will use pre- and post- workshop assessments to determine the effectiveness of different aspects of these participant workshops, and to improve our approach from the first workshop to the second.

The University of California, Merced (UCM) is a minority serving institution. The majority of undergraduate students are from minority ethnic groups (87%), speak a language other than English at home (68.4%), are first-generation college students (67.3%), and come from low income families (Pell grant recipients, 61%). Biology is the largest undergraduate major at UCM. While women and minorities are underrepresented in the sciences in general, that underrepresentation is particularly stark in the computational sciences. By introducing programming in a biological context, we will have the opportunity to reach students who may be intimidated by pursuing computational science, or have not been otherwise exposed to it. EJM will develop and teach a semester long computational skills and data management class directed at upper level undergraduates with backgrounds in biology. The budget at UCM includes four years of funding for undergraduate assistants. Involving undergraduates from UCM in OpenTree will provide research experience and computational skill development for students from groups underrepresented in academic biology and computation. Two undergraduate students at UCM from underrepresented groups have already completed summer research projects contributing to and applying OpenTree. EJM will be involved in the UCM SACNAS (an organization dedicated to fostering the success of Chicano/Hispanic and Native American scientists) chapter to recruit undergraduate researchers, as well as to identify further opportunities for undergraduate researchers.

In the third year of the grant, MTH will develop a course at KU that is focused on Python programming for upper level undergraduate and graduate students in biology. The course will make extensive use of OpenTree APIs to teach web programming and will use online taxonomic resources as parsing challenges for the students. Because of KU's Biodiversity Institute and KU's Natural History Museum, there are a very large number of graduate students working in systematics (approximately 30 in most years). So using examples from systematics is a natural fit for such a course, and many of the students will be able to provide curation and taxonomic feedback to the project. All course materials will be made available under open source licenses or the CC0 waiver.

The "References Cited" section contains the citations to published works, but also has URLs for links to external projects mentioned in the proposal.

References

1. Cody E Hinchliff, Stephen A Smith, James F Allman, J Gordon Burleigh, Ruchi Chaudhary, Lyndon M Coghill, Keith A Crandall, Jiabin Deng, Bryan T Drew, Romina Gazis, Karl Gude, David S. Hibbett, Laura A. Katz, Dail H Laughinghouse IV, Emily Jane McTavish, Peter E. Midford, Christopher L. Owen, Richard H. Ree, Jonathan A. Rees, Douglas E. Soltis, Tiffani Williams and Karen A. Cranston "Synthesis of phylogeny and taxonomy into a comprehensive tree of life". *Proceedings of the National Academy of Sciences* 112.41 (2015), pp. 12764–12769. DOI: 10.1073/pnas.1423041112. URL: http://www.pnas.org/content/112/41/12764.
2. *Tree of Life AMA in Reddit AskScience AMA Series*. 2015. URL: https://www.reddit.com/r/askscience/comments/3lxde9.
3. Bryan T. Drew "Data deposition: Missing data mean holes in tree of life". *Nature* 493.7432 (Jan. 2013), pp. 305–305. ISSN: 0028-0836. DOI: 10.1038/493305f. URL: http://www.nature.com/nature/journal/v493/n7432/full/493305f.html.
4. Bryan T. Drew, Romina Gazis, Patricia Cabezas, Kristen S. Swithers, Jiabin Deng, Roseana Rodriguez, Laura A. Katz, Keith A. Crandall, David S. Hibbett and Douglas E. Soltis "Lost Branches on the Tree of Life". *PLOS Biol* 11.9 (Sept. 2013), e1001636. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1001636. URL: http://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.1001636.
5. Arlin Stoltzfus, Brian O'Meara, Jamie Whitacre, Ross Mounce, Emily L. Gillespie, Sudhir Kumar, Dan F. Rosauer and Rutger A. Vos "Sharing and re-use of phylogenetic trees (and associated data) to facilitate synthesis". *BMC Research Notes* 5 (2012), p. 574. ISSN: 1756-0500. DOI: 10.1186/1756-0500-5-574. URL: http://bmcresnotes.biomedcentral.com/articles/10.1186/1756-0500-5-574.
6. Dominique G. Roche, Robert Lanfear, Sandra A. Binning, Tonya M. Haff, Lisa E. Schwanz, Kristal E. Cain, Hanna Kokko, Michael D. Jennions and Loeske E. B. Kruuk "Troubleshooting Public Data Archiving: Suggestions to Increase Participation". *PLOS Biol* 12.1 (Jan. 2014), e1001779. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1001779. URL: http://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.1001779.
7. Andrew F. Magee, Michael R. May and Brian R. Moore "The Dawn of Open Access to Phylogenetic Data". *PLoS ONE* 9.10 (Oct. 2014), pp. 1–10. DOI: 10.1371/journal.pone.0110268. URL: http://dx.doi.org/10.1371\%2Fjournal.pone.0110268.
8. *Phylotastic project*. 2013. URL: http://phylotastic.org/.
9. Richard Dawkins and Yan Wong *The ancestor's tale: a pilgrimage to the dawn of evolution*. 2nd ed. Second Mariner Books, 2016. ISBN: 978-0-544-85993-7.
10. Luke J Harmon, Jeffrey Baumes, Charles Hughes, Jorge Soberón, Chelsea D Specht, Wesley Turner, Curtis Lisle and Robert W Thacker "Arbor: comparative analysis workflows for the tree of life". *PLOS Currents Tree of Life* (2013). DOI: 10.1371/currents.tol.099161de5eabdee073fd3d21a44518dc. URL: http://currents.plos.org/treeoflife/article/arbor-comparative-analysis-workflows-for-the-tree-of-life/.

11. François Michonneau, Joseph W. Brown and David J. Winter "rotl: an R package to interact with the Open Tree of Life data". *Methods in Ecology and Evolution* (2016). DOI: 10.1111/2041-210X.12593. URL: http://onlinelibrary.wiley.com/doi/10.1111/2041-210X.12593/abstract.

12. Guangchuang Yu *dlstats: Download Stats of R Packages*. R package version 0.1.0. 2017. URL: https://CRAN.R-project.org/package=dlstats.

13. Wayne P Maddison and David R Maddison "Mesquite: a modular system for evolutionary analysis." (2001). URL: http://mesquiteproject.wikispaces.com/.

14. Mark A Miller, Wayne Pfeiffer and Terri Schwartz "Creating the CIPRES Science Gateway for inference of large phylogenetic trees". In: *Gateway Computing Environments Workshop (GCE), 2010*. IEEE. 2010, pp. 1–8.

15. Nico M Franz, Naomi M Pier, Deeann M Reeder, Mingmin Chen, Shizhuo Yu, Parisa Kianmajd, Shawn Bowers and Bertram Ludäscher "Two Influential Primate Classifications Logically Aligned". *Systematic Biology* (2016). DOI: 10.1093/sysbio/syw023. URL: http://sysbio.oxfordjournals.org/content/early/2016/03/22/sysbio.syw023.abstract.

16. *PhyloRef project*. 2015. URL: http://www.phyloref.org/.

17. *TaxonWorks project*. 2016. URL: http://taxonworks.org/.

18. Emily Jane McTavish, Cody E Hinchliff, James F Allman, Joseph W Brown, Karen A Cranston, Mark T Holder, Jonathan A Rees and Stephen A Smith "Phylesystem: A git-based data store for community-curated phylogenetic estimates". *Bioinformatics* (2015), btv276. DOI: 10.1093/bioinformatics/btv276. URL: http://bioinformatics.oxfordjournals.org/content/early/2015/05/30/bioinformatics.btv276.full.

19. Emily Jane McTavish, Mike Steel and Mark T Holder "Twisted trees and inconsistency of tree estimation when gaps are treated as missing data–The impact of model mis-specification in distance corrections". *Molecular phylogenetics and evolution* 93 (2015), pp. 289–295. DOI: 10.1016/j.ympev.2015.07.027. URL: http://www.sciencedirect.com/science/article/pii/S1055790315002316.

20. Benjamin D. Redelings and Mark T. Holder "A supertree pipeline for summarizing phylogenetic and taxonomic information for millions of species". *PeerJ* 5 (Mar. 2017), e3058. ISSN: 2167-8359. DOI: 10.7717/peerj.3058. URL: https://doi.org/10.7717/peerj.3058.

21. Mark T. Holder and Emily Jane McTavish "Phylogenetic tree comparison". In: *Encyclopedia of Evolution, Volume 3*. Ed. by R. Kliman. 2016, pp. 277–283. ISBN: 9780128004265. DOI: 10.1016/B978-0-12-800049-6.00213-4. URL: http://store.elsevier.com/product.jsp?lid=0&iid=5&sid=0&isbn=9780128004265.

22. Jonathan A Rees and Karen Cranston "Automated assembly of a reference taxonomy for phylogenetic data synthesis". *Biodiversity Data Journal* 5 (2017).

23. F. Izquierdo-Carrasco, J. Cazes, S. A. Smith and A. Stamatakis "PUmPER: Phylogenies Updated Perpetually". *Bioinformatics* 30.10 (2014), pp. 1476–7. DOI: 10.1093/bioinformatics/btu053. URL: http://bioinformatics.oxfordjournals.org/content/early/2014/01/28/bioinformatics.btu053.

24. Alexandre Antonelli, Hannes Hettling, Fabien L. Condamine, Karin Vos, R. Henrik Nilsson, Michael J. Sanderson, Hervé Sauquet, Ruud Scharn, Daniele Silvestro, Mats Töpel, Christine D. Bacon, Bengt Oxelman and Rutger A. Vos "Towards a Self-Updating Platform for Estimating Rates of Speciation and Migration, Ages, and Relationships of Taxa (SUPERSMART)". *Systematic Biology* (2016). DOI: 10.1093/sysbio/syw066. URL: https://academic.oup.com/sysbio/article/doi/10.1093/sysbio/syw066/2418028/Towards-a-Self-Updating-Platform-for-Estimating.

25. Michael J. Sanderson, Darren Boss, Duhong Chen, Karen A. Cranston and Andre Wehe "The PhyLoTA Browser: Processing GenBank for Molecular Phylogenetics Research". *Systematic Biology* 57.3 (June 2008), pp. 335–346. ISSN: 1063-5157, 1076-836X. DOI: 10.1080/10635150802158688. URL: http://sysbio.oxfordjournals.org.www2.lib.ku.edu/content/57/3/335 (visited on 07/10/2015).

26. Graham J. Slater and Luke J. Harmon "Unifying fossils and phylogenies for comparative analyses of diversification and trait evolution". *Methods in Ecology and Evolution* 4.8 (2013), pp. 699–702. ISSN: 2041-210X. DOI: 10.1111/2041-210X.12091. URL: http://dx.doi.org/10.1111/2041-210X.12091.

27. Massimo (lead developer) Di Pierro and other contributors to the open source project *Web2py Web Framework*. 2007. URL: http://www.web2py.com/.

28. *Pyramid web framework*. 2010. URL: http://www.pylonsproject.org/.

29. *Apache maven build tool*. 2004. URL: https://maven.apache.org/.

30. *Redis in-memory data store*. 2009. URL: http://redis.io/.

31. *Java programming language*. 2014. URL: https://java.com/.

32. *RabbitMQ message queuing*. 2007. URL: https://www.rabbitmq.com/.

33. *neo4j graph database management system*. 2007. URL: https://neo4j.com/.

34. *Ansible*. 2012. URL: https://www.ansible.com/.

35. *Travis CI continuous integration service*. 2010. URL: https://travis-ci.org/.

36. Stephen A Smith, Joseph W Brown and Cody E Hinchliff "Analyzing and synthesizing phylogenies using tree alignment graphs". *PLOS Comput Biol* 9.9 (2013), e1003223. URL: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003223.

37. Leonore Reiser, Tanya Z Berardini, Donghui Li, Robert Muller, Emily M Strait, Qian Li, Yarik Mezheritsky, Andrey Vetushko and Eva Huala "Sustainable funding for biocuration: The Arabidopsis Information Resource (TAIR) as a case study of a subscription-based funding model". *Database* 2016 (2016), baw018. DOI: 10.1093/database/baw018. URL: http://database.oxfordjournals.org/content/2016/baw018.