## Predicting the Optimal Computing Platform for Climate-Driven Ecological Forecasting Models

### Scott Sherwin Farley

#### Master’s Thesis

#### Advisor: John W. Williams

# Introduction

Global environmental change, including climate warming, anthropogenic land use, and the spread of invasive species, threatens to severely alter biodiversity patterns worldwide in the coming century (Lowe et al., 2011; Root & MacMynowski, 2005; Thuiller, 2007; Thuiller et al., 2008a). Habitat loss and fragmentation and loss of climatically suitable areas are expected to result in the extinction of nearly one-third of all species, in addition to large-scale biotic reorganizations (Thomas, 2010). Climate has been shown to exert significant control over species distributions, particularly over those of vascular plants (Salisbury, 1926; Woodward, 1987). Statistical methods that quantify species responses to climatic gradients can be used to inform ecologists and conservation managers of future biotic assemblages under different warming scenarios (Clark, Gelfand, Woodall, & Zhu, 2014; Guisan & Thuiller, 2005a; Guisan & ZImmerman, 2000; Guisan et al., 2013; Maguire et al., 2015a; Thuiller et al., 2008a).

Despite the potential for widespread, irreversible changes to the Earth’s biosphere (Barnosky et al., 2012), a growing volume of ecological data puts global change researchers in a position to forecast and mitigate catastrophic changes. Environmental monitoring efforts, such as the Long Term Ecological Research program (LTER, (Hobbie, Carpenter, Grimm, Gosz, & Seastedt, 2003)), National Ecological Observatory Network (NEON, (Schimel, Keller, Duffy, Alves, & Aulenbach, 2009)), community curated databases, like the Neotoma Paleoecological Database (http://neotomadb.org) and the Paleobiology Database (PBDB, http://paleobiodb.org), and modern biodiversity occurrence databases, such as the Global Biodiversity Information Facility (GBIF, http://www.gbif.org), form a network of informatics infrastructure for supporting global environmental change research. New cyberinfrastructures for ecoinformatics provide a tremendous amount of information to researchers attempting understand and forecast responses to perturbations in the earth system (Brewer, Jackson, & Williams, 2012; Michener & Jones, 2012). Paleoenvironmental proxy data, including fossil pollen, macrofossils, and freshwater and marine diatoms, can be an important supplement to modern biodiversity data. When forecasting to future climate spaces, paleoecolgical proxies can enhance understanding of bioclimatic gradients not found on Earth today (Veloz et al., 2012). Understanding responses to these vanished climates can improve prediction of distributions under no modern analog climates likely to emerge in the near future(Williams & Jackson, 2007). However, though collections of modern and historical biodiversity data offer enormous potential for forecasting studies, these data can be challenging to work into workflows because of their volume, uncertainty, and heterogeneity (Hampton et al., 2013).

Statistical methods can be used to mine datasets for new insight, extract parameters, and simulation of future scenarios, though these techniques can be computationally demanding, particularly on large datasets (Huang et al., 2013). Many contemporary Big Data applications, such as the popular microblogging service Twitter (http://twitter.com), require distributed, parallel, streaming methods to identify key analytical trends in real time (Bifet, Holmes, Pfahringer, & Gavalda, 2011). With the recent growth in monitoring and occurrence data, some ecological analyses now apply sophisticated machine-learning techniques to ecological forecasting problems and have seen significant increases in predictive skill (Elith et al., 2006a). With over 1 billion occurrence records in Neotoma and GBIF, traditional statistical methods for analyzing and forecasting ecological processes cannot be applied without compromising analysis scope. With so much data, ecologists will need to adopt techniques already common in fields like bioinformatics (Schatz, Langmead, & Salzberg, 2010), geonomics (Stein, 2010), climate analytics (Schnase et al., 2014), and private industry (Mosco, 2014), including distributed processing, ensemble methods, and cloud computing.

Climate-driven ecological forecasting models, variously called ecological niche models (Peterson, 2003), predictive habitat distribution models (Guisan & ZImmerman, 2000), and species distribution models (SDMs)(Guisan & Thuiller, 2005b), have seen extensive application in ecology, global change biology, evolutionary biogeography(Thuiller et al. (2008a), (Araújo, Whittaker, Ladle, & Erhard, 2005)), reserve selection (Guisan et al., 2013), and invasive species management (Ficetola, Thuiller, & Miaud, 2007). SDMs characterize a species’ biospatial responses to environmental gradients (Franklin, 2010), and use these response to forecast potential future distributions under climate change scenarios. SDMs apply supervised statistical learning techniques to relate a species distributional data with its environmental covariates. As more occurrence data becomes available through cyberinfrastructures like GBIF and Neotoma, SDMs can be fit on more data, yielding more detail climatic response surfaces. Along with the increased occurrence records, many scholars are now applying computationally intensive modeling approaches, including nonparametric data-driven techniques (Elith et al., 2006a) and Bayesian methods that rely on repeated sampling of full joint probability distributions (Clark et al., 2014; Dawson et al., 2016). Most leading SDM methods, despite being popular in the literature, are not scalable to very large datasets because they are not designed to take advantage of parallel processing or distributed computing. As the volume of ecological data increases and the need for high resolution, accurate projections of biotic distributions becomes more pressing, reducing project scope (e.g., (Bolker et al., 2009a)) or modifying model parameter sets can no longer be considered a valid option.

Cloud computing offers a technological solution to some of the problems posed by the increasing Bigness of ecological data (Hampton et al., 2013; Michener & Jones, 2012). Cloud computing is a architectural design pattern that enable “ubiquitous, convenient, and on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort” (Hassan, 2011; Mell & Grance, 2012; Vaquero, Rodero-Merino, Caceres, & Lindner, 2008). With the rapid commercialization of cloud computing and the widespread availability of public cloud providers like Amazon Web Services (AWS) and the Google Cloud Compute Engine (GCE), scientists have, a seemingly unlimited supply of infitely configurable computing resources at their disposal. The Cloud has been touted by many of the largest players in Silicon Valley, and credited with Obama’s 2012 presidential election win, Netflix’s ability to provide streaming entertainment to millions of consumers, and Amazon’s massive success in online retailing ((Mosco, 2014)). In 2010, the U.S. Federal Government has embraced the efficiency, speed, and economy of the cloud, requiring all federal agencies to adopt a “Cloud-First” policy when considering new information technology (IT) developments (Kundra, 2010). Accordingly, the National Aeronautics and Space Administration (NASA) and the National Science Foundation (NSF) have both officially endorsed cloud technology (Mosco, 2014). Moreover, researchers across the sciences have posited that the cloud is the key to solving future computing and modeling challenges (Hsu, Lin, Ouyang, & Guo, 2013; Yang et al., 2011).

Although the cloud offers a promising technological solution to the data demands of ecological forecasting, there are few guiding principles on when the benefits, in reduced computing time, outweigh the financial costs of a cloud-based solution. The cloud introduces a novel expense model for computing: by partitioning large physical resources into smaller virtual machines, cloud providers can charge consumers for the use of computing resources, rather than their purchase (Hassan, 2011). Under this new operational expense model, consumers are no longer tied to a single hardware configuration, rather they can scale their virtual instances up or down depending on computational demand (Armbrust, Fox, Griffith, Joseph, & Katz, 2009). Despite being more complex to set up and maintain, Cloud-based solutions are more complex to set up and maintain, they can offer significant advantages over traditional desktop computing. While the complexity costs of engineering and implementing a cloud based solution are difficult to estimate, the computational time gains achieved by running models on high performance virtual instances can be measured empirically and combined with cost estimates to provide guidance on when a cloud based solution would be economically rational.

In this thesis, I develop a theoretical framework to predict the optimal computing solution for a set of species distribution modeling activities. Treating workflow characteristics as discrete model parameters, I posit a theoretical predictive model for minimizing computational time and financial cost for model runs. I gather data on the runtimes of four different classes of species distribution models under different parameterizations and on different computing hardware. Using this empirical dataset, I fit a nonparameteric learning model to this data to determine the drivers of model runtime and predict the execution time of future modeling scenarios. My findings suggest that if SDMs, and ecology more generally, are to benefit from Cloud computing, future effort must be directed towards developing models that more explicitly take advantage of parallelism and distributed processing frameworks.

The remainder of this thesis proceeds as follows: in the following section I introduce my research questions, motivated by the tension between a potential decrease algorithm runtime and an associated increase in computational cost. Section 3 works with the “Four V’s Framework” of Big Data to justify ecological datasets as Big Data. Section 4 reviews relevant background literature on system benchmarking, runtime modeling, species distribution modeling, and cloud computing. In Section 5, I present a theoretical framework for predicting the optimal computing solution for a given ecological forecasting modeling. Section 6 describes the methodology used to collect runtime and accuracy data on four different SDM instances. Section 7 presents the results of the experiments and comments on their implications. Conclusions and future work is discussed in Section 8.

1. Research Questions

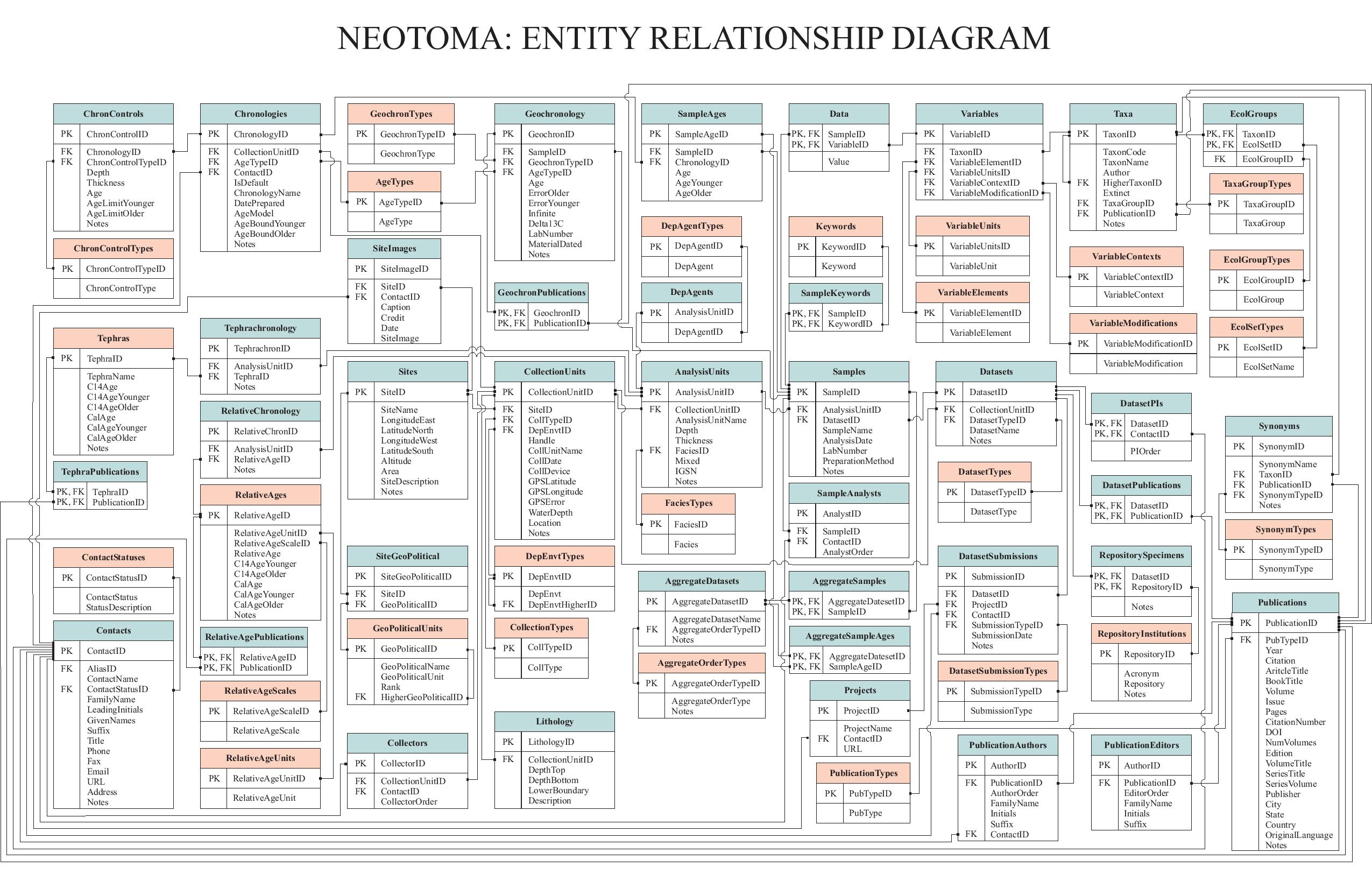
* Might need to revise?

1. To what degree can the runtime of climate-driven ecological forecasting models be predicted?
   1. Can a predictive model out-predict a null model that suggests that all researchers utilize a single desktop computer for all modeling activities?
2. Can an optimal solution for a given modeling workflow be predicted using workflow characteristics?
   1. If so, what are the most influential workflow characteristics in making this decision?
   2. Do contemporary published studies vary in the characteristics that matter most in execution time?
3. What modeling scenarios are best suited for transition to the cloud, if any?
4. Justify ecological data as big data The vast expansion of data the sciences has necessitated the development of revolutionary measures for data management, analysis, and accessibility (Schaeffer, Pierre, Twigger, White, & Rhee, 2008). Worldwide data volume doubled nine times between 2006 and 2011, and successive doubling has continued into this decade (Chen, Mao, & Liu, 2014). With the large influx of massive geonomic sequences, long term environmental monitoring projects, phylogenetic histories, and biodiversity occurrence data, robust, expressive and quantitative methods are essential to the future the field (Schaeffer et al., 2008). As datasets become larger, significant challenges are encountered, including inability to move datasets across networks, necessity of high performance and high throughput computing techniques, increased metadata requirements for storage and data discovery, and the need for greater agility to respond to new and previously unsupported uses for the data in data access and analysis applications (Schnase et al., 2014).

Ecological occurrence data are records of presence, absence, or abundance or individuals of a species, clade or higher taxonomic grouping that are key to biodiversity analyses, ecological hypothesis testing, and global change research. These data are increasingly being stored in dedicated, large community-curated databases like the Neotoma Paleoecological Database, the Global Biodiversity Information Facility (GBIF), and the Paleobiology Database (PBDB). Since the early 1990s, the internet and associated information technology and an increased willingness to share primary data between scientists precipitated rapid influxes of enormous numbers of digital occurrence records. While there are known problems with the quality and consistency of data records in large occurrence databases (Soberón, Arriaga, & Lara, 2002), they provide a low-friction way to consume large amounts of data that would otherwise be prohibitively time consuming to derive from the literature or in the field (Beck, Böller, Erhardt, & Schwanghart, 2014), (Grimm et al., 2013)]. Entire new fields, namely ‘Biodiversity Informatics’ (Soberon & Peterson, 2004), ‘Ecoinformatics’ (Michener & Jones, 2012), and ‘Paleoecoinformatics’ (Brewer et al., 2012) have been developed and delineated to address the growing challenges and opporunties presented by the management, exploration, analysis and interpretation of primary data regarding life, particularly at the species level now presented to ecological researchers (Soberon & Peterson, 2004).

The term Big Data is typically used to describe very large datasets, whose volume is often accompanied by lack of structure and a need for real-time analysis. Big Data, while posing significant management and analysis challenges, brings the opportunity for discovering new insights to difficult problems (Chen et al., 2014). Though the precise definition of Big Data is loose, there are two prominent frameworks for discriminating Big Data from traditional data. One characterizes Big Data as “a loosely defined term used to describe data sets so large and complex that they become awkward to work with using standard statistical software” (Snijders, Matzat, & Reips, 2012). This ambiguous delineation is echoed in the advertising and marketing literature that accompanies products like cloud computing that facilitate Big Data analysis. For example, Apache Hadoop, a popular distributed computing framework, has described Big Data as “datasets which could not be captured, managed, and processed by general computers within an acceptable scope” (Chen et al., 2014).

Under this framework, the Bigness of the data is specific to both the time of analysis and the entity attempting to analyze it. (Manyika, Chui, Brown, Bughin, & Dobbs, 2015) suggest that the volume of data required to be Big can change over time, and may grow with time or as technology advances. Furthermore, the criteria for what constitutes Big Data can vary between problem domains (Chen et al., 2014), the size of datasets common in a particularly industry and the kinds of software tools that are commonly used in that industry (Manyika et al., 2015). The Big Data label is most often applied to datasets between several terabytes and several petabytes (2^40 to 2^50 bytes). However, because of ecology’s lack of experience with massive datasets and limited analysis software common in the discipline, ecological occurrence data can be shown to fall under the pretense of Big Data.

The recent development of complex relational databases that store spatiotemporal occurrence records and their metadata demonstrate that traditional methods of data handling were not sufficient for modern ecological occurrence data. While the datasets are not particularly large in storage volume, they are composed of millions of heterogenous records with complex linkages. Consider the complexity of the relationships between different data records, for example. Figure {X} shows the Neotoma relational table structure, and the complicated web of relationships between each entity. Now, consider keeping track each attribute for hundreds of millions of records, and synchronizing datasets among thousands of independent researchers, and it becomes clear why dedicated databases like GBIF and Neotoma have been developed. Further developments, like application programming interfaces and language specific bindings, supplement the tasks of accessing, filtering and working with the large occurrence datasets (Goring et al., 2015). While occurrence data does not require the disk space of popular commercial applications like Twitter and Youtube, it has clearly demonstrated a need for new, custom built tools to store, analyze, and use large numbers of records. 

A second important framework by which to assess Big Data is the ‘Four V’s Framework’. First introduced by IBM, it was used by large technological companies in the early 2000’s to characterize their data, it is now a popular and flexible framework under which to describe Big Data. Under this framework, a dataset’s Bigness is described by its Volume, Variety, Veracity, and Velocity. (Yang & Huang, 2013) describe this framework, suggesting that “volume refers to the size of the data; velocity indicates that big data are sensitive to time, variety means big data comprise various types of data with complicated relationships, and veracity indicates the trustworthiness of the data” (Yang & Huang, 2013) p.276. Using this framework as a rubric, it can be shown that ecological occurrence demonstrates three of the four V’s.

Since the late 1990s, the scale of biodiversity information alone has become challenging to manage. Figures [X] and [X] track the growth in collections of Neotoma and GBIF through time. In 1990, only 2 of the records now stored in Neotoma were in digitized collections. Today, there are over 14,000 datasets containing [XXX] individual occurrence records, and associated spatial, temporal, and taxonomic metadata, corresponding to an average growth rate of 1.4 records per day. Nearly all records in Neotoma are derived from sedimentary coring or macrofossil extraction efforts, data gathering techniques that require large expenditures of time and effort (Davis, 1963), (Glew, Smol, & Last, 2002). By number of records, GBIF’s collections are far larger than Neotoma’s, perhaps reflecting the lower degree of effort required to gather modern ecological occurrence data. GBIF houses digital records of well over 600 million observations, recorded specimens (both fossil and living), and occurrences noted in the scientific literature. Since its first operation in 2001, the facility’s holdings have grown nearly 300%, from about 180 million records in 2001 to approximately 614 million records in 2016, demonstrating the data’s volume. Note that GBIF’s reliance on literature and museum specimens allow its holdings to extend beyond its origin in 2001.

The second characteristic of Big Data in the four V’s framework is the Variety of the data, and its ‘various types with complicated relationships’ (Yang & Huang, 2013). Biodiversity data is highly diverse with many very complicated relationships and interrelationships. Neotoma’s holdings feature 23 dataset categories, including X-ray fluorescence (XRF) and isotopic measurements, macro fossils or verebrates and plants, modern and fossil pollen records, and freshwater diatom and water chemistry series. Similarly, in GBIF, there are 9 distinct record types, including human observations, living and fossil specimens, literature review, and machine measurements. Though the records coexist in large biodiversity database, they are distinctly different, derived using different protocols by different researchers.

The data’s spatial and temporal nature causes complex interrelationships between data entities. All of Neotom’s records and 87.6% of GBIF’s records are georeferenced to specific places on the earth’s surface. The spatial information in these databases is supplemented by other fields that describe the location of the observation, such as depositional setting, lake area, and site altitude, to improve contextual interpretation of occurrence data. Managing data with a spatial component is nearly always more challenging than managing data without it. Representations must grapple with data that is a discrete representation of a continuous physical phenomenon, correlations between parameters, space and time, and processes differ from each other in spatial and temporal scales(Yang, Wu, Huang, Li, & Li, 2011). Furthermore, occurrence data represents the work of many dispersed, individual researchers and research teams. The controlled vocabularies and organization of aggregating databases helps to efficiently assimilate large numbers of records, however, nearly every record was collected, analyzed and published by a different scientist. While some scientists have contributed many datasets to occurrence databases, most have only contributed one or two. The median number of datasets contributed to Neotoma is only 2 and the third quantile value is just 7 datasets. Each researcher is likely to use different equipment, employ different lab procedures, and utilize different documentation practices, contributing to a highly variable dataset.

Biodiversity data also has high levels of uncertainty associated with it – the third V in the Four V’s Framework. Some of the sources of uncertainty in the data, like spatial or temporal position uncertainty can be estimated (Wing, Eklund, & Kellogg, 2005) or modeled (Blaauw, 2010). Other sources of uncertainty have yet to be quantified, for example inter-researcher identification differences, measurement errors, and data lost in the transition from field to lab to database. A recent paper by the Paleon working group used expert elicitation to quantify the differences between the dates assigned to European settlement horizon, a process they argue varies between sites, and depends on the “temporal density of pollen samples, time-averaging of sediments, the rapidity of forest clearance and landscape transformation, the pollen representation of dominant trees, which can dampen or amplify the ragweed signal, and expert knowledge of the region and the late-Holocene history of the site.” The findings of this exercise suggest that paleoenvironmental inference from proxy data is highly variable between researchers. Moreover, some information will undoubtedly be lost in the process of going from a field site through a lab workflow to being aggregated in the dataset. Though some procedural information accompanies the data records, not all process details can be incorporated into database metadata fields, and probably more importantly, contextual details essential to proper interpretation of the data often gets lost on aggregation.

Both Neotoma and GBIF show high levels of quantifiable uncertainty, and are likely to show high levels of unquantifiable uncertainty as well. Of a random sample of 10,000 records of the genus *Picea* from GBIF, over half did not report spatial coordinate uncertainty. Of the 4,519 records that did, the average uncertainty was 305 meters, and the maximum was 1,970 meters. Clearly, such high levels of uncertainty might be problematic for modeling efforts (Beck et al., 2014). Neotoma records show a similar uncertainty in their temporal information. Neotoma records each have a minimum, maximum, and most likely age for each age control point (e.g., radiocarbon date). Out of a sample of 32,341 age controls in the database, only 5,722 reported any age uncertainty at all. The summary statistics for these age controls suggest that the median age model tie point has a temporal uncertainty of 260.0 years. The 25% percentile is an uncertainty of 137.5 years and the 75% 751.2 years, suggesting that dates are only identifiable down to ± 130 years of the actual date. [NEOTOMA UNCERTAINTY THROUGH TIME]. Considering sediment mixing, laboratory precision, and other processes at work this is a relatively minor uncertainty, but it certainly contributes to occurrence data’s lack of veracity.

The final piece of the Big Data framework is the dataset’s velocity, which characterizes the dataset’s sensitivity to time. High velocity data must be analyzed in real time as a stream to produce meaningful insights. Tweets, for example, are analyzed for trends as they are posted. User’s are drawn to participation in up-to-the minute discussion, and significant effort has been put towards sophisticated algorithms that can detect clusters and trends in real time (Kogan, 2014), (Bifet et al., 2011). The rate of increase in data volume in both Neotoma and GBIF is not fast enough to invalidate the results from previous analyses, suggesting that it’s velocity is not enough to warrant streaming Big Data techniques. Neotoma’s growth rate of approximately 1.4 new datasets each day (1990-2016 average) and GBIF’s daily growth rate of about 59,000 records (2000-2015 average) are small compared to the total number records in the database. Unlike in many private sector applications, there is little incentive to researchers to immediately analyze new biodiversity records, since all new findings will be reported on in the academic paper cycle, typically several months to years. Moreover, automated analyses of distributional data have been warned against, due to the overall poor data quality (Soberón et al., 2002) and high levels of uncertainty.

While not time sensitive, ecological occurrence data requires advanced, sophisticated techniques to store and analyze, and demonstrates high volume, low veracity, and significant variety, and should therefore fall under the auspices of Big Data. Remaining traditional techniques of occurrence data storage analysis are likely to be unsuitable in the coming years due to large annual data growth rates. Both GBIF and Neotoma are experiencing sustained and increasing growth since the early 1990s. To fully and accurately derive value from new data being added to distributional databases, new advanced techniques for modeling and analyzing this data are required. Many of the modeling algorithms used in ecological data analysis should be reevaluated to bring them into the world of big data and to take advantage of the advanced computational infrastructure now available.

#### Cloud Computing in the Sciences

To lessen the computational burden on big data, large enterprise technology companies have promoted cloud computing, a strategy of partitioning their own resources into small virtual servers they can rent out for a profit. Cloud computing is a privatized incarnation of grid computing, a standardization of requests for computer resources to leverage distributed networks of physical machines. As (Foster, Zhao, Raicu, & Lu, 2008) suggest, “Cloud Computing not only overlaps with Grid Computing, it is indeed evolved out of Grid Computing and relies on Grid Computing as its backbone and infrastructure support. The evolution has been a result of a shift in focus from an infrastructure that delivers storage and compute resources (such is the case in Grids) to one that is economy based aiming to deliver more abstract resources and services (such is the case in Clouds).” While there are technical differences as well, the business model between the two is a major differentiating point, as grid computing is more collective and project based that the utility-style introduced in Cloud computing (Foster, 2005). Perhaps because of this difference, the grid never achieved large-scale success in the private sector, they did succeed in forming massive federated systems providing computing power and data to scientists that still exist today, such as the Earth System Grid. However, in recent years, the Cloud – the ability to access a seemingly unlimited supply of computing and networking resources with just a credit card (Hassan, 2011) – has seen a rise in use in both the private and scientific spheres. Suddenly, scientists with little or computational infrastructure can have access to scalable and cost-effective computational resources(Hsu et al., 2013).

Major science-support organizations, including the NSF and NASA, have made major pushes to promote cloud computing in their own operation and the projects they support. Spurred by the U.S. Office of Management and Budget’s 2010 “25 Point Plan to Reform Federal Information Technology Management” (Kundra, 2010), federal agencies are now required to adopt a “Cloud First” policy when “contemplating IT purchases and evaluate secure, reliable, and cost-effective cloud computing alternatives when making new IT investments” (Nasa Report: REPORT NO. IG-13-021). Federal programs were implemented to help agencies adopt cloud technologies reducing the effort needed to screen cloud providers for data security policies and enable rapid and provision of cloud services (Kundra, 2010). NASA has demonstrated it a pioneer in the adoption of private clouds, having developed the Nebula private cloud infrastructure at Ames Research Center (Kundra, 2010). Open source private clouds are still a popular method for large academic organizations (Huang et al., 2013), of which there are several computing options. Cloud technology, both public and private, has been extensively lauded for its application to the bioinformatics and geonomics fields (Hsu et al., 2013, Issa et al. (2013); Stein, Isakov, Godowitch, & Draxler, 2007). Cloud based solutions for bioinformatics research relieve the large memory requirement often present in geonomics and drug-design data (Hsu et al., 2013). Biomedical specific applications are being developed to specifically support bioinformatics science, including reducing latency time between user input and analysis using streaming methods (Issa et al., 2013), the development of biology-application specific operating systems designed supporting for cloud-based protein analytics right out of the box(Kaján et al., 2013). Climate analytics and reanalysis are also demonstrative of fields adopting cloud computing technology early in its history. (Schnase et al., 2014) describes attempts to develop data-proximal analytics on massive climate datasets that eliminate the need for moving large datasets and take advantage of infrastructure scalability and virtualization.

Cloud computing has also been posited as the future of the geosciences. Yang et al (2011) note that despite advanced computing technologies, geoscientific problems are still limited by computational ability, including data volume bottlenecks, processing limitations, multi-user concurrency, and spatiotemporal velocity of data (Yang et al., 2011). Many scholars suggest that the cloud provides a means of overcoming some of these intensities by optimizing distributed computational resources without increasing the carbon footprint or financial budget (Yang et al., 2011). Public clouds like Amazon Web Services now support several large scale spatial data infrastructure (SDI) projects, though geospatial algorithms are more difficult to implement on public cloud infrastructure than on more traditional computing environments like grid computing (Huang, Yang, Nebert, Liu, & Wu, 2010). High performance cloud instances have demonstrated theoretical and practical uses in modeling dust storms in real time(C. Yang et al., 2011). The adoption of cloud technology, and distributed computing infrastructure more broadly (Yang & Raskin, 2009), has been posited by many to be an important new tool in enabling new approaches to new and existing scientific questions by contributing essential computing architectures and methods for solving data and computing intensive problems (C. Yang et al., 2011).

#### Species Distribution Models

Species Distribution Models (SDMs) quantify the relationships between a species and its environmental range determinants through statistical methods (Svenning, Fløjgaard, Marske, Nogués-Bravo, & Normand, 2011). While these models sometimes include mechanistic or process components, they most often refer to correlative models, after (Elith & Leathwick, 2009a). SDMs rely on ecological occurrence data to provide training data to which statistical learning procedures are applied to estimate the species-specific response to a particular environmental or climatic covariate. With the widespread availability of statistical software and machine learning code libraries, and increased availability of environmental and occurrence data, the utilization of this technique has grown substantially in recent years (Franklin, 2010; Svenning et al., 2011). SDMs are used in a variety of fields related to global change biology and have been shown to provide reliable estimates of climate-driven ecological change when compared to independent datasets. Figure [x] shows the dramatic increase of academic literature focusing on “topic=species distribution models” in Web of Science. The NSF estimated an annual 2.8% growth rate over all science and engineering citations between 2001 and 2011, while SDM literatures maintained a 9.6% growth rate over this period. Furthermore, in the last five years (2011-2016), the field has shown an annual average growth rate of 10.1%.

SDMs work by approximating the functional form of the species niche. Hutchinson ((Hutchinson, 1957)) characterized a species’ fundamental niche as an n-dimensional hypervolume that defines the environmental spaces where the intrinsic population growth rate of the species is positive (Williams & Jackson, 2007). The realized nice describes the subset of environmental space that the species acutally occupies at some point in time, and is smaller than the fundamental niche due to biotic interactions with other species. Most scholars argue that SDMs come close to approximating the species’ realized niche (Guisan & ZImmerman, 2000; Miller, Franklin, & ASPINALL, 2007; Soberón & Peterson, 2005), though the inclusion of fossil data in the model fitting process can increase the likelihood that calibration captures the fundamental niche (Veloz et al., 2012) and improve the assumption of niche conservatism (Thuiller et al., 2008b).

While SDMs are often used in the context of forecasting the effects of 21^st century ecological change, their calibration and application to paleogeographic problems can provide important commentary on their function and accuracy. The paleorecord provides a well-documented set of species occurrences and community responses to large, rapid, and/or persistent environmental changes and spatial extents ranging from local to global and at temporal resolutions ranging from subannual to millennial (Maguire et al., 2015b; Nogués-Bravo, 2009a). While niche models fit with paleodata face a number of additional challenges, often related to the data’s veracity, they have the potential to harness information provided by additional training data, mitigate the effect of *a priori* assumptions, and enable ecological hypothesis testing of the drivers of environmental ranges.

SDMs rely on three important assumptions, As a fundamental justification for applying predictions across space and time, all SDMs assume niche conservatism, i.e., that the niche of species remains constant across all spaces and times (Pearman, Guisan, Broennimann, & Randin, 2008). While the addition of paleodata to model fitting increases enlarges the modeled niche, niche adaptation, evolution and speciation are not modeled. (Peterson, Soberon, & Sánchez-Cordero, 1999) suggests that species typically demonstrate niche conservatism on multi-million year time scales. Second, SDMs rely on the assumption that species are at equilibrium with their environment, a phenomenon that occurs when a species occurs in all environmentally suitable areas while being absent from all unsuitable ones (Nogués-Bravo, 2009b). Given dispersal limitations and biotic interactions between species, this is rarely the case in practice. For example, many European species are still strongly limited by postglacial migrational lag (Svenning, Normand, & Skov, 2008).

Finally, SDMs must deal with extrapolation to novel and no-analog climates for which there is no training data. As inductive learning algorithms, SDMs are fitted with a set of labeled target examples to develop a mapping between the features of the examples and the output of the example. In this case, environmental covariates to species presence are used to learn species presence or abundance. Inductive learning is severely impacted when it is used to predict onto future examples that were not included in the set of training examples. Williams et al 2007 (Williams & Jackson, 2007) note the high likelihood of encountering novel and no-analog climates in the near future. Fitting the models with data from the paleorecord increases the likelihood that climatic assemblages will have been encountered by the learning algorithm during the fitting process. However, given rapid and highly uncertain climate change, the problem of projecting models onto unseen climates still exists.

Despite the strong assumptions that must be made, SDMs have been used for a wide variety of paleo and contemporary studies of geographic and environmental distribution. In the paleo domain, SDMs have been used to support hypotheses on the extinction of Eurasian megafauna (Nogués-Bravo, Rodríguez, Hortal, & Batra, 2008), identifying late-Pleistocene glacial refugia (Fløjgaard, Normand, & Skov, 2009; Keppel et al., 2011; Waltari et al., 2007), and to assess the effect of post-glacial distributional limitations and biodiversity changes (Svenning et al., 2008). SDMs are often combined with genetic, phylogeographic, and other methods to develop a complete assessment of a species biogeographical history (Fritz et al., 2013). [THIS PARAGRAPH IS IMPORTANT – CHANGE FOCUS TO IMPORTANCE OF ECOLOGICAL FORECASTING]

#### A Taxonomy of Species Distribution Models

SDMs can range from simple ‘boxcar’ algorithms that develop a ‘climate envelope’ for a species to a multivariate bayesian techniques that use Markov Chain Monte Carlo methods (MCMC) to develop probability distributions around projections. While all have the same fundamental goal of characterizing responses to climatic gradients, (Franklin, 2010) notes the multiple ways in which SDMs can be categorized. One conceptually meaningful way to group modeling algorithms is into data-driven, model-driven, and stochastic algorithms. The data-driven model-driven dichotomy is introduced in (Hastie, Tibshirani, & Friedman, 2009) and employed by (Franklin, 2010) in her text on SDMs. I add the burgeoning methods of stochastic, probability-based Bayesian methods to this taxonomy due to their recent uses and high accuracy. No individual method or class of methods has consistently outperformed any other (ARAUJO & NEW, 2007; Elith et al., 2006b; Veloz et al., 2012), though many scholars have attempted to assess interclass variation (Araújo & Guisan, 2006; Elith et al., 2006b), and variation between different parameterizations of the same model class (ARAUJO & NEW, 2007; Thuiller et al., 2008a; Veloz et al., 2012).

The goal in SDM is to use a learning algorithm and training examples to approximate the relationship between a set of inputs and outputs. The supervised learning paradigm relies observation of process to assemble labeled training examples or mappings between inputs and outputs where both values are known,

The learning algorithm approximates the real relationship by evaluating a loss function based on the difference . The observed inputs may be a p-dimensional vector of observed input features, $\boldmath{X} = x\_1, x\_2, ..., x\_p, p=1, 2, ..., P$. The resulting functional approximation then has a p-dimensional domain. Hastie et al (2009) argues for approaching supervised learning in terms of functional approximation, noting that it encourages the utilization of geometrical concepts of Euclidean spaces and mathematical concepts of probabilistic inference.

Model-driven, parametric, or statistical methods fit parametric statistical models to a dataset. Hastie et al (2009) suggests that these models demonstrate low bias but high variance, in other words, it relies heavily on *a priori* assumptions about the parametric form of the chosen model. These models were the first to see substantial use in SDM applications and have seen widespread continued use because of their strong statistical foundations and ability to realistically model ecological relationships (Austin, 2002). The earliest models were simple boxcar algorithms, fitting simple multidimensional bounding boxes around species presence in niche space (Guisan & ZImmerman, 2000). Other model-driven techniques include variants of logistic regression and generalized linear models on binary outputs (Franklin, 2010; Vincent & Haworth, 1983).

In terms of asymptotic complexity, parametric methods tend to be some of the least complex. Consider the generalized linear model, which generalizes simple linear models into their multivariate case, so that

$$\gamma\_i = \beta\_0 + \beta\_1\boldmath{x\_i} + ... + \beta\_n\boldmath{x\_n}$$

The user of this model must then specify a link function that describes how the mean of depends on the linear predictor, e.g., , as well as a variance function that describes how the variance of depends on the mean . Depending on the matrix decomposition method, the asymptotic runtime complexity of least squares is either operations or operations. Fitting a model with lasso regression also has this complexity (Hastie et al., 2009). Sure to converge?

The increase in available computing power has spurred the development and application of non-parametric, data-driven, machine learning modeling algorithms. These models, have, in some cases been shown to significantly out perform their model-driven counterparts (Elith et al., 2006b). These models demonstrate high bias but low variance, as they do not rely on any stringent assumptions about the underlying data, and can adapt to any situation, though any particular subregion of the model depends on a handful on input points, making them wiggly and highly sensitive to small changes in the input data. Data-driven algorithms include genetic algorithms (Elith et al., 2006b), regression trees (Elith, Leathwick, & Hastie, 2008), artificial neural networks (Hastie et al., 2009), support vector machines, and maximum entropy techniques (Elith et al., 2010). Since 2006, MaxEnt, a maximum entropy algorithm has seen widespread use and has demonstrated its ability to perform consistently even on small sample sizes (Phillips, Anderson, & Schapire, 2006). A review of recent SDM literature suggests that MaxEnt is the most popular SDM method in use today. Recent evaluations of MaxEnt, however, suggest that its performance, especially on small, presence only datasets, may be questionable when compared with other SDM algorithms (Fitzpatrick, Gotelli, & Ellison, 2013).

The asymptotic complexity of data-driven models tends to be larger than that of model-driven algorithms, because more passes over the data are typically required. To fit an additive models with -dimensional inputs and training examples, the total number of operations needed to fit the models is , where m is the number of applications of a smoothing method, typically less than 20 (Hastie et al., 2009). Support vector machines (SVMs) with m support vectors require . Multivariate adaptive regression splines (MARS) require operations to build an M term model. Building regression trees require operations, suggesting that in the worst cases, trees require operations. Random forests build Q full trees and average the results, making their complexity . Boosting, a technique that combines many weak learners into a committee ensemble also increases on the complexity of building a standard tree because it sequentially builds trees stagewise until a loss function has been minimized. Specific implementations of any specific algorithm depends on its implementation and language details. Not sure to converge?

Following a broader trend in modeling approaches across science, the third category of species distribution model utilizes Bayesian methods to develop probability densities and uncertainties with estimates of species presence. Advantages of Bayesian methods is the ability to include prior process knowledge and the ability to estimate model uncertainty without the need for bootstrapping procedures (Dormann et al., 2012; Elith & Leathwick, 2009b). Bayesian methods have become exceedingly popular in the last five years [cite? web of science], with the computational to efficiently sample from full joint probability distributions using MCMC. Several recent species distribution modeling efforts have attempted to incorporate these techniques (Hegel, Cushman, Evans, & Huettmann, 2010). (Golding & Purse, 2016) introduce species distribution models that incorporate Gaussian processes, highly effective statistical methods that demonstrate both high predictive accuracy and ecologically sound predictions. (Clark et al., 2014) describe a model that explicitly models the full joint distribution of species in an ecosystem, taking into account both climatic predictors and the effects of species competing in niche space. While Bayesian methods are on the forefront of SDM research, some teams are assembling and producing R packages designed to promote their use in SDM. (Hegel et al., 2010) notes the difficulties of transitioning from a frequentist view of ecological modeling to a Bayesian approach, and most models are carried out not in R, but in WinBugs or JAGS. The package ‘hSDM’ is a package developed for hierarchical Bayesian regression modeling that provides an easy to understand user interface in R for ecologists. While MCMC methods are computationally very expensive, numerical approximations and analytical solutions can, when available, significantly reduce computational burden (Golding & Purse, 2016).

While Bayesian methods offer several advantages over data or model driven SDMs, they are not yet mainstream, as a review of the contemporary literature exists. Of 100 randomly sampled papers from the most recent 4,000 citations in Web of Science that met the query ““(Species Distribution Model*) OR (Ecological Niche Model*) OR (Habitat Suitability Model\*).“, the overwhelming majority utilized techniques that fell within the second tier presented here. Out of 203 modeling methods described, 38 fell into the first category, 131 in the second, and 1 in the third. 33 additional experiments were not coded because they used unsupervised learning methods that don’t apply to the tiers described here. Figure [X] shows the breakdown of the reviewed modeling application by algorithm and how they fall into the tiers described in this section. In total, there were 47 different methods described, of which by far the most popular of which was MaxEnt (64 applications). The first tiers experiments were dominated by generalized linear model (15), but also included other variants of linear (2) and logistic regression (5). After MaxEnt in the second tier were generalized additive models (GAMs, 11) and genetic algorithms (11).

Because of the overwhelming propensity of scholars to employ methods in the second category, I focus my runtime analysis on this class of algorithms. Many authors have described the limitations imposed by computational complexity, though few have estimated those limits precisely. Elith et al (2006) recorded the execution time of the runs they used in their often-cited review of novel SDM techniques. They note execution times of several hours and up to several weeks for some modeling algorithms (Elith et al., 2006b). One of the main problems they note in their cursory analysis is the lack of ability to split SDMs across multiple cores. While their analysis was published in 2006 (with slower processors), this is still a problem with the state-of-the-art modeling algorithms. The longest-running algorithm they evaluate is the genetic algorithm for ruleset production (GARP) which they state takes 6 weeks to run on a single processor. Techniques is the most popular categories listed above also take significant time to execute, including boosted regression trees (80 hours), GAM (17 hours), generalized linear model (GLM), Maxent (2.75 hours). MARS was noted to take less than 15 minutes.

In some cases, practical papers advise against large modeling efforts due to computational limitations. For example, a 2009 review paper in *Trends in Ecology and Evolution* entitled “Generalized linear mixed models: a practical guide for ecology and evolution”, suggest that, when fitting a generalized linear mixed model (GLMM), if a user encounters insufficient computer memory or time limitations, the user should reduce model complexity, perhaps using a subset of the original dataset (Bolker et al., 2009b). Many review and method papers, while not encouraging users to reduce model scope, warn users of the computational expense of running SDMs. (Peterson, 2003) suggests “considerable computational capacity is necessary for the development of models even for a single species”. (Thuiller et al., 2008a) warns that “Limits to the broad application of this approach may be posed … by the computational challenges encountered in the statistical fitting of complex models.” While the increasing speed of consumer-grade computer processors are likely reducing computational limitations, the increase of data available in biodiversity data to use in model fitting is compounding the problem.

#### Algorithm Execution Time: Drivers and Measurement

Analyzing the constantly evolving and multifaceted dimensions of computer performance has posed problems for analysts and scholars since the advent of modern computing (Nordhaus & Yale University. Cowles Foundation for Research in Economics, 2001). While some figures, such as millions of floating point operations per seconds (megaFLOPS), are used in the literature, a computer’s performance depends on the way in which it is used (Lilja, 2009), rendering such universal metrics invalid. While it is difficult to effectively and meaningfully quantify computer performance, both empirical and theoretical methods of estimating algorithm runtime exist.

Theoretically, it is possible to determine the upper, lower, and average run times using asymptotic complexity analysis. In this exercise, the algorithm is assessed for its running time as its input is increased to infinity, so that only the order of growth is relevant(Knuth, 1976). Such complexity analysis is useful, because an algorithm that is more efficient asymptotically will be the best choice for all but very small inputs, in most cases (Cormen, 2009). The asymptotic complexity of an algorithm is most often used when determining how well an algorithms will scale to new input sets(Goldsmith, Aiken, & Wilkerson, 2007). Worst case run time, sometime referred to as Big-O notation, can usually be obtained by inspecting the structure of the algorithm and counting how many operations are required when n is sufficiently large. However, it is important to remember that, for any given n, the actual runtime will vary (Cormen, 2009). (Goldsmith et al., 2007) notes that many real-world programs contain large, interconnected data structures whose actual runtime will depend on the actual workload.

Empirical complexity studies have attempted to bridge the gap between asymptotic characterization and real-world workloads. These studies use empirical measurements of algorithm runtime under real workloads to build models to predict the execution time of future, unseen workload sizes and paramaterizations. These techniques seek to combine empirical measurements “with the generality of a big-O bound by measuring and statistically modelling [sic] the performance … across many workloads” (Goldsmith et al., 2007). Empirical complexity models have become an important subfield of artificial intelligence and have important applications to algorithm selection and automatic configuration of parameterized algorithms (Hutter, Xu, Hoos, & Leyton-Brown, 2014). (Brewer & Brewer, 1995) describes an perhaps the first attempt to develop a statistical model for the run and compile time of programs. (Fink, 1998) uses simple linear regression to develop a statistical relationship between problem size and computational time. (Hutter, Xu, et al., 2014) provides a comprehensive analysis of strategies and methods uses for empirical runtime models. (Hutter, Xu, et al., 2014) notes that parameterized algorithms can be treated the same as nonparametric algorithms by including model parameters in the execution time model, “notwithstanding the fact that they describe the algorithm rather than the problem instance, and hence are directly controllable by the experimenter”. (Hutter, Hoos, & Leyton-Brown, 2014) suggest that empirical performance models are essential in development of algorithm selection, which select the best algorithm for a problem, given some salient features of the problem. Nonlinear, tree based methods for empirical performance modeling were shown to be superior to other methods because of their ability to group similar inputs together and characteristic local-fitting, so that some large outliers do not interfere with the predictions of other groups (Hutter, Hoos, et al., 2014; Hutter, Xu, et al., 2014).

Concurrently running programs, operating system tasks, and other processes may affect the execution time of a real program at any point in time. Changes in dynamic system state are stochastic and can cause unpredictable, non-linear and non-additive changes in program runtime (Jones & Kalibera, 2013; Lilja, 2009). Random variation in system state makes deterministic statistical modeling of hardware influence on execution time difficult. These variations have been suggested to be a result of the way in which memory access patterns differ in space and time when small changes are made to the operating system state, timing device, or algorithm or its inputs (Lilja, 2009), and few attempts have been made to model them. (Jones & Kalibera, 2013) suggest that models based on benchmarked runtime may provide an accurate estimate of an upper bound (slowest time) of execution. Due to potentially large, nondeterminstic, system-induced variance in empirical results, it is important to perform the benchmarking experiment several times to estimate central tendency and analyze variance (Jones & Kalibera, 2013). (Dongarra, Martin, & Worlton, 1987) suggest that failure to properly characterize the workload, running ‘toy’ benchmarks too simplistic to provide real-world influence, or running benchmarks in inconsistent environments can lead to meaningless results.

Despite the challenges, several empirical runtime studies have attempted to model the affect of computer hardware on algorithm performance and achieved highly accurate results. Wu and Datla (2011) contend that the execution time of a program depends on the complexity of the algorithm and its input data, the static hardware configuration of the resource (e.g., amount and type of RAM, CPU clock rate), and the dynamic system state (e.g., number of processes competing for resources)(Wu & Datla, 2011). (Sadjadi et al., 2008) model execution time as a linear combination of contributions from elements of the computer’s hardware characteristics. Multivariate nonlinear systems may be more appropriate to capture some of the nonlinear changes possible in the empirical data, the approach taken in (Wu & Datla, 2011).

Computer hardware profiles can contain any number of elements, including processing and memory components, display and auxiliary peripheral devices, and networking capabilities. Previous attempts to model empirical runtimes have relied on a smaller subset of components thought to directly affect performance. (Sadjadi et al., 2008) include CPU rate (number of operations per seconds) and number of CPU cores, though other studies have also included memory amount and type, buffer size, and CPU cache size (Wu & Datla, 2011). While increasing the clock rate of a CPU is nearly guaranteed to improve execution time, since it can increase the number of operations able to be processed per second, number of CPU cores can also affect execution time by allowing multiple programs to execute in parallel. Algorithms must be specifically designed to run in parallel and the addition of processor cores will only improve performance up to a point, given by Amdhal’s law, after which all benefits of parallelism will have been reaped (Gustafson, 1988). If additional cores are available to a process able to take advantage of them, execution time can be reduced by offloading either other processes to a separate CPU core or splitting the algorithm workload across the processors. Computer memory can affect a program’s runtime by reducing the number of times a computer must retrieve data from the physical storage device (e.g., hard disk) and can improve the the amount of concurrent work able to be done on a machine (Wu & Datla, 2011).

#### Theoretical Problem Formulation

Here I present a theoretical framework for linking the goals of a model user to the time it takes to complete a modeling exercise and the computing resources required.

1. Consider a pool of computing resources, . As posited by (Wu & Datla, 2011), at any time , the effective processing power of is related to both the static and dynamic configuration of its hardware and software. Thus,
2. in which is the effective processing power, represents the static, hardware capabilities of the machine that do not change with time and represent the portions of the system that do vary with time. Execution times can vary non-deterministically with hardware due to stochastic changes in system state, so represents process uncertainty and unexplained variance.
3. Consumers of computing services are part of a market driven by supply and demand, and face a costs set by computing providers dictated by the effective computing power provided. Figure [X] demonstrates an instance of a Google’s cost surface as a function of memory and CPUs.
4. Now, formalize the goals of the model user. Every user of an application has a particular set of goals for using it in the first place (Norman, 1984). Scenario-based interface design informs our development of a finite set of use cases for a given application that fall within the bounds of existing or expected use (Carroll, 1999; Rosson, 2002). For example, consider a hypothetical scenario that could apply to a typical species distribution model user (adapted from (Smith et al., 2013)):

*Jessica Smith is a land manager at Yellowstone National Park, interested in understanding how Mountain Pine Beetle infestations may change under different anthropogenic climate change scenarios. Dr. Smith primarily wishes to characterize how the beetle range might change under the three different IPCC emissions pathways (Moss et al., 2010), rather than differences between algorithms or characterization of modeling uncertainty.*

From this brief scenario, we formalize Dr. Smith’s goals with the SDM applications. She wishes to model one species (*Dendroctonus ponderosae*), in a single area of known size (Yellowstone National Park, ~3,500 mi^2), under three climate scenarios. She requires only a single modeling algorithm.

Using brief user-based scenarios like this, we formalize the user, as a vector of characteristics that fully characterize the user’s goals in the scenario. The components of include a number of experiments to complex, as well as user traits such as experience with the model and familiarity with the interface employed, motivation, skill, and accuracy required. Associated with each user is a list of experiments to be completed that specify the number and character of modeling runs the users wants to do. Each element of the experiments list is a vector of model characteristics that contain enough information to specify the runtime behavior of a single model run, including modeling algorithm, spatial resolution, training examples, number of input predictor/covariate layers, and number of future time periods to project onto.

1. The time to compute a given algorithm with a dataset of some size is directly proportional to , the effective computing power, but several other terms can contribute to the total time spent by a user on a modeling task. The total time elapsed during a modeling experiment can be expressed as
2. In this formulation, represents the portion of time that is spent by user gathering the resources needed to model. In a species distribution modeling context, this term represents the time needed to find and download occurrence points and find and download predictor variables. can be thought of as a function of computing resources available to the user (how fast can data be downloaded?) and the experiment (what is the data?). is the time required by the modeler to prepare the data for entry into an algorithm. In this case, time might include data cleaning, projection, and conversion, as well as setting up and configuring computing platforms and environments, like R. This component can vary widely between modelers and between model applications, based on data source and quality, user skill and motivation, and the interface and equipment user has on hand. (Elith et al., 2006b) notes the potential impact of how experienced a user is with a model on the modeling time and results. includes the time it takes to return the output from the computation to the user, which may be non-trivial if the model is run on a set of remote resources and the output must be downloaded over a network to reach the client’s machine. Finally, represents the amount of time spent by the user evaluating model output and determining whether her goals were met during the modeling process.
3. Single experiments can be combined together to form workflows, so that a user’s time-to-goal for a workflow of modeling experiments can be expressed as
4. Combining equations from (2) and (5), we find the total time cost of a modeling experiment is the sum of total time of spent modeling and the total monetary cost is the hourly rate in dollars per hour for a given pool of computing resources. Thus, we derive a multivariate cost function for a modeling scenario that takes into account both time spent modeling and the cost of provisioning resources to do that modeling: $$C(U, H(t))= f(T\_g(U, H(t)), C\_Compute(H(t)))
5. Each user-based scenario will have its own cost curve. Taking one of these scenarios, specified by , and holding it constant, we obtain a unique cost function for this set of activities that depends only on the effective computing power. C is defined for all computing solutions, however, some may be suboptimal. We arrive at the optimal computing solution for this scenario by finding the multidimensional minimum of C that meets the following requirements:
   1. The accuracy of the model produced during the experiments is at or above a threshold inherent in the scenario,
   2. The total financial cost of the modeling experiment is at or below the user’s available funds, and
   3. the user obtained all desired information I suggest that this minimum point is the one that jointly minimizes the time spent modeling while simultaneously minimizing the cost of provisioning the resources. We can find this point by calculating the Euclidean distance between the origin and all candidate resources (the set C for a given user), and selecting the candidate with the smallest distance to the origin.

If multiple users are considered, it is possible to determine clusters of users and modeling activities that require similar computing resources. Furthermore, it may be possible to identify thresholds in cost (either financial or time) over which certain modeling experiments are not feasible. Similarly, technological breakpoints, such as the transition to cloud computing, may become clear when applying this framework.

### Methods

#### Approach

For the remainder of this thesis, I focus on the development of an empirical model for , because it seems to be the most tractable component of the framework, and its analysis can provide valuable insights to the drivers of model runtime, so that future model developers can promote development in the correct areas. My methods collect data from different potential user scenarios and use this dataset to develop a predictive model for any future input scenarios. The dataset is also interrogated for the drivers of computational runtime and accuracy within the SDMs. I finish by building calculating for different scenarios, though I do not include the terms other than , so the prediction is clearly not fully optimal.

I first developed a distributed network of computing nodes with access to a central database that was set up to run SDM runtime experiments on the Google Cloud Compute Engine. Using this infrastructure as a testbed, I recorded the runtime and accuracy of over 2,000 configurations of algorithm input and hardware. These experimental results were then tested using analysis of variance, partial dependency, and sensitivity testing methods to develop and understanding of the components of the models and their environments that contribute to increased runtime. The results were then used to develop a predictive model using regression trees. The predicted model was tested against independent testing sets to determine its accuracy in predicting future inputs. Finally, the model of computing time was combined with an estimate of computing cost, and the multidimensional minimum of cost and computing time was determined as the optimal computing platform for that SDM scenario.

#### Limitations

This methodological approach has several important limitations. The first and most important to the development of an optimal prediction is that the model does not take into account several of the most time consuming tasks of a SDM workflow, namely the data gathering and cleaning and interactive data analysis by the user . These terms in the workflow expression are highly variable, depend on many factors, and are nearly impossible to estimate. The two other terms, and are also excluded to further simplify the problem into a manageable scope. (C. Yang et al., 2011) suggest that download times can be accurately estimated as a function of the network speed and data size, so the addition of these terms would be a fairly easy extension to this work. However, these terms are relatively minor (< 25 seconds per experiment) and so their inclusion is unlikely to be influential in the model.

My analysis is limited to virtual instances hosted on Google Cloud Computing Engine (GCE), not real-work physical machines. This is both provides validity to my benchmarks by providing them with a consistent environment unaffected by other tasks or concurrent programs (Dongarra et al., 1987), and limits their interpretation as a proxy for real-world workflows. During a typical SDM workflow, a user is not limited to waiting for the model to finish computing, and can start other processes, which may take processor and memory capacity away from the SDM program. I suggest that ability to control operating system and background tasks is fundamental to correct interpretation of benchmark results, since it is difficult to characterize what the typical user would be doing while waiting for the model to finish computing. Thus, the use of dedicated instances provides robust results, but the results may be biased towards the fast end of what could reasonably be expected in the real world.

Another limitation of using the GCE is the inability to alter CPU clock rate as an experimental variable. The runtime of an algorithm is directly related to the number of cycles a processor can complete in a second. I ran my tests on a single processor speed, so though the derived models do not factor in clock rate as a predictor variable, they are not biased by running the tests on various machine types. All tests were run on a state-of-the-art 2.6 GHz Intel Xeon E5 processors.

Moreover, I limit my analysis to the prediction of runtimes of models in the data-driven tier of species distribution models, and not attempt to characterize the computational time of more traditional statistical methods or Bayesian approaches. Systematic literature review suggests that this is a reasonable simplification to make, because a majority of contemporary SDM users utilize these machine learning methods. Furthermore, I limit my modeling to the implementations of these models that are run through an R interface using popular, open-source R packages. Some SDM models, MaxEnt (Java), for example, are written in other languages, which can be faster and provide a more customizable interface. However, given the popularity of the R environment, a review of the SDM literature, and the open-source nature of R packages suggest that an increasing number of global change researchers utilize R to run the SDMs.

Finally, my methodological approach was strongly limited by computational cost, both financial and time. Each SDM configuration was tested at between five and time times to ensure robust results. Experiments in my set ranged from less than five seconds to greater than 11,000 seconds (3.3 h). In order to gather enough data to develop a robust predictive model, I limited the number of very long running models. Similarly, I limited my experimentation on virtual servers with very high vCPU counts or memory allocations. The cost of these instances was more than an order of magnitude of larger than smaller instances (> $1/hr), so experimentation was shifted to less costly servers. More data collected in all areas, particularly on virtual instances with high memory and many CPUs may improve the robustness of my results.

#### Data Collection

#### SDM Data Preparation

I collected data on the run time and accuracy of four SDM algorithms that have shown competitive accuracy results in the literature: multivariate adaptive regression splines (MARS) (Leathwick, Elith, & Hastie, 2006), gradient boosted regression trees (GBM-BRT)(Elith et al., 2008; Friedman, 2001; Natekin, 2013), generalized additive models (GAM) (Guisan, Edwards, & Hastie, 2002; Yee & Mitchell, 1991), and Random Forests (Breiman, 2006; Elith & Graham, 2009). All of the models were fit using the R statistical environment (R Core Team, 2016) with standard packages for fitting these models. GBM-BRT tree models were fit using the ‘dismo’ package version 1.1-1 (**???**), GAMs were fit using the ‘gam’ package, version 1.12 (Hastie, 2015), and MARS were fit using the ‘earth’ package version 4.4.4 (Hastie & wrapper, 2016), and random forests were fit with the package ‘randomForest’ (Liaw & Wiener, 2002). These standard packages were chosen for their popularity in the field, and are designed to represent normal use cases in species distribution modeling workflows.

Input data for the SDMs was obtained from the Neotoma Paleoecological Database in April 2015. All records for the genuses *Picea* (spruce), *Quercus* (oak), *Tsuga* (Hemlock), and *Betula* (birch) were downloaded using the ‘neotoma’ R package (Goring et al., 2015). Occurrence records were filtered to only include those in the last 22,000 and in North America. For each record, the latitude, longitude, age, and relative abundance of the taxon was retained using a comma separated value format.

The climatic predictor layers used were downscaled and debiased Community Climate System Version 3 (CCSM3) model simulations for North America (David J Lorenz et al., 2016). The post-processed model output was obtained in NetCDF format with a 0.5 degree spatial resolution and decadal temporal resolution for the last 22,000 years (D J Lorenz et al., 2016). Bioclimatic variables (BV) (O’Donnell & Ignizio, 2012) were calculated for each timestemp using the biovars function in the ‘dismo’ R package (**???**). BV values were extracted to the Neotoma fossil occurrence data at each space-time location.

The occurrence-climate datasets were then filtered to include on the six least correlated predictors, a common practice when applying learning algorithms. Collinearity among predictors can decrease model performance and can cause situations in which small changes in data produce large swings in parameter estimates, and “in truly extreme cases, prevent the numerical solution of a model.” (Obrien, 2007). The Variance Inflation Factor was used to determine variable correlation. VIF quantifies the expected amount of variance in a regression coefficient that is dues to collinearity in its predictors, lower bounded by 1 (no inflation) with no upper bound. Variance inflation was calculated using the ‘usdm’ package. Based on the result of this analysis, I retained the six least intercorrelated variables, leaving a maximum correlation of 0.51. The variables I retained were BV2 (mean diurnal temperature range), BV7 (annual temperature range), BV8 (mean temperature of westtest quarter), BV15 (precipitation of warmest quarter), BV17 (precipitation of warmest quarter), and BV18 (precipitation of driest quarter). All SDMs were fit with these variables as predictor features.

Future climate layers for AD2100 were obtained from the CMIP project, HadCM3 climate model. These layers model expected climate variables under the UN IPCC RCP 8.5, a scenario that assumes high population, moderate economic growth, and a sustained dependence on fossil fuels (Riahi et al., 2011). These layers were converted to bioclimatic variables and resampled to various resolutions for their use as output layers in different experiments.

With the trend of ecological Big Data, it is foreseeable that future datasets may exceed any currently available. To assess the affect of very large input datasets (n > 1e7), I created a four simulated datasets of given sizes: 250MB, 500MB, 1000MB, 2000MB. These datasets were created using a python script that randomly generated a latitude, longitude, age, abundance, and bioclimatic variable assemblage. Each row was 327 bytes of information. Rows were randomly simulated until a given file size had been reached.

#### Computing Infrastructure

I used the Google Cloud Compute Enginge (GCE) to complete all of my experiments. A popular Infrastructure-as-a-service (IaaS) provider (Hassan, 2011), the platform rents out a wide array of virtual server instances, from the most basic (1 CPU, 0.6 GB RAM) to exceptionally powerful (32 CPU, 208 GB RAM). The platform also provides a set of application programming interfaces (API) to allow workflow automation on the virtual computing instances, as well as a graphical user interface (GUI) for interactive resource provisioning.

Google’s IaaS platform was chosen over other public cloud vendors because of its ability to create ‘custom’ instance types of user defined specifications. Other vendors (e.g., Amazon Web Services) provide a larger number of predefined instance types, some even more powerful than Google’s top-end, but do not allow you to create an instance with an arbitrary number of processors and memory. By providing the ability to create custom types, Google’s service fits well into my experimental design, and lets me avoid using software solutions to artificially alter hardware parameters.

I set up a distributed computing system to complete my experiments, featuring one centralized database node and multiple distributed computing nodes. Fault tolerance was important, because I utilized Google’s less expensive ‘preemptible’ resources, which function like normal instances, but can be shutdown at any time if other customers require additional computational power. One Master Node hosts a MySQL database and a control script (written in python), and a pool of computing nodes that are fault tolerant and designed only for computing are provisioned and decommissioned as needed. The compute nodes are given only enough information to complete a given SDM, and the Master Node control script manages the progress of the project as a whole. A node.js script provides programmatic access to real-time database content.

An outline of the system is described below and are illustrated in Figure [X].

1. First, a pool of computing nodes is assembled. The Master Node control script queries the central database for experiments that have yet to be completed or threw an error the last time it was run. The database responds, via the API, a JSON object that contains the number of cores and memory needed for the next experiment. The python script parses the response and uses the ‘gcloud’ tools associated with the GCE to create a pool of virtual instances that have the memory and CPUs required by the next experiment.
2. Each node in the pool automatically begins running a startup script that begins the modeling process. First, a number of system-wide software packages, including R and Git are installed on the new instance. Git is used to clone the most recent version of the project repository which contains all files necessary to compute an SDM. Once all packages have been installed, the timing script is initialized in a new R session. The R script queries the central database, identifying itself as a computing node with cores and memory. The database responds with the parameters needed to run a single experiment on the given infrastructure. The script then loads the necessary variables and runs the SDM. When finished, it reports its results to the database and marks the experiment as completed. Experiments are continued until there are no more experiments that can be computed on this instance. If an instance is preempted by the system or otherwise crashes, a shutdown script will be executed, marking the in-progress experiment as interrupted, and that it should be attempted again by another computing node.
3. Because Google charges by the minute for the use of their virtual machines, instances must be torn down as soon as possible. While the computing nodes execute the experiments, the Master Node repeatedly polls the central database to determine the current position within the experiment table, attempting to determine the percentage completion of the current group of experiments. If the group is complete, Master Node will use the gcloud tools to deleted the individual instances, the instance pool, and the template that was used to create each instance. After this, the Master Node is the only instance that remains online. At this point, Master Node returns to Step 1 to build a new pool of instances for the new memory/cores combination.

#### SDM Model Protocol

The experimental parameters are communicated to the worker node by the central database. The computing node parses the database’s response and starts a new experiment session. First, the set of occurrences corresponding with the species to be modeled is loaded from the disk. It is then randomly partitioned two nonoverlapping subsets, a training set of occurrences, and a testing set of 20% of the total number (*Picea* = 9935, *Quercus* = 8953, *Betula* = 10226, *Tsuga* = 7140). All examples were converted to binary presence-absence values using the Nieto-Lugilde (Nieto-Lugilde, Maguire, Blois, Williams, & Fitzpatrick, 2015) method for determining local presence from fossil pollen records. The training set is then sent to the specified learning function where an SDM model is fit using predictors. The model use then used to predict that taxon’s range in 2100 AD under the RCP 8.5 scenario. The holdout testing set is used to evaluate the model’s ability to discriminate presence-absence from the predictors. During the execution separate times are recorded for model fitting, prediction onto the gridded surface, and accuracy calculation, as well as the total time. Furthermore, several measures of accuracy, including the Area Under the Receiver Operator Curve (AUC), a popular method of evaluating logistic output (but see (Lobo, Jiménez-Valverde, & Real, 2008)). There is no database I/O inside of the timing script, so results should not be slow-biased by network connection or context switching. Learning parameters (learning rate, number of trees, tree complexity, etc) are held constant for all runs except a small subset which were designed specifically to determine sensitivity to these parameters. Timing was done within R using the proc.time function.

Due to limited time and budget constraints, and thus inability to cover all possible parameterizations, experiments were divided into several categories in which different variables were systematically altered to determine sensitivity. This compromise aims to capture as much within-parameter variance as possible while simultaneously capturing the influence of interactions between variables. One basic series of experiments was run for all SDMs using a small subset of algorithm inputs (training examples and memory) on a wide variety of VM types (core/memory combinations). Five additional, separate analyses were completed to determine the sensitivity to specific parameterizations.

The largest number of SDM experiments was done on a small combination of training examples and spatial resolutions but over a large number of computing instances, and were used to assess the performance of serial SDM algorithms under default conditions. Attempts were made to capture interactions between variables and to capture the contribution of algorithms on increasingly more power machine types. Because execution time can vary non-linearly when the hardware parameters are changed, I tested as many combinations of memory and CPUs as possible. On each computer, a standard set of 160 experiments were run for each sequential SDM (MARS, GBM-BRT, GAM), including four spatial resolutions, four training example sets, and 10 replicates of each cell. All experiments were done on the *Picea* pooled niche data set. 20,225 experiments were completed for this category. Settings for each algorithm are included in Table [X].

[XXXXXXXXXX TABLE X XXXXXXXXXX] DO TABLE STUFF [XXXXXXXXXX TABLE X XXXXXXXXXX]

Individual, target experiment sets were done to assess the contribution of individual or sets of parameterizations. While no theoretical difference would suggest that execution times should vary between different taxa, a set of 191 model runs were done to evaluate whether differences exist in practice. The model uses aspatial input sets, which suggests that geographic range, abundance, or taxon-specific patterns should not bias the results of the experiments. Using all four taxa on six different VM instance types, inter-taxonomic sensitivity was recorded. The number of training examples and spatial resolutions was held constant for these runs.

Empirical Performance Models suggest that specific algorithm parameterizations will take longer to execute than others (Cannon & John, 2007). SDM have a large number of potential parameters with which to alter, though many ecologists use the defaults, or packages that make it difficult to change the default parameter values (e.g., dismo (**???**)). To assess the magnitude of changes in execution time due to different parameterizations, the GBM-BRT was tested on a set of 180 [CITE THIS NUMBER WHEN DONE WITH EXPERIMENTS] different parameterizations. During these experiments, the learning rate, tree complexity, and number of training examples were systematically altered. The learning rate parameter of the GBM-BRT model is a shrinkage parameter that reduces the impact of each additional fitted tree, driven by the boosting paradigm of fitting a model with many small models rather than fewer large trees. If one of the greedy iterations does not improve model fit, the contribution of that iteration can be easily reversed in the subsequent iterations (Natekin, 2013). The tree complexity parameter controls whether interactions between predictors are fitted. If tree complexity is 1, the tree will be an additive model with no interactions. A tree complexity of will produce a model with -way interactions between variables(Elith et al., 2008). These two variables together control the total number of trees needed to fit the model (Elith et al., 2008). All of these experiments (N=?????) were run on a single instance type of 1 CPU and 3.75 GB RAM. All experiments used the *Picea* niche set. Refer to Table [X] fora parameterization details.

To assess the relative performance of parallel methods over sequential models, random forest SDMs were fit both sequentially and in parallel on instances up to 24 CPU cores. Spatial resolution, memory, and taxon were held constant while number of ensemble members and number of training examples were systematically altered for each core. In total, 3300 random forests were fit using randomForest with the foreach package providing parallelization support. Sequential runs were fit using the same function but with number of cores set to only 1. Three parameters of training examples and three of number of ensemble members were altered as well. Table [X] shows the parameterizations used in this series.

In addition to performance gains made by increasing the number of CPU cores and leveraging parallel methods, increasing instance memory should improve performance for very large datasets. Using the simulated datasets, I attempted to assess the performance of the model when faced with more than 1 million input examples. R has little support for high memory tasks, and these tests routinely crashed the computer when trying to fit the SDM due to inability to allocate memory space.

Finally, I evaluated the effect of varying the number of predictors on the execution time of the algorithm. The literature on theoretical complexity of algorithms (e.g., (Hastie et al., 2009)) often characterize the complexity of machine learning algorithms in terms of both number of training examples and number of features in each example. I systematically modified the number of training examples between 1000 and 11000 and the number of predictors on GBM-BRT and sequential random forests. Because both of these algorithms run serially, they can safely be run on a single processor without the need for estimating the effects of additional cores. [ADD MORE DETAILS ABOUT THIS HERE].

[Add summary statistics about the modeling process here. In total there were x configurations, blah blah blah] Need to wait on this.

#### Modeling Execution Time and Accuracy

To model algorithm execution time, I created a separate model for each SDM type. Gradient Boosted Regression trees, implemented in R though the ‘gbm’ package (Ridgeway, 2006), because tree based models have been previously shown to be highly effective in empirical performance models (Hutter, Xu, et al., 2014). For each model, a holdout set of 100 observations was randomly selected for evaluation purposes. Each model was developed to predict the log-transform total time of execution, which is the sum of the SDM fitting time, prediction time, and time to calculate accuracy statistics. Log transforms are used because they do not allow negative predictions, which are possible under non-transformed inputs and have been shown to be more accurate when observed responses span large ranges (Hutter, Xu, et al., 2014). The GBM models were fit using the R package defaults of a tree complexity of 1 (no interaction) and a learning rate of 0.01, with a bag fraction of 0.75. 15000 trees were calculated initially, and the best subset of those were used by calculating ‘gbm.perf’ for each model.

SDM accuracy was modeled in the same way as execution time. A separate accuracy model was built for each SDM type. Separate models were calculated rather than using the model algorithm as a categorical variable because of the different assumptions and parameters that go into each algorithm. Accuracy was not log transformed before modeling.

#### Resource Utilization

To develop a more thorough understanding of the way in which hardware variables contribute to SDM execution speed, I monitored the runtime environment and record CPU and memory utilization as the models were being executed. Using a python script and the ‘psutil’ (https://github.com/giampaolo/psutil) module, I recorded the relative utilization of each individual CPU core, total CPU utilization, and memory utilization. These measurements were recorded in the central database and later linked to individual SDM runs by using measurement timestamps. Resource utilization was also recorded by the Google infrastructure itself and displayed in real time as the models run. These data were downloaded as JSON and interpreted in conjunction with the python monitoring.

#### Model Evaluation

Contributions to each predictive model were evaluated using partial dependency plots and variable influence, the non-linear analog to ANOVA testing to a linear model. Performance and accuracy models were built using a holdout testing set of 100 randomly selected observations, and evaluated using the difference between observed and predicted values. The residual sum of square and mean prediction error are reported for all models. The best model is selected as the model with the lowest residual sum of squares.

Model sensitivity to changes in input was analyzed using generalized Sobol’ indices, which provide a mechanism for comparing model structure across SDM classes. Sobol’ indices, also refered to as a functional anova.

#### Optimal Prediction

The optimal prediction for computing platform was done by integrating the model of computing time with a model of computing cost. Because the computing runtime models are able to predict to any unseen combination of input parameters, it can be used to predict the cost of an arbitrary number of future experiments. For a set of given scenarios, , the time for computing them on 287 different computing instance types was calculated by multiplying the hourly rate of the instance by the expected computing time on that instance. In this context, the VM’s effective computing power is composed of its number of CPUs and amount of memory, though these results could easily be applied to higher dimensional feature vectors if additional components of computing power were considered. The cost

### Results and Discussion

#### 

1. Selected literature review

* This will need some serious revision from last spring
* Focus more on the ecological dimensions of why this is important
* Then connect to computing, machine learning, etc
* Finally, review algorithms and optimization techniques

1. Species distribution models
2. What are they? (brief)
3. Ecological foundations, niches, use of paleodata to improve accuracy
   * Data availability
4. Machine learning and species distribution models
   * Models used to be simple (boxcar models)
   * Now they’re very complex
   * High variance, low bias
   * Low variance, high bias
   * Look at cited AUC/accuracy metrics
   * No clear winner for all tasks
   * All methods are still widely used
   * Maxent and its popularity
   * Ensemble and parallel methods and their application/accuracy
5. Prediction and hindcasting using models as a key way to understand the past and future
   * Cite land manager uses here (this is more than just hypotheses for ecological testing)
   * These are real issues that need support (invasive species)
6. Meta-analysis/results of targeted reading
   * Other papers commenting on the growth of the field
   * This will flow nicely from the review of what people actually use these models for
7. Cloud computing as a technology to support researchers
8. Support for machine learning
9. Designed for big data and distributed processing
   * We’ve already clarified that ecological data is Big Data, so this will be easy to reinforce here
10. The cloud as a research tool, rather than a market device
    * Not too much on this, but note the economic underpinnings of the computing as a service
    * Cite NSF/NASA/others that require cloud computing for research
11. Benchmarking, timing, and why it matters
12. Systems evaluation and benchmarking
    * Overview of types of benchmarks
    * Application level benchmarks are the best
    * Need for repeated measurements
    * Point of section: stochastic variance in benchmarks
    * Non-linear, complex, hard to model
    * But it’s okay
    * Potentially, consequences of using virtual instances –> few, using monitor scripts
13. Algorithms Optimization
    1. What affect’s an empirical/theoretical runtime?
    * Introduce my experimental variables
    * Need to read more on the theoretical underpinnings of memory/paging/CPU/etc
    * Briefly touch on theoretical runtime complexity
    1. Other attempts at empirical runtime modeling
    * Need to read more on this
    * We extend this away from just algorithm inputs to hardware inputs too.
    1. Sensitivity analysis vs. optimization analysis
    * Maybe we need to change some terminology here,
    * I think with the alg. opt. literature I can still call it optimization and prediction.
14. Problem Formulation

* Do I need to update this? Probably more or less close to being done

1. Specific components of the framework to address in the thesis

* The framework introduces six components involved in the optimization
* I just look at one of the central components (time to compute, and address the others tangentially)
* Demonstrate the proof of concept of the framework, leave the other components to other researchers

1. Methods
2. Data collection
   1. Species distribution modeling inputs
   * GBIF and Neotoma
   * Climate model output
   * Data preparation and cleaning
   1. Simulated data for large memory experiments
   * Do I need to do this? Maybe GBIF would let me do a real species.
   * Simulated data would make more sense from a computing standpoint
   * Real data would make more sense from a user/thesis standpoint
   1. Cost model data
   * Does this go in data? probably
3. Computing experiments
   1. Computing set up
   * Flowchart framework
   * Google cloud description
   1. Serial SDM experiments
   2. Inter-model differences
   3. Taxonomic differences
   4. Parameter sensitivity
   5. Training example sensitivity
   6. Serial SDMs with large memory requirements
   * I think this will be a nice flow of experiment descriptions
   1. Parallel SDM experiments
   * Need to specifically introduce that these need to be considered separately in my framework, because they respond to differences in cores
   * Might have less accuracy or cost more than methods above,
   * Might have more accuracy than methods above, and can be executed on a single core
   * Just random forests
     + Parallel machine learning methods are a topic of active CS research,
     + This probably needs to go into literature review, or could go into discussion/conclussion
4. Predictive Modeling Building
   1. Runtime prediction
   2. Linear model
      * Do I even need to show results of LM?
      * Ref: comments from CI
   3. GBM
      * Able to capture non-linearities
   4. Accuracy prediction
   * Build one accuracy model for each SDM class
   * Can we test this from the literature too?
   1. Cost optimization model building
5. Discussion and Results
6. Computational runtime prediction accuracy assessment
   * Should formalize this
     + Least squares?
7. Accuracy prediction assessment
   * Parallel methods and their accuracy
8. Cost optimization assessment
   * This will be tricky to assess quantitatively
   * Need to think about this more
   * Qualitatively, we can do this fairly easily
9. Case study
   * Need to find a good case study
   * Illustrate model results and utility
   * Discuss limitations and uncertainties
   * Discuss confidence in results
10. Limitations of current approach
    * How much will the additional components of the framework influence the results?
    * Modeling expertise can do more than predictive modeling
    * Stress uncertainties and lack of predictive skill
    * Scientific realities over modeled optima
    * we should try to find some literature about compromising workflows to meet computational demands.
11. Conclusion
12. Reiterate and answer research questions
13. Next steps to reduce uncertainty remaining in the model
14. Areas where additional research is needed
    * Parallel machine learning methods
15. Bibliography

Discussion Idea -> R’s language design and memory

ARAUJO, M., & NEW, M. (2007). Ensemble forecasting of species distributions. *Trends in Ecology & Evolution*, *22*(1), 42–47.

Araújo, M. B., & Guisan, A. (2006). Five (or so) challenges for species distribution modelling. *Journal of Biogeography*, *33*(10), 1677–1688.

Araújo, M. B., Whittaker, R. J., Ladle, R. J., & Erhard, M. (2005). Reducing uncertainty in projections of extinction risk from climate change. *Global Ecology and Biogeography*, *14*(6), 529–538.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., & Katz, R. H. (2009). Above the clouds: A berkeley view of cloud computing.

Austin, M. P. (2002). Spatial prediction of species distribution: an interface between ecological theory and statistical modelling. *Ecological Modelling*, *157*(2-3), 1–18.

Barnosky, A. D., Hadly, E. A., Bascompte, J., Berlow, E. L., Brown, J. H., Fortelius, M., Getz, W. M., et al. (2012). Approaching a state shift in Earths biosphere. *Nature*, *486*(7401), 52–58.

Beck, J., Böller, M., Erhardt, A., & Schwanghart, W. (2014). Spatial bias in the GBIF database and its effect on modeling species’ geographic distributions. *Ecological Informatics*, *19*(C), 10–15.

Bifet, A., Holmes, G., Pfahringer, B., & Gavalda, R. (2011). Detecting Sentiment Change in Twitter Streaming Data. *WAPA*.

Blaauw, M. (2010). Methods and code for “classical” age-modelling of radiocarbon sequences. *Quaternary Geochronology*, *5*(5), 512–518.

Bolker, B. M., Brooks, M. E., Clark, C. J., Geange, S. W., Poulsen, J. R., Stevens, M. H. H., & White, J.-S. S. (2009a). Generalized linear mixed models: a practical guide for ecology and evolution. *Trends in Ecology & Evolution*, *24*(3), 127–135.

Bolker, B. M., Brooks, M. E., Clark, C. J., Geange, S. W., Poulsen, J. R., Stevens, M. H. H., & White, J.-S. S. (2009b). Generalized linear mixed models: a practical guide for ecology and evolution. *Trends in Ecology & Evolution*, *24*(3), 127–135.

Breiman, L. (2006). randomForest: Breiman and Cutler’s random forests for classification and regression.

Brewer, E. A., & Brewer, E. A. (1995). *High-level optimization via automated statistical modeling* (Vol. 30). New York, New York, USA: ACM.

Brewer, S., Jackson, S. T., & Williams, J. W. (2012). Paleoecoinformatics: applying geohistorical data to ecological questions. *Trends in Ecology & Evolution*, *27*(2), 104–112.

Cannon, A. R., & John, C. H. S. (2007). Measuring Empirical Computational Complexity. *Organizational Research Methods*, *10*(2), 1–10.

Carroll, J. M. (1999). Five Reasons for Scenario-based Design. *HICSS 1999*, *Track3*, 11.

Chen, M., Mao, S., & Liu, Y. (2014). Big Data: A Survey. *Mobile Networks and Applications*, *19*(2), 171–209.

Clark, J. S., Gelfand, A. E., Woodall, C. W., & Zhu, K. (2014). More than the sum of the parts: forest climate response from joint species distribution models. *Ecological Applications*, *24*(5), 990–999.

Cormen, T. H. (2009). *Introduction to Algorithms*. MIT Press.

Davis, M. B. (1963). On the theory of pollen analysis. *American Journal of Science*, *261*(10), 897–912.

Dawson, A., Paciorek, C., McLachlan, J., Goring, S., WIlliams, J., & Jackson, S. T. (2016). Quantifying pollen-vegetation relationships to reconstruct ancient forests using 19th-century forest composition and pollen data, 1–76.

Dongarra, J., Martin, J. L., & Worlton, J. (1987). Computer benchmarking: Paths and pitfalls: The most popular way of rating computer performance can confuse as well as inform; avoid misunderstanding by asking just what the benchmark is measuring. *IEEE Spectrum*, *24*(7), 38–43.

Dormann, C. F., Schymanski, S. J., Cabral, J., Chuine, I., Graham, C., Hartig, F., Kearney, M., et al. (2012). Correlation and process in species distribution models: bridging a dichotomy. *Journal of Biogeography*, *39*(12), 2119–2131.

Elith, J., & Graham, C. H. (2009). Do they? How do they? WHY do they differ? On finding reasons for differing performances of species distribution models. *Ecography*, *32*(1), 66–77.

Elith, J., & Leathwick, J. R. (2009a). Species Distribution Models: Ecological Explanation and Prediction Across Space and Time. *Annual Review of Ecology, Evolution, and Systematics*, *40*(1), 677–697.

Elith, J., & Leathwick, J. R. (2009b). Species Distribution Models: Ecological Explanation and Prediction Across Space and Time. *Annual Review of Ecology, Evolution, and Systematics*, *40*(1), 677–697.

Elith, J., H Graham, C., P Anderson, R., Dudík, M., Ferrier, S., Guisan, A., J Hijmans, R., et al. (2006a). Novel methods improve prediction of species distributions from occurrence data. *Ecography*, *29*(2), 129–151.

Elith, J., H Graham, C., P Anderson, R., Dudík, M., Ferrier, S., Guisan, A., J Hijmans, R., et al. (2006b). Novel methods improve prediction of species distributions from occurrence data. *Ecography*, *29*(2), 129–151.

Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Ecology*, *77*(4), 802–813.

Elith, J., Phillips, S. J., Hastie, T., Dudík, M., Chee, Y. E., & Yates, C. J. (2010). A statistical explanation of MaxEnt for ecologists. *Diversity and Distributions*, *17*(1), 43–57.

Ficetola, G. F., Thuiller, W., & Miaud, C. (2007). Prediction and validation of the potential global distribution of a problematic alien invasive species - the American bullfrog. *Diversity and Distributions*, *13*(4), 476–485.

Fink, E. (1998). How to Solve It Automatically: Selection Among Problem Solving Methods. *AIPS*.

Fitzpatrick, M. C., Gotelli, N. J., & Ellison, A. M. (2013). MaxEnt versus MaxLike: empirical comparisons with ant species distributions. *Ecosphere*, *4*(5), art55–15.

Fløjgaard, C., Normand, S., & Skov, F. (2009). Ice age distributions of European small mammals: insights from species distribution modelling. *Journal of Biogeography*, *36*(6), 1152–1163.

Foster, I. (2005). Service-Oriented Science. *Science*, *308*(5723), 814–817.

Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008). Cloud computing and grid computing 360-degree compared. *2008 Grid Computing …*, 1–10.

Franklin, J. (2010). *Mapping Species Distributions*. Spatial inference and prediction (Vol. 44). Cambridge University Press.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*.

Fritz, S. A., Schnitzler, J., Eronen, J. T., Hof, C., Böhning-Gaese, K., & Graham, C. H. (2013). Diversity in time and space: wanted dead and alive. *Trends in Ecology & Evolution*, *28*(9), 509–516.

Glew, J. R., Smol, J. P., & Last, W. M. (2002). Sediment Core Collection and Extrusion. In *Tracking environmental change using lake sediments* (pp. 73–105). Dordrecht: Springer Netherlands.

Golding, N., & Purse, B. V. (2016). Fast and flexible Bayesian species distribution modelling using Gaussian processes. *Methods in Ecology and Evolution*, *7*(5), 598–608.

Goldsmith, S. F., Aiken, A. S., & Wilkerson, D. S. (2007). *Measuring empirical computational complexity*. New York, New York, USA: ACM.

Goring, S., Dawson, A., Simpson, G. L., Ram, K., Graham, R. W., Grimm, E. C., & Williams, J. W. (2015). neotoma: A Programmatic Interface to the Neotoma Paleoecological Database. *Open Quaternary*, *1*(6), 1–17.

Grimm, E. C., Bradshaw, R. H. W., Brewer, S., Flantua, S., Glesecke, T., Lezine, A.-M., Takahara, H., et al. (2013). Databases and Their Application. *Encycolpedia of Quaternary Science*, 1–9.

Guisan, A., & Thuiller, W. (2005a). Predicting species distribution: offering more than simple habitat models. *Ecology Letters*, *8*(9), 993–1009.

Guisan, A., & Thuiller, W. (2005b). Predicting species distribution: offering more than simple habitat models. *Ecology Letters*, *8*(9), 993–1009.

Guisan, A., & ZImmerman, N. (2000). Predictive habitat distribution models in ecology. *Ecological Modelling*, *135*(2-3), 1–40.

Guisan, A., Edwards, T. C., & Hastie, T. (2002). Generalized linear and generalized additive models in studies of species distributions: setting the scene. *Ecological Modelling*, *157*(2-3), 89–100.

Guisan, A., Tingley, R., Baumgartner, J. B., Naujokaitis-Lewis, I., Sutcliffe, P. R., Tulloch, A. I. T., Regan, T. J., et al. (2013). Predicting species distributions for conservation decisions. *Ecology Letters*, *16*(12), 1424–1435.

Gustafson, J. L. (1988). Reevaluating Amdahl’s law. *Communications of the ACM*, *31*(5), 532–533.

Hampton, S. E., Strasser, C. A., Tewksbury, J. J., Gram, W. K., Budden, A. E., Batcheller, A. L., Duke, C. S., et al. (2013). Big data and the future of ecology. *Frontiers in Ecology and the Environment*, *11*(3), 156–162.

Hassan, Q. (2011). Demystifying Cloud Computing. *CrossTalk*, 16–21.

Hastie, S. M. D. from mda mars by T., & wrapper, R. T. U. A. M. F. utilities with T. L. leaps. (2016). *earth: Multivariate Adaptive Regression Splines*.

Hastie, T. (2015). *gam: Generalized Additive Models*.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. *International Statistical Review*, *77*(3), 482–482.

Hegel, T. M., Cushman, S. A., Evans, J., & Huettmann, F. (2010). Current State of the Art for Statistical Modelling of Species Distributions. In *Spatial complexity, informatics, and wildlife conservation* (pp. 273–311). Tokyo: Springer Japan.

Hobbie, J. E., Carpenter, S. R., Grimm, N. B., Gosz, J. R., & Seastedt, T. R. (2003). The US Long Term Ecological Research Program. *BioScience*, *53*(1), 21–32.

Hsu, C.-H., Lin, C.-Y., Ouyang, M., & Guo, Y. K. (2013). Biocloud: Cloud Computing for Biological, Genomics, and Drug Design. *BioMed Research International*, *2013*, 1–3.

Huang, Q., Yang, C., Liu, K., Xia, J., Xu, C., Li, J., Gui, Z., et al. (2013). Evaluating open-source cloud computing solutions for geosciences. *Computers and Geosciences*, *59*(C), 41–52.

Huang, Q., Yang, C., Nebert, D., Liu, K., & Wu, H. (2010). *Cloud computing for geosciences: deployment of GEOSS clearinghouse on Amazon’s EC2*. Deployment of geoss clearinghouse on amazon’s ec2. New York, New York, USA: ACM.

Hutchinson, G. E. (1957). Concluding Remarks. *Cold Spring Harbor Symposia on Quantitative Biology*, *22*(0), 1–7.

Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2014). Identifying Key Algorithm Parameters and Instance Features using Forward Selection. *High Performance Computing for Computational Science - VECPAR 2012*, *7997*(Chapter 40), 1–15.

Hutter, F., Xu, L., Hoos, H. H., & Leyton-Brown, K. (2014). Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence*, *206*(C), 79–111.

Issa, S. A., Kienzler, R., El-Kalioby, M., Tonellato, P. J., Wall, D., Bruggmann, R., & Abouelhoda, M. (2013). Streaming Support for Data Intensive Cloud-Based Sequence Analysis. *BioMed Research International*, *2013*(8), 1–16.

Jones, R., & Kalibera, T. (2013). *Rigorous benchmarking in reasonable time* (Vol. 48). ACM.

Kaján, L., Yachdav, G., Vicedo, E., Steinegger, M., Mirdita, M., Angermüller, C., Böhm, A., et al. (2013). Cloud Prediction of Protein Structure and Function with PredictProtein for Debian. *BioMed Research International*, *2013*(3), 1–6.

Keppel, G., Van Niel, K. P., Wardell-Johnson, G. W., Yates, C. J., Byrne, M., Mucina, L., Schut, A. G. T., et al. (2011). Refugia: identifying and understanding safe havens for biodiversity under climate change. *Global Ecology and Biogeography*, *21*(4), 393–404.

Knuth, D. E. (1976). Big Omicron and big Omega and big Theta. *ACM Sigact News*, *8*(2), 18–24.

Kogan, J. (2014). Feature Selection Over Distributed Data Streams. In *Data mining for service* (pp. 11–26). Berlin, Heidelberg: Springer Berlin Heidelberg.

Kundra, V. (2010). 25 Point Implementation Plan to Reform Federal information Technology Management.

Leathwick, J. R., Elith, J., & Hastie, T. (2006). Comparative performance of generalized additive models and multivariate adaptive regression splines for statistical modelling of species distributions. *Ecological Modelling*, *199*(2), 188–196.

Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. *R News*, *2*(3), 18–22.

Lilja, D. (2009). **Measuring Computer Performance: A Practitioner’s Guide**. Cambridge: Cambridge University Press.

Lobo, J. M., Jiménez-Valverde, A., & Real, R. (2008). AUC: a misleading measure of the performance of predictive distribution models. *Global Ecology and Biogeography*, *17*(2), 145–151.

Lorenz, D. J., Nieto-Lugilde, D., Blois, J. L., Fitzpatrick, M. C., & Williams, J. W. (2016). Downscaled and debiased climate simulations for North America from 21,000 years ago to 2100AD. *Scientific Data*, *3*, 160048–19.

Lorenz, D. J., Nieto-Lugilde, D., Blois, J. L., Fitzpatrick, M. C., & Williams, J. W. (2016). *Data from: Downscaled and debiased climate simulations for North America from 21,000 years ago to 2100AD*.

Lowe, C. B., Kellis, M., Siepel, A., Raney, B. J., Clamp, M., Salama, S. R., Kingsley, D. M., et al. (2011). Rapid Range Shifts of Species Associated with High Levels of Climate Warming. *Science*, *333*(6045), 1019–1024.

Maguire, K. C., Nieto-Lugilde, D., Fitzpatrick, M. C., Williams, J. W., & Blois, J. L. (2015a). Modeling Species and Community Responses to Past, Present, and Future Episodes of Climatic and Ecological Change. *Annual Review of Ecology, Evolution, and Systematics*, *46*(1), 343–368.

Maguire, K. C., Nieto-Lugilde, D., Fitzpatrick, M. C., Williams, J. W., & Blois, J. L. (2015b). Modeling Species and Community Responses to Past, Present, and Future Episodes of Climatic and Ecological Change. *Annual Review of Ecology, Evolution, and Systematics*, *46*(1), 343–368.

Manyika, J., Chui, M., Brown, B., Bughin, J., & Dobbs, R. (2015). *Big data: The next frontier for innovation, competition, and productivity. 2011*. URL http://www. mckinsey. ….

Mell, P. M., & Grance, T. (2012). The NIST definition of cloud computing. *National Institude of Standards and Technology*, 1–7.

Michener, W. K., & Jones, M. B. (2012). Ecoinformatics: supporting ecology as a data-intensive science. *Trends in Ecology & Evolution*, *27*(2), 85–93.

Miller, J., Franklin, J., & ASPINALL, R. (2007). Incorporating spatial dependence in predictive vegetation models. *Ecological Modelling*, *202*(3-4), 225–242.

Mosco, V. (2014). To the Cloud: Big Data in a Turbulent World. *Popular Communication*, *12*(4), 271–269.

Moss, R. H., Edmonds, J. A., Hibbard, K. A., Manning, M. R., Rose, S. K., Vuuren, D. P. van, Carter, T. R., et al. (2010). The next generation of scenarios for climate change research and assessment. *Nature*, *463*(7282), 747–756.

Natekin, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, *7*, 1–21.

Nieto-Lugilde, D., Maguire, K. C., Blois, J. L., Williams, J. W., & Fitzpatrick, M. C. (2015). Close agreement between pollen-based and forest inventory-based models of vegetation turnover. *Global Ecology and Biogeography*, *24*(8), 905–916.

Nogués-Bravo, D. (2009a). Predicting the past distribution of species climatic niches. *Global Ecology and Biogeography*, *18*(5), 521–531.

Nogués-Bravo, D. (2009b). Predicting the past distribution of species climatic niches. *Global Ecology and Biogeography*, *18*(5), 521–531.

Nogués-Bravo, D., Rodríguez, J., Hortal, J., & Batra, P. (2008). Climate change, humans, and the extinction of the woolly mammoth. *PLoS Biology*, *6*(4), e79.

Nordhaus, W. D., & Yale University. Cowles Foundation for Research in Economics. (2001). The Progress of Computing.

Norman, D. A. (1984). Stages and levels in human-machine interaction. *International journal of man-machine studies*, *21*(4), 365–375.

Obrien, R. M. (2007). A Caution Regarding Rules of Thumb for Variance Inflation Factors. *Quality & Quantity*, *41*(5), 673–690.

O’Donnell, M. S., & Ignizio, D. A. (2012). Bioclimatic predictors for supporting ecological applications in the conterminous United States. *US Geological Survey Data Series*.

Pearman, P. B., Guisan, A., Broennimann, O., & Randin, C. F. (2008). Niche dynamics in space and time. *Trends in Ecology & Evolution*, *23*(3), 149–158.

Peterson, A. T. (2003). Predicting the geography of species’ invasions via ecological niche modeling. *The quarterly review of biology*, *78*(4), 419–433.

Peterson, A. T., Soberon, J., & Sánchez-Cordero, V. (1999). Conservatism of Ecological Niches in Evolutionary Time. *Science*, *285*(5431), 1265–1267.

Phillips, S. J., Anderson, R. P., & Schapire, R. E. (2006). Maximum entropy modeling of species geographic distributions. *Ecological Modelling*, *190*(3-4), 231–259.

R Core Team. (2016). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Riahi, K., Rao, S., Krey, V., Cho, C., Chirkov, V., Fischer, G., Kindermann, G., et al. (2011). RCP 8.5A scenario of comparatively high greenhouse gas emissions. *Climatic Change*, *109*(1-2), 33–57.

Ridgeway, G. (2006). *gbm: Generalized boosted regression models*. R package version.

Root, T. L., & MacMynowski, D. P. (2005). Human-modified temperatures induce species changes: joint attribution. In *Proceedings of the national academy of sciences* (pp. 7465–7469).

Rosson, M. B. (2002). Scenario-Based Design. In J. Jacko & A. Sears (Eds.), *The human-computer interaction handbook fundamentals, evolving technology and emerging applications* (pp. 1032–1050). CRC Press.

Sadjadi, S. M., Shimizu, S., Figueroa, J., Rangaswami, R., Delgado, J., Duran, H. A., & Collazo-Mojica, X. J. (2008). A modeling approach for estimating execution time of long-running scientific applications. *IPDPS*, 1–8.

Salisbury, E. J. (1926). The Geographical Distribution of Plants in Relation to Climatic Factors. *The Geographical Journal*, *67*(4), 312.

Schaeffer, M., Pierre, S. S., Twigger, S., White, O., & Rhee, S. Y. (2008). *The future of biocuration* (Vol. 455). Nature.

Schatz, M. C., Langmead, B., & Salzberg, S. L. (2010). Cloud computing and the DNA data race. *Nature Biotechnology*, *28*(7), 691–693.

Schimel, D., Keller, M., Duffy, P., Alves, L., & Aulenbach, S. (2009). *The NEON strategy: Enabling continental scale ecological forecasting*. NEON Inc.

Schnase, J. L., Duffy, D. Q., Tamkin, G. S., Nadeau, D., Thompson, J. H., Grieg, C. M., McInerney, M. A., et al. (2014). MERRA Analytic Services: Meeting the Big Data challenges of climate science through cloud-enabled Climate Analytics-as-a-Service. *Computers, Environment and Urban Systems*, 1–14.

Smith, S. E., Mendoza, M. G., Zúñiga, G., Halbrook, K., Hayes, J. L., & Byrne, D. N. (2013). Predicting the distribution of a novel bark beetle and its pine hosts under future climate conditions. *Agricultural and Forest Entomology*, *15*(2), 212–226.

Snijders, C., Matzat, U., & Reips, U.-D. (2012). *“Big Data” : Big Gaps of Knowledge in the Field of Internet Science* (Vol. 7).

Soberon, J., & Peterson, T. (2004). Biodiversity informatics: managing and applying primary biodiversity data. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *359*(1444), 689–698.

Soberón, J., & Peterson, A. T. (2005). Interpretation of Models of Fundamental Ecological Niches and Species Distributional Areas. *Biodiversity Informatics*, *2*(0).

Soberón, J., Arriaga, L., & Lara, L. (2002). Issues of quality control in large, mixed-origin entomological databases. *Towards a global biological information infrastructure*, *70*, 15–22.

Stein, A. F., Isakov, V., Godowitch, J., & Draxler, R. R. (2007). A hybrid modeling approach to resolve pollutant concentrations in an urban area. *Atmospheric Environment*, *41*(40), 9410–9426.

Stein, L. D. (2010). The case for cloud computing in genome informatics. *Genome biology*, *11*(5), 1–7.

Svenning, J.-C., Fløjgaard, C., Marske, K. A., Nogués-Bravo, D., & Normand, S. (2011). Applications of species distribution modeling to paleobiology. *Quaternary Science Reviews*, *30*(21-22), 2930–2947.

Svenning, J.-C., Normand, S., & Skov, F. (2008). Postglacial dispersal limitation of widespread forest plant species in nemoral Europe. *Ecography*, *31*(3), 1–11.

Thomas, C. D. (2010). Climate, climate change and range boundaries. *Diversity and Distributions*, *16*(3), 488–495.

Thuiller, W. (2007). Biodiversity: Climate change and the ecologist. *Nature*, *448*(7153), 550–552.

Thuiller, W., Albert, C., Araújo, M. B., Berry, P. M., Cabeza, M., Guisan, A., Hickler, T., et al. (2008a). Predicting global change impacts on plant species distributions: Future challenges. *Perspectives in Plant Ecology, Evolution and Systematics*, *9*(3-4), 137–152.

Thuiller, W., Albert, C., Araújo, M. B., Berry, P. M., Cabeza, M., Guisan, A., Hickler, T., et al. (2008b). Predicting global change impacts on plant species distributions: Future challenges. *Perspectives in Plant Ecology, Evolution and Systematics*, *9*(3-4), 137–152.

Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2008). A break in the clouds. *ACM SIGCOMM Computer Communication Review*, *39*(1), 50–56.

Veloz, S. D., Williams, J. W., Blois, J. L., He, F., Otto-Bliesner, B., & Liu, Z. (2012). No-analog climates and shifting realized niches during the late quaternary: implications for 21st-century predictions by species distribution models. *Global Change Biology*, *18*(5), 1698–1713.

Vincent, P. J., & Haworth, J. M. (1983). Poisson Regression Models of Species Abundance. *Journal of Biogeography*, *10*(2), 153–160.

Waltari, E., Hijmans, R. J., Peterson, A. T., Nyári, Á. S., Perkins, S. L., & Guralnick, R. P. (2007). Locating Pleistocene Refugia: Comparing Phylogeographic and Ecological Niche Model Predictions. *PLoS ONE*, *2*(7), e563–11.

Williams, J. W., & Jackson, S. T. (2007). Novel climates, no-analog communities, and ecological surprises. *Frontiers in Ecology and the Environment*, *5*(9), 475–482.

Wing, M. G., Eklund, A., & Kellogg, L. D. (2005). Consumer-grade global positioning system (GPS) accuracy and reliability. *Journal of forestry*.

Woodward, F. I. (1987). Climate and plant distribution.

Wu, Q., & Datla, V. V. (2011). On Performance Modeling and Prediction in Support of Scientific Workflow Optimization. In *2011 ieee world congress on services (services)* (pp. 161–168). IEEE.

Yang, C., & Huang, Q. (2013). *Spatial Cloud Computing*. A practical approach. CRC Press.

Yang, C., & Raskin, R. (2009). Introduction to distributed geographic information processing research. *International Journal of Geographical Information Science*, *23*(5), 553–560.

Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., Bambacus, M., et al. (2011). Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing? *International Journal of Digital Earth*, *4*(4), 305–329.

Yang, C., Wu, H., Huang, Q., Li, Z., & Li, J. (2011). Using spatial principles to optimize distributed computing for enabling the physical science discoveries. *Proceedings of the National Academy of Sciences*, *108*(14), 5498–5503.

Yee, T. W., & Mitchell, N. D. (1991). Generalized additive models in plant ecology. *Journal of vegetation science*, *2*(5), 587–602.