CAREER: ABI: Assembling Extremely Large Metagenomes

C. Titus Brown

Monday, July 23rd, 2012

Keywords: assembly, bioinformatics, next generation sequencing, metagenomics,

CAREER: ABI: Assembling Extremely Large Metagenomes

PI: C. Titus Brown
Lead Institution: Michigan State University

Project Summary

Complex microbial populations participate in and drive many biochemical and geochemical processes, ranging from greenhouse gas flux, to nitrogen fixation in soil, to processing of gut nutrients, and beyond [NRC, 2007]. Only recently has it become possible to deeply sample the contents of these populations using next-generation sequence analysis [Mackelprang et al., 2011, Tyson et al., 2004]. Reference-free assembly of these metagenomes is a critical endeavor in modern biology, in part because we have yet to sample even a small fraction of the tree of life, and have no reference genomes for most environmental organisms.

De novo assembly techniques have not kept up with the advances in sequencing. A large class of modern assemblers, de Bruijn graph assemblers, has been developed for the express purpose of short-read assembly and can scale to assemble single human genomes on commodity hardware [Miller et al., 2010, Gnerre et al., 2011]. However, these assemblers are neither designed for nor scale to the volume of data being generated for metagenomes, which can contain many times the novel sequence in genomic samples. Scaling metagenome assembly is an important bioinformatic problem.

Research Objectives: I propose a research plan centered on **combining a compressible graph representation with novel streaming online data reduction and graph analysis algorithms** to provide a general scaling solution to the problem of metagenome assembly. We will combine these novel and existing approaches to develop a **usable reference implementation** that can be applied to existing and emerging sequencing data sets.

Intellectual merits: This project will contribute significantly to *biological understanding* of complex metagenomic samples by enabling the assembly of larger, deeper samples, which in turn will improve the foundation of many biological investigations. It will also provide a new set of *computational approaches* for understanding and taking advantage of assembly graph structures.

Broader impacts: I propose to extend our existing efforts in interdisciplinary bioinformatics education to specifically address the underrepresentation of women and minorities in computational biology at the undergraduate level. This intertwining of education, outreach, and research is already inextricably part of my career, and a strong focus of the BEACON NSF Center here at MSU. We will develop a three-phase program of education, culminating in a co-mentored research experience in biology. Our long-term goal is to increase training, awareness, and participation in bioinformatics and computational biology among undergraduates taken from biology majors at MSU, North Carolina A&T, and elsewhere.

As with our previous work, we will also maximize the utility and reusability of our approaches by: publishing in open-access journals using the ipython notebook ”executable paper” format to maximize reproducibility; making our software available under a BSD-like open source license, on github.com, with automated tests and documentation; providing tutorials and accessible online discussions of our approach; and blogging regularly about our work.

CAREER: ABI: Assembling Extremely Large Metagenomes

PI: C. Titus Brown
Lead Institution: Michigan State University

1 Background and Significance

1.1 Introduction and Overview

Complex microbial populations participate in and drive many biochemical and geochemical processes, ranging from greenhouse gas flux, to nitrogen fixation in soil, to processing of gut nutrients, and beyond [NRC, 2007]. Investigating the ecological and molecular principles underlying their participation in these processes is extremely challenging, because the vast majority of microbes exist in complex ensembles and cannot be cultured or studied in the lab [Sogin et al., 2006]. However, DNA and RNA from these assemblages can be readily extracted, and so one approach that has been developed over the last 15 years is gene-targeted or random (shotgun) sequencing of whole metagenomes [Tyson et al., 2004, Venter et al., 2004]. Random shotgun sequencing of metagenomes is known as "metagenomics"; **metagenomics is essentially the only way to completely characterize unculturable microbial assemblages** [Tringe and Rubin, 2005].

Sequencing technologies are only beginning to scale to the depth of sampling necessary to investigate metagenomic samples with a shotgun sequencing approach. With Sanger and Roche 454, low complexity communities can be investigated thoroughly and inexpensively [Tyson et al., 2004]. Medium complexity samples such as human gut and cow rumen can be sequenced to high coverage with Illumina today [Qin et al., 2010, Hess et al., 2011]. However, higher complexity communities such as soil and seawater possess thousands or millions of species of bacteria and archaea, and may require terabases of sequencing in order to fully sample low-abundance microbes. The complete number and extent of microbial communities is unknown, but some environments are estimated to contain literally millions of species [Gans et al., 2005]; the Earth Microbiome Project proposes to generate petabases (10^{15}) or more of environmental sequence in discovery [Gilbert et al., 2010a].

One approach to metagenomic analysis focuses on targeted gene sequencing via PCR amplification from the community ("pyrotag" sequencing), which is biased towards genes we have already identified
[Sogin et al., 2006]. Another approach analyzes raw sequencing reads using homology search to categorize likely gene content [von Mering et al., 2007]. This approach neither scales well to many reads, nor has high specificity with short reads [Glass et al., 2010, Angiuoli et al., 2011]. Eventually, single cell isolation and sequencing may provide a highly specific way to extract many genomes, but this is extremely challenging for low abundance microbes [Woyke et al., 2010].

De novo metagenome assembly approaches offer a number of advantages [Henry et al., 2011]. Assembly collapses short reads and produces contig sequences containing multiple genes and operons, facilitating computational analysis of putative protein function and ultimately synthetic biology approaches for investigation of metagenomic function [Llewellyn and Eisenberg, 2008, Gibson et al., 2008, Hess et al., 2011]. However, there are a number of associated challenges [Pignatelli and Moya, 2011].

1.1.1 Challenges in metagenome assembly.

Metagenome assembly faces two challenges in an era of large, short-read data sets. The first is *variable abundance of source organisms*, and the second is *sensitivity and scaling*. Variable abundance of the source organisms leads to variation in sequence sampling, which confuses assemblers that use “expected coverage” as a way to trim errors and detect repetitive sequence (e.g. Velvet’s approach, [Zerbino et al., 2009]). This is especially important with the increased depth of sampling necessary for assembly with short read sequencers, and potentially results in smaller and error prone assemblies due to misidentification of repeats. Some attempts have already been made to tackle this challenge using local coverage or graph partitioning [Peng et al., 2011, Namiki et al., 2012, Peng et al., 2011, Pell et al., 2012].

Considerably more work has been done on scaling, with two recent publications on *de novo* assembly of 200 GB+ short-read (Illumina) data sets. The first, on human intestinal tract bacteria (MetaHIT), sequenced samples from 100+ individuals [Qin et al., 2010]. The second sequenced cow rumen gut samples [Hess et al., 2011]. In both cases, very stringent abundance filtering was used to eliminate many reads and scale assembly, and a single set of assembly parameters was used. As no other approach has yet been applied to these data sets, it is difficult to estimate what effect these approaches had on the final assembly.

Sensitivity is coupled to scaling because deep sampling is required to robustly sample rare metagenome members. We have very little concrete idea of what increased sensitivity will bring us, because we are consistently pushing the boundaries by sequencing more and more deeply; it truly is tackling “unknown unknowns”. In metagenomic samples, ribotyping detects 16s rRNA genes that appear to be from distinct species at an abundance of 1 in 100,000 or lower; these are impossible to observe with current sequencing depths, yet these species may be ecologically significant members (the “rare biosphere” hypothesis, [Sogin et al., 2006]). **If metagenomics is to be a foundation for environmental investigations, then sensitive detection of low-abundance population members is critical.** This requires scaling!

1.1.2 Scaling challenges

Another problem with assembly scaling lies in the pace at which new sequencing technologies are being developed. Sequencing technologies are now scaling faster than Moore’s Law, and it is possible to generate sequencing data sets much faster and more cheaply than it is possible to analyze or assemble them. Because of the volume of data needed to sample complex metagenomes, computational hardware improvements are and will continue to fall behind the sequencing curve. Advances in data structures and algorithms are desperately needed for the future, rather than simply increasing hardware capacity.

Scaling is unfortunately quite difficult to achieve, for both theoretical and practical reasons. The primary block to scalability is memory: next-gen sequencing data sets generate enormous assembly graphs, containing billions of nodes. The most common assembly formalism, de Bruijn graphs, relies on exact k-mer matches (fixed-length words of DNA) to implicitly detect overlaps, and hence scales with the number of unique k-mers [Pevzner et al., 2001, Miller et al., 2010]. Because they scale with sample novelty rather than with read number, de Bruijn graphs underlie the majority of recently developed assemblers, including ABySS, Velvet, SOAPdenovo, and ALLPATHS-LG [Miller et al., 2010]; but sufficiently deep metagenome sequencing overmatches these assemblers. For example, both the cow rumen and MetaHIT metagenome projects required $>$ 300 GB of RAM for fewer than 2bn reads [Qin et al., 2010, Hess et al., 2011]. For one soil sample of 3bn reads, a modified Velvet running on the Blacklight large-memory node required over 3 TB of memory (pers communication).

In the immediate future, Pacific Biosciences and other technologies will yield substantially longer reads with sufficient accuracy to be useful [Eid et al., 2009]. Unfortunately, **in addition**

to long reads, we need deep sampling for metagenomics, to detect rare community members and transcripts. Longer reads will give us substantially *better quality* assemblies, but they are not a replacement for algorithmic advances in short read assembly. Perhaps the best future hope is that experimental advances in single cell sorting and sequencing will solve the sensitivity problem for metagenomics, but this technology cannot yet be applied.

1.2 Significance and applications

De novo assembly approaches are a key method of analysis for many samples already being generated. Since Roche, Illumina, and Pacific Biosciences sequencers are within purchasing range of individual sequencing centers, and individual sequencing runs are now in the $10k range, many academic and industrial research groups are generating their own metagenome samples; these groups often struggle to find computational resources and expertise capable of dealing with the sequence [Pennisi, 2011]. In addition, large scale sequencing programs like the JGI's Community Sequencing Program and the Earth Microbiome Project are generating vast amounts of metagenome sequence for environmental samples but provide relatively little in the way of computational support [Gilbert et al., 2010b]. This leads to a significant analysis gap between acquisition of sequence and any sort of analysis that can support hypothesis-driven research.

Scaling de novo metagenome assembly by an order of magnitude or more will be transformative. Scaling *de novo* assembly will enable the use of rental or "cloud" computers (e.g. Amazon Web Services), thus democratizing assembly for small groups [Schatz et al., 2010]. It will also permit more and faster iterations with different parameter sets, improving assemblies. And it will leverage CPU power, enabling more and better graph reduction heuristics to be applied.

The applications of better *de novo* assembly approaches in metagenomics are hard to overstate. Deeper, faster, and better metagenomics will applicable to nearly every facet of evolutionary research, ecological research, medical research, and environmental processes. In particular, more and better reference genomes for metagenomics will enable perturbation studies at an environmental and ecological level and improve agricultural and medical investigations [NRC, 2007].

In tandem with the technological challenge of making sense of sequence, we also face a social challenge: we lack the human capital to take advantage of these amazing opportunities in sequencing. So another challenge is how to grow educational infrastructure to train the next generations of biologists, who will of necessity be part *computational* biologist. This must be addressed with outreach and education.

1.3 Sparse and compressible graph representations

The general area of assembly has received quite a bit of attention in recent years due to the increase in throughput of next-generation sequencers. In particular, string-graph and compression approaches have emerged that decrease the overall memory required for assembly [Simpson and Durbin, 2012]. However, these compression approaches rely on having large amounts of redundancy present in the data set, and underperform on low-coverage data sets like metagenomes from high-complexity environments. Sparse graph approaches are likewise being developed, but these have not yet been applied to metagenomes either [Ye et al., 2012]. Neither approach promises to change the underlying scaling behavior of assembly, however.

Our group has recently published a lightweight probabilistic de Bruijn graph representation that we deployed for the purpose of graph partitioning [Pell et al., 2012]. The underlying data structure we developed is extremely memory efficient, and can store nodes or k-mers in 4-16 bits of mem-

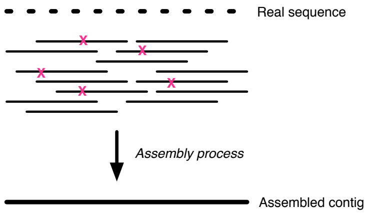

Figure 1: In *de novo* assembly the source sequences (dashed line) are unknown and must be reconstructed using overlaps between the sequencing reads; errors in sequencing (red Xs) complicate this process.

ory. This is 5-10 times better than the best possible exact storage [Conway and Bromage, 2011]. Combined with the partitioning approach, we have been able to gain about a factor of 20x leverage on the problem of metagenome assembly. Despite these significant advances this improvement, too, is insufficient for dealing with the current data volume.

2 Research proposal

2.1 Introduction

The goal of our bioinformatics research is to provide data structures and algorithms that make *de novo* assembly of extremely large and complex metagenomes possible. We are focused on short-read data from Illumina machines because it offers by far the deepest sampling: metagenomics of complex environments such as soil and marine environments depends on sensitive reconstruction of rare genomes of deep sampling [Brown and Tiedje, 2011]. **Our most ambitious goal is to be able to assemble the approximately 50 Tbp, or 500 bn reads, of Illumina sequence needed to thoroughly sample soil** [Gans et al., 2005]. This is currently simply impossible!

In this context, we have been investigating efficient data structures and algorithms for next-gen sequence data analysis for several years. In large part because of the youth of the field – extremely large volumes of sequence data only started to become readily available within the last 5 years – there are many opportunities for improved analysis strategies. Our work has focused particularly on **sketch data structures**, ultra-efficient data structures for representing some subset of information; on **streaming** and **online** algorithmic approaches that can efficiently process large volumes of data; on lossy compression and error detection within large data sets; and on **prefiltering approaches** that scale or improve downstream approaches by **compressing or correcting their input data**, without requiring reimplementing existing software.

We have previously published a compression approach for de Bruijn assembly graphs with which we scaled down memory usage in metagenome assembly by an order of magnitude [Pell et al., 2012]. Below, we describe an orthogonal suite of prefiltering approaches that enable dramatic and substantial increases in analysis efficiency by taking advantage of the deep sampling and well-mixed reads from short-read shotgun sequencing. Our research proposal focuses on combining the published assembly graph compression approach with these in-review prefiltering approaches to develop novel graph alignment, error correction, and online assembly algorithms.

2.2 Specific Aim 1: Develop a novel read-to-graph alignment approach

The goal of assembly is to reconstruct the source genomes in the population; this is generally done by building "contigs", or contiguous DNA sequences, based on overlaps between shotgun sequencing reads (see Figure 1). *De novo* assembly is a notoriously challenging offline problem that generally involves the storage and analysis of large graphs; it has, so far, been viewed as an intrinsically all-by-all analysis problem that is not amenable to distribution to the lack of locality in the graph. The current dominant assembly paradigm is to use de Bruijn graphs to build contigs by breaking reads down into easily hashed fixed-length sequences, or k-mers, that can then be connected by overlap and traversed as a graph [Compeau et al., 2011]. De Bruijn graphs have the clear advantage of scaling in memory usage with the number of k-mers present in the data as opposed to the data set size; unfortunately, de Bruijn graphs are sensitive to errors in the original reads, and often the majority of the k-mers present in de Bruijn graphs are due to errors in the original reads [Conway and Bromage, 2011]. Error detection, removal, and/or correction is thus central to assembly approaches, and many packages exist for doing error trimming or correction (see references in [Kelley et al., 2010]). However, metagenomes cannot use error detection and correction approaches developed for genomic samples, and so there is only one error analysis approach currently being used [Keegan et al., 2012].

Below, as our first specific aim, we propose to develop a read-to-graph alignment algorithm that will enable computationally efficient error correction for metagenomic data sets, with the significant benefit of improving an existing data reduction approach. Underpinning this algorithm are several novel approaches currently either submitted for publication or in preparation, including: a memory efficient data structure for counting k-mers; a streaming data reduction algorithm for assembly; a streaming error detection algorithm for metagenomes; and HMM-guided graph search.

2.2.1 Preliminary results: A memory-efficient data structure for counting k-mers.

We have developed and implemented an efficient probabilistic data structure for counting k-mers. The counting method is based on our successful use of Bloom filters to represent de Bruijn graphs with high efficiency [Pell et al., 2012], and is essentially identical to a CountMin Sketch or Counting Bloom filter [Cormode and Muthukrishnan, 2005]. Briefly, to increment the count for a k-mer, we hash it into multiple hash tables, and increment the corresponding entry in each table; then, to retrieve the count for a given k-mer we select the minimum count across all of the hash entries. This results in a very memory efficient data structure, albeit one with slightly incorrect counts: if multiple k-mers hash to the same location, then an incorrect count may be retrieved for each of those k-mers. Nonetheless, because NGS data sets are dominated by low-count k-mers from sequencing errors, these miscount values are generally low [Melsted and Pritchard, 2011]; also (Zhang and Brown, in prep). Both theoretically and practically our implementation is substantially lower memory than other k-mer counting implementations [Pell et al., 2012, Brown et al., 2012].

Our k-mer hashing, counting, and graph analysis implementation is written in C++ and wrapped in Python, and available on github under a BSD license: github.com/ged-lab/khmer/. It has been used in [Pell et al., 2012] as well as digital normalization, discussed below, and is currently in use by several dozen groups. The khmer package comes with documentation, tutorials, and automated tests (Zhang and Brown, in prep), and has a community mailing list.

2.2.2 Preliminary results: A streaming data reduction approach for assembly

Driven by the need to assemble multiple very large metagenomic data sets, we next developed a single-pass streaming algorithm for lossy compression of Illumina data prior to assembly (in review) [Brown et al., 2012]. Our approach, which we term “digital normalization”, relies on an online construction of a de Bruijn graph to select a subset of informative reads using locus-specific graph analysis. The key observation that lies behind digital normalization is that most short-read

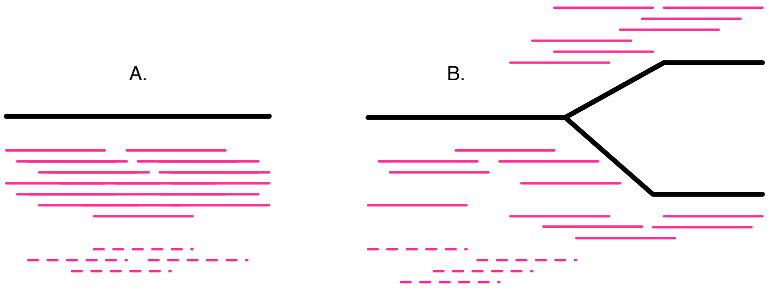

Figure 2: Digital normalization is an efficient approach for locus-specific downsampling of shotgun data sets. Given reads (short red lines) from multiple unknown source sequences, digital normalization chooses reads that provide coverage for independent loci up to a specified point (here, a coverage of 5; solid red lines) and discards reads past that coverage as redundant (dashed red lines). This allows less expensive reconstruction of the source sequences by using a only subset of the data. Both (A) simple and (B) more complex graph structures are retained by digital normalization.

sequencing data sets are massively redundant, because they contain many reads that have been sampled from overlapping locations; by using a simple estimator of sampling depth, we can detect and eliminate much of this redundancy online. Our current implementation relies on a simple fixed-memory estimator of coverage, median k-mer count per read; see [Brown et al., 2012] for details.

Digital normalization ("diginorm") is an extraordinarily effective data reduction technique for assembly. On a typical 100x E. coli sequencing sample for which 100x coverage has been obtained (e.g. 5m Illumina reads of length 100), we can eliminate 95% of the reads prior to *de novo* assembly and achieve an essentially identical assembly based on the remaining 5x coverage. For high-coverage mRNAseq and single-cell sequencing data sets, data reduction rates approach 98% or more, due to the presence of many very high abundance components in the data set; see [Brown et al., 2012] for details.

We have used digital normalization successfully on microbial and mRNAseq data sets [Brown et al., 2012], as well as on small and large metagenomes (Howe and Brown, in prep; Howe, Tiedje and Brown, in prep). Our crowning achievement thus far has been to reduce a 300 Gbp soil metagenome assembly to needing only 300 GB of RAM from the original 3 TB of RAM when combining digital normalization and partitioning (Howe, Tiedje Brown, in preparation; discussed below).

2.2.3 Preliminary results: A streaming few-pass error detection method based on digital normalization.

We have further adapted the digital normalization algorithm to detect and trim likely sequencing errors within short-read data sets. Our current method builds on the error elimination pass described in the diginorm paper, in which low-abundance k-mers are eliminated after normalization to a specific coverage. Essentially, we combine the online graph-based coverage estimation used by digital normalization with the k-mer abundance approach commonly used to detect errors [Pevzner et al., 2001, Kelley et al., 2010].

We have implemented a $<$ 2-pass approach that uses the diginorm algorithm to determine when a particular region of the de Bruijn graph has accrued enough coverage for error correction to be applied (Figure 3). Once a region of the graph has 20x coverage, then reads from this region that

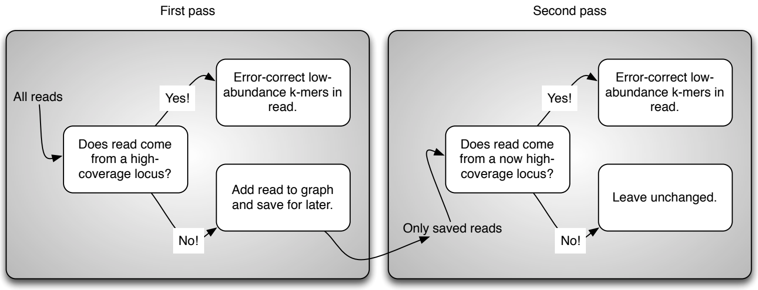
First pass
All reads
Does read come from a high-coverage locus?
Yes!
Error-correct low-abundance k-mers in read.
No!
Add read to graph and save for later.
Second pass
Only saved reads
Does read come from a now high-coverage locus?
Yes!
Error-correct low-abundance k-mers in read.
No!
Leave unchanged.

Figure 3: Few-pass error detection and correction. In a first pass, reads that add coverage to a locus below a specified threshold are loaded into the graph but not error-corrected, for lack of sufficient information; reads that belong to already high-coverage loci are immediately error-corrected. In the second pass, those reads that were used to build the graph are inspected to see if their source loci have sufficient coverage to error correct them. For a 100x coverage genomic shotgun sample with a read collection threshold of 20, this approach would be approximately 1.2-pass: about 20% of the reads would be collected on the first pass, and then examined and error-corrected on the second pass.

contain k-mers with abundance of 1 are likely to be erroneous and can be trimmed or (potentially) corrected. Because we must observe some minimum number of reads in order to accumulate sufficient coverage to do error detection, at least some of the reads must be analyzed more than once, unlike diginorm; however, for deep data sets, the majority of reads will only need to be seen once, making it a < 2 pass approach.

Interestingly, this algorithm provides solutions for a number of the problems with existing error correction approaches: it does not rely on uniform coverage to determine k-mer abundance cutoffs, and so can be used on metagenomic data, and it is an online algorithm that does not require two passes over the data. Moreover, because we have implemented the approach using khmer's CountMin Sketch data structure, it remains extremely efficient in memory use.

2.2.4 Preliminary results: HMM-guided graph search

In collaboration with Jordan Fish and James Cole of the Ribosomal Database Project at MSU, we have developed an profile Hidden Markov Model-guided graph traversal and assembly system called HMMgs (Fish et al., in preparation). Rather than attempting to assemble paths around systematic Hamiltonian or Eulerian graph traversal, HMMgs uses a Dijkstra-like graph search strategy to find paths that best match a specific Hidden Markov Model. HMMgs uses an A* search algorithm to search all possible protein translations of a DNA de Bruijn graph, weighting paths using a heuristic cost function that is guaranteed to never overestimate the actual score.

HMMgs is essentially a gene-targeted assembler that uses HMMs to extract the DNA sequence reads underlying high-scoring protein sequences. Because it uses the probabilistic de Bruijn graph developed in [Pell et al., 2012], it is lower memory than any of the existing *de novo* assemblers; more importantly, preliminary results indicate that it is both more sensitive and specific than *de novo* assemblers, because of the additional information provided by the HMM guide. Below, we propose to modify this search algorithm to align input sequence reads to a DNA graph.

2.2.5 Subaim 1a: Develop a read-to-graph sequence alignment approach

Using a CountMin Sketch [Cormode and Muthukrishnan, 2005] and the probabilistic de Bruijn graph framework developed in [Pell et al., 2012], we will develop a read-to-graph alignment algorithm using a Dijkstra-like graph search strategy. To do this, we add vertices to our probabilistic de Bruijn graph in order to account for indels and SNPs. Insertions, deletions, and mismatches are penalized according to a basic nucleotide scoring matrix. Because Dijkstra-style algorithms cannot handle edges with negative weights, we assign edges toward SNP/indel/mismatch versions of k-mers to have higher weights than edges toward matches between the graph and read. Searches are started by using an exact k-mer seeding strategy based on matching k-mers between the graph and the read; from there, we search for the lowest-cost path. Because the Principle of Optimality is maintained, we are guaranteed to find an optimal alignment based on the scoring matrix.

To optimize the graph search, we will add a heuristic cost function to Dijkstra's algorithm (i.e. A* search) that compares possible paths based on number of mismatches and indels. Furthermore, vertices will be pruned based on an allowable number of mismatches determined by the length of the read of interest. Finally, we will incorporate coverage level information into the algorithm by preferentially choosing paths that match the expected k-mer coverage level. We will initially develop a scoring matrix based on aligning Illumina reads to the graph; Illumina reads are short but relatively accurate, with a 1% substitution error rate and very few indels. From this, for each read we will generate read-to-graph alignments that yield an estimated path coverage for each corrected read.

The resulting alignment approach will be immediately useful for two very interesting purposes: improved digital normalization and error correction.

Application to digital normalization: Read-to-graph alignment can be used to estimate per-read coverage more accurately than the estimator currently being used, median k-mer count in the read [Brown et al., 2012]. The median k-mer count is efficient to calculate, but has several drawbacks, including an inability to determine when k-mers are missing due to undersampling, as well as a propensity to retain highly erroneous reads as novel [Brown et al., 2012]. By using a read-to-graph alignment approach and then calculating the path coverage for the aligned read in the graph, we will be able to more accurately determine whether or not a read contains novel information and hence should be kept as part of the diginorm algorithm. This will better retain undersampled portions of the graph as well as discarding reads with many errors, significantly improving digital normalization.

Application to error correction: A read-to-graph alignment approach will also serve as a general error correction strategy. Given such an alignment approach, we can adapt the diginorm-based error-trimming strategy described above to return corrected reads, rather than truncating them at low-abundance k-mers. Thus this first Subaim would directly yield a few-pass fixed-memory error-correction approach for Illumina reads. Moreover, this error correction approach would apply to metagenomic data, unlike existing error correction approaches.

So, for this Subaim, we will implement a read-to-graph alignment based on an A* graph search algorithm; integrate it into the digital normalization algorithm; and extend our existing error trimming software to output the error-corrected read, i.e., the alignment.

For evaluation of the alignment algorithm, we have a plethora of simulated and real data to use, including: the data from [Pell et al., 2012] and [Brown et al., 2012]; simulated metagenomic data from high-complexity samples [Pignatelli and Moya, 2011]; and a Human Microbiome Project "mock" data set consisting of real sequencing done from a mock assembly of known microbes [HMP, 2012]. For all of these data sets we know the ground truth and can evaluate both sensitivity and specificity of the alignment.

The main challenge we expect to face is with respect to reads from repetitive sequences, which will be alignable but could nucleate misassemblies if corrected. As part of another grant, we are extending digital normalization to use paired-end and mate-pair reads to identify and elide repeats. Moreover, in practice metagenomes contain relatively few repeats, and those that are present tend to "break" contig assembly and thus require scaffolding to assemble ((Howe and Brown, in prep) and see [Iverson et al., 2012]).

2.2.6 Subaim 1b: Build scoring matrices and alignment models for long reads

We will expand our initial graph alignment algorithm to do read-to-graph alignment of sequence reads from other sequencing machines. For example, both Roche 454 and Pacific Biosciences SMRT sequences generate significantly longer reads than Illumina, albeit with higher error rates. In particular, PacBio reads can be up to 20kb, but have a $\approx 15\%$ error rate, with high indel rates. This will require building different scoring matrices and alignment models that match the error rates of these other sequencers. Note that this is not a novel idea – in [Koren et al., 2012], an algorithm was introduced that successfully corrected PacBio reads to better than 99.9% accuracy. However, their approach is notably heavyweight, and moreover not suitable for sequences from metagenomes.

We will train a scoring matrix for handling alignments between Illumina and PacBio reads using the data sets in Koren et al. as training data. Then, we will use the trained scoring matrix to correct PacBio reads and compare speed, accuracy, and memory footprint with the results obtained in [Koren et al., 2012]. Given the low memory footprint and streaming nature of our algorithms, we hope for a 10-100x improvement in both memory and speed in a non-parallel implementation. In addition, we note that our data structures and algorithms are straightforward to parallelize and rely on only a simple hash table data structure with virtually no access contention.

2.2.7 Specific Aim 1: Concluding thoughts

The results from Specific Aim 1 will include an implementation of a novel algorithm for aligning reads to graphs, as well as derivative approaches that can do computationally efficient error correction of both short Illumina reads and longer PacBio reads. It will also yield significant improvements to the basic digital normalization algorithm. Error correction and improved digital normalization will significantly decrease the error rate and improve sampling retention for metagenomic samples. These are important (although not critical) to Specific Aim 2, which focuses on online partitioning of metagenome data.

2.3 Specific Aim 2: Online assembly of metagenomes through partitioning

In [Pell et al., 2012], we developed and implemented a partitioning approach to metagenome assembly that used de Bruijn graph connectivity to separate reads into disconnected "bins" (see Figure 4). This approach resulted in identical assemblies after partitioning for our test data set, and, since it relied on a novel compressible graph representation, scaled metagenome assembly by a factor of about 20x. This proof of concept has been followed by several additional assemblies, described below.

We have also developed a separate streaming data reduction algorithm, digital normalization (described above), that eliminates redundant data from shotgun sequencing. This approach scales assembly data set size and memory requirements to the underlying genomic content, which makes most assembly problems very tractable. However, because metagenomes have such immense underlying genomic complexity, digital normalization gives us less leverage on complex metagenomes

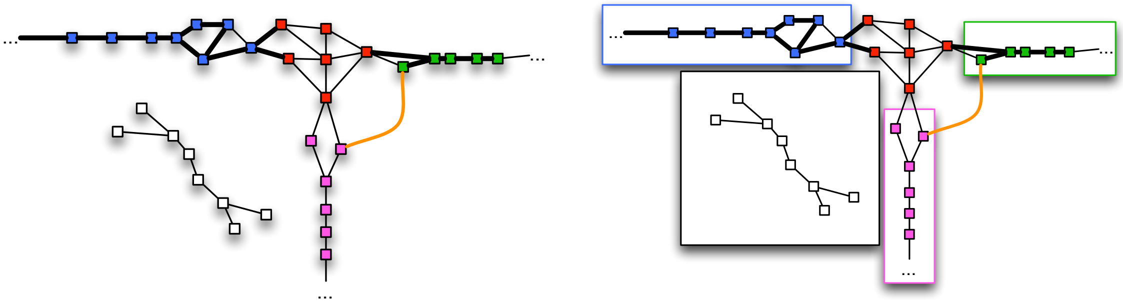

Figure 4: **Left:** A simplified de Bruijn assembly graph for a metagenomic sample containing four distinct organisms (white, blue, green, and purple nodes) in high (blue, green) and low (white, purple) relative abundances. The purple and green organisms share a highly conserved gene and hence their graphs are connected (orange line). The graph also contains erroneous nodes generated by Illumina sequencing bias (red) that artificially connect three of the organisms. The white organism does not share any k-mers with the other organisms and hence does not connect. **Right:** Partitioning separates organisms by connectivity and can split based on complex graph structures.

than on other samples: for 3 bn soil reads, fewer than 50% of the reads have a coverage higher than 10.

For our second specific aim, we propose to combine partitioning and digital normalization to build an *online* partitioning algorithm that partitions regions of graphs once they pass a minimum coverage threshold. This will enable progressive and hence effectively infinite assembly of metagenomes. In addition to using our published partitioning and in-review digital normalization approaches, we also rely on some preliminary results from partitioning and assembling several large metagenomes, discussed next.

2.3.1 Preliminary results: partitioning metagenomes

Our published results include a demonstration that partitioning works on real data, with impressive and substantial memory reduction [Pell et al., 2012]. We have since expanded these results to a variety of additional data sets, including the HMP mock data set [HMP, 2012], the permafrost soil data from [Mackelprang et al., 2011], and the rumen data set [Hess et al., 2011], as well as several extremely large soil metagenomes totaling 2 Tbp of sequence (unpublished).

A persistent problem in all of the large data sets has been the existence of a single, large, connected partition, discovered due to our reliance on partitioning. This partition, which we call the "lump", is caused by real connectivity from highly conserved genes, as well as spurious connectivity caused by sequencer bias towards certain k-mers (red nodes and orange edge, Figure 4). The existence of this partition limits the utility of partitioning, which relies on breaking the data set down into many small partitions. We have developed several approaches to breaking up this lump, including a systematic graph traversal that finds and eliminates highly-connected k-mers, as well as a diginorm-based heuristic elimination of high-abundance k-mers (which are necessarily highly-connected). On simulated data, we obtain assemblies that are 97% identical after applying these techniques. This "delumping" comparison is the subject of a soon-to-be submitted paper (Howe and Brown, in prep.)

2.3.2 Preliminary results: assembling rather large metagenomes

We have applied partitioning and digital normalization to several real data sets (Howe, Jansson, Tiedje, Brown, in preparation). In particular, we have reassembled the Human Microbiome Project (HMP) data set [HMP, 2012], and been able to do so in approximately 5x less memory than SOAP-denovo with comparable results; we have also assembled two soil metagenomes, one from agricultural soil and one from native prairie, both sequenced as part of the DOE Great Prairie Grand Challenge project. These assemblies were each completed in under 15 days in 300 GB of RAM, in comparison to other approaches which required 3 TB of memory and several weeks; the two assemblies resulted in a total of 6 Gbp of assembled contigs $>$ 300 bp in length, containing an estimated 5 million protein coding genes. We are still working to analyze the resulting contigs.

We believe that our current approaches (digital normalization and partitioning) will scale enough to allow the assembly of a total of 1-2 Tbp of data within 1 TB of RAM, but we require approximately 50 Tbp of data for each gram gram of soil [Gans et al., 2005]. New approaches are needed!

2.3.3 Specific aim 2a: Combine digital normalization and partitioning data structures

We implemented both digital normalization and partitioning within a single software package, khmer, but the data structures used for the different algorithms are distinct: the partitioning data structure uses 1 bit (presence/absence) in each hash table for each k-mer, while the counting data structure uses a byte. The ancillary data structures required for partitioning, including tagging and graph traversal, are also not available on the k-mer counting data structure. We will provide a single interface built on top of variable-sized bins such that both partitioning and digital normalization can use the same underlying data structure. This interface will also let us implement exact counting and graph storage data structures underneath, as well as other probabilistic data structures, e.g. the dynamic d-left Counting Bloom Filter [Bonomi et al., 2006].

Specifically, we will integrate the partitioning and digital normalization code in khmer to use the same underlying data structure and class implementation. To evaluate the correctness of our implementation, we will ensure that our current automated test framework continues to run, and use example data as well as code coverage and branch analysis to target new tests and verify that the tests cover the majority of the code. We will also replicate the results in [Pell et al., 2012] and diginorm [Brown et al., 2012] using the new data structures.

2.3.4 Specific aim 2b: Build an online partitioning algorithm

De Bruijn graph-based *genome* assembly is intrinsically all-by-all: there is relatively little locality in the assembly graph, and so distributing graph nodes across multiple machines tends to result in only incremental performance gains [Schatz et al., 2010]. However, *metagenomes* differ from genomes in two important ways: first, metagenomes consist of many distinct genomes, which permits exact or heuristic partitioning based on graph connectivity; and second, metagenomes contain genomes with varying abundances, some high and some low. This latter characteristic means that with shotgun sequencing, which randomly samples from the metagenome, sequences from high abundance components will arrive more frequently than sequences from low abundance components. In turn, this allows us to extract partitions from high abundance genomes from small subsets of the data. (This abundance variation is used in an offline approach developed by the Banfield lab at UC Berkeley (Banfield, pers. communication).)

We propose to combine partitioning with digital normalization to extract partitions from metagenomic shotgun sequence data in a streaming model. More specifically, we will use digital normalization to detect when local regions of the de Bruijn graph reach a high coverage threshold, at which point we will then run the partitioning algorithm to extract graph components connected to the high coverage region. Algorithmically, we will enter each sequence read into the "tagging" data structure used for partitioning [Pell et al., 2012], and also estimate their local graph coverage with diginorm. When a read's coverage is sufficiently high (≈ 20), we will trigger the partitioning step from Pell et al., in which the local graph is explored using the tags to track connectivity. Once all of the connected tags have been discovered, we will re-iterate over the reads collected thus far to extract all of the reads that belong to the partitioned tags. These reads can then be assembled separately.

Specifically, we will load reads into the graph using the digital normalization algorithm. Whenever a read is rejected due to high local graph coverage, we will execute the partitioning algorithm from [Pell et al., 2012], which does a breadth-first search to discover closely connected sequences in the graph. The transitively connected set of graph tags, aka the "partition", will then be used to retrieve reads from sequence data set for assembly.

We will evaluate this approach on our existing test data sets [Pell et al., 2012, Brown et al., 2012], as well as on simulated metagenomic data [Pignatelli and Moya, 2011] and the Human Microbiome Project "mock" data set [HMP, 2012]. In all of these cases, we have already executed complete partitioning and can compare the results of our online approach on these partitioned data sets, where they should match exactly.

2.3.5 Specific aim 2c: Integrate lump removal into partitioning

In our current implementation, partitioning is done without regard to graph complexity or k-mer abundance; removal of the lump is done in a preprocessing step (Howe and Brown, in preparation). We will implement lump "breakup" within the partitioning algorithm.

Our current lump breakup is done in two ways: first, we apply a heuristic that removes k-mers that are more abundant than the local graph coverage, since these k-mers definitely nucleate highly connected regions; and second, we exhaustively find k-mers that are highly-connected by tracking the number of tags each k-mer is connected to; k-mers that are within the tag density T of many tags are in a complex region of the graph, and are used to stop traversal upon partitioning. The first step is heuristically efficient and the second step is exhaustive, and together they *guarantee* lump removal (Howe and Brown, in preparation). Moreover, the majority of these removed k-mers are not used when assembling unfiltered data, as current assemblers tend to break paths on high-complexity regions.

We will move these approaches directly into the partitioning algorithm. Specifically, we will track over-abundant k-mers when loading reads into the graph, and use them to break partition traversal; moreover, as we partition, we will identify regions with many closely connected tags, and traverse from them to identify the highly-connected k-mers. As we have applied these lump breakup approaches to several different data sets (see Preliminary Results, above) we have a number of test data sets on which to evaluate our implementation.

2.3.6 Specific aim 2d: Break traversal on abundance variation

Our current partitioning approach does not store or examine coverage variation in the graph, and partitions solely on connectivity. This can cause problems where multiple closely related strains exist at different abundances (analogous to splice variation in mRNAseq [Grabherr et al., 2011]). After integrating k-mer counting into partitioning (Aim 2a), we will be able to break partitions where significant differences in coverage exist, as in MetaVelvet or Meta-IDBA [Namiki et al., 2012, Peng et al., 2011]. Intriguingly, this means that repetitive sequence will be partitioned well in advance of other sequence, which may have useful downstream applications for e.g. CRISPR analysis of complex metagenomes.

Specifically, we will implement the heuristic developed in MetaVelvet to further break the graph into partitions based on graph-local coverage variation [Namiki et al., 2012]. This will result in improved extraction of partitions from the data, including more accurate partitioning and downstream assembly. We will evaluate this on the MetaVelvet and Meta-IDBA test data sets, as well as on the HMP mock data set with skewed abundances [Namiki et al., 2012, Peng et al., 2011, HMP, 2012].

2.3.7 Specific aim 2e: Implement efficient sequence retrieval code based on tagging

The partitioning approach developed above (Subaim 2a-2c) will yield a bunch of tagged k-mers that can be used to extract all reads belonging to a partition. Currently this requires a complete pass over the reads. However, sequence retrieval can be done more efficiently if we index the reads during our first pass and then use reverse indices to connect the tagged k-mers to their origination reads.

We will therefore integrate a reverse indexing system into our existing simple database for forward indexing of reads (see the screed software package, github.com/ged-lab/screed/). This reverse indexing will rely on a basic on-disk hash table, although we will seek more optimized data structures as needed.

2.3.8 Specific aim 2f: Filter incoming sequences

The overall goal of this Specific Aim is to enable efficient progressive partitioning. As part of this goal, we must efficiently be able to filter out incoming sequences if we have already partitioned and assembled the regions from which they come; if we can do this effectively, we can progressively and dramatically reduce the amount of data being stored and partitioned as more and more data is seen. The main challenge is to minimize or even reduce memory usage as our online partitioning algorithm continues.

There are a variety of options for filtering incoming sequences, and it is difficult to predict which option will work best, so we propose to implement several. For example, ew sequence reads can be searched against assembled sequences, or a graph derived from assembled sequences. This is effectively the same as using digital normalization as a reference-based prefilter, with the addition of periodically updating the reference. The drawback to this approach is that its efficiency is dependent on the quality of our progressive assemblies from Subaim 2a, which may vary with the complexity of the source organisms. Alternatively, new sequence reads can be assigned to existing partitions based on tag membership, i.e. the partition tags used to extract reads into partition bins can be used to preprocess new sequence reads. These partitioned reads can then be independently normalized, error-corrected, and assembled to improve the partitioned assemblies.

More generally, we face the problem that we must remove or expire already-assembled regions of the online graph, which cannot be done with our current Bloom filter implementation. The simplest approach to solving this problem is to periodically reload the online graph from the unpartitioned reads, but this is inefficient; therefore we will also investigate the dynamic d-left Counting Bloom Filter for graph storage, which will permits node removal [Bonomi et al., 2006].

2.3.9 Specific aim 2g: Combine partitions after extraction

The use of a coverage-based partitioning heuristic will inevitable result in incomplete partitions due to partial sampling and normal deviations from average coverage: that is, partitioning will be primarily executed on those regions that are highly sampled by chance, which will in turn result in fragmented assemblies after 2(e). This can partly be remedied by using a high coverage cutoff for the normalization step, together with a large tolerance for deviation from the average for partitioning, but this will be inefficient for large, low-coverage samples. (Note that this is not a problem for MetaVelvet, which uses an offline algorithm which takes into account all of the data at once [Namiki et al., 2012].)

The tagging approach developed for our partitioning approach can also be retained with the partitioning process, and can be used to connect partitions after the fact [Pell et al., 2012]. Therefore, when we extract partitions, we will also retain the tags in those partitions and periodically reconnect partitions based on these shared tags. In practice this may require a heuristic approach to determine when the coverage of a given pair of partitions matches closely enough to examine their overlap.

2.3.10 Specific aim 2: Additional ideas

There are a number of possible extensions of these aims that, should we have enough time, we can work on. For example, we could integrate genome-scale scaffolding for metagenomes as in [Iverson et al., 2012]; estimate the abundance of a partition during the digital normalization step to do online abundance counting; and work on multicore integrated circuit optimization of our algorithms. There are also many heuristic approaches that could be applied, including only tagging high-coverage k-mers prior to partitioning, and expiring nodes from the de Bruijn graph as a way to avoid accumulating errors.

2.3.11 Summary of Proposed Research

In summary, we propose to implement a novel read-to-graph alignment approach, and use it for improved digital normalization and a novel error correction algorithm for sequencing data. We also will develop an online algorithm for partitioning that will allow extremely large metagenomes to be assembled – something that is not possible today.

2.4 Broader Impacts and PI's Research Plan

The PI's group is both generating their own data and collaborating with a number of other researchers on data analysis. Metagenomes are by far the most intractable of the data sets we analyze, due to the size of the data sets. The research plan described above proposes to tackle this, while providing ancillary outputs that will impact other fields (e.g. our error correction approach will also work for genomic sequences and mRNAseq; online partitioning will also apply to mRNAseq).

2.4.1 Software development, "executable papers", and reproducible research

Dr. Brown has been developing software for over 25 years in commercial, academic, and open source environments; he is the author of a number of open source packages, including several utilities for automated testing in Python. Our group uses version control, automated tests, code coverage analysis of the tests, and continuous integration practices as a daily part of software development, in an attempt to ensure code accuracy. We also teach this mode of development in regularly held workshops at MSU and elsewhere.

In the interests of reproducible science, we have also been making our complete source code and data sets available via github.com and Amazon Web Services such that anyone can replicate our results with their own virtual machine. We have also begun to distribute IPython notebooks [Pérez and Granger, 2007] for data analysis and figure generation, which makes our papers "executable", in the sense that all of the primary computational analyses can be rerun with only a few commands (see [Brown et al., 2012] for an example).

2.5 Outreach, education, and training

Our lab engages in many outreach, education, and training activities. Specifically,

- Dr. Brown runs the yearly MSU Analyzing Next-Generation Sequencing Data workshop. This course is run for grad students, postdocs, and faculty, and has attracted over 300 applicants in 3 years. It is currently funded by an NIH R25 grant (through 2013). Our goal in this workshop is to provide biologists with an opportunity to learn basic bioinformatics skills. The course materials, at http://ged.msu.edu/angus/, are available under a Creative Commons license permitting remixing and adaptation, and we provide instructions on how to reuse them. Approximately 60,000 unique visitors per year visit the primary site, according to Google Analytics.

- For the past two summers we have participated in the Summer Research Opportunity Program for underrepresented minorities. In 2011 we trained two UMs in bioinformatics, and in 2012 we trained three. For 2013 we are proposing a more general model in which we will train a large group of SROP undergrads in general computation, supplanting an ineffective statistics workshop; the students will then go on to their individual research labs, but with regular touchback meetings with our group. This grant will support several such undergrads (see Budget Justification and Outreach/Education component).

- Dr. Brown runs an introductory graduate course on computational science for biologists under the auspices of the MSU BEACON NSF STC on "Evolution in Action". The course is teleconferenced across U. Idaho, UW Seattle, and UT Austin. This course includes a module focusing on sequence analysis considerations for evolutionary biologists and ecologists, where software including our tools and approaches are discussed, demonstrated, and then executed by the students.

- Members of the lab, including the PI, regularly participate in "Software Carpentry" workshops focusing on computational science skills (http://software-carpentry.org/). These workshops are held at MSU and elsewhere.

- All publications in the lab are posted to github.com/ged-lab/ and arXiv.org upon submission, and we intend to publish them all as Open Access. All source code in the lab is made available via the github.com/ged-lab/ version control archive, under the Open Source BSD license that permits maximal use and reuse of the code.

3 Results of Prior NSF Support

Project Proposal Title: Symbiont Separation and Investigation of the Novel Heterotrophic Osedax Symbiosis using Comparative Genomics; NSF 09-23812 (PI Brown); Project Location: Michigan State University; Total Award Amount: $50k; Starting Date: 01/01/10; Ending Date: 12/31/12.

This project is a collaborative project with Dr. Shana Goffredi at Occidental College, in which we are analyzing sequence from MDA-amplified metagenomic samples. These samples originated from a bead-based enrichment of *Oceanospirallales sp.* symbionts taken from an *Osedax* bone eating worm. We received our first Illumina samples in May, and applied the digital normalization technique described above to the data sets. We obtained an est. 85% complete genome assembly of the desired microbe, a better than 2-fold increase over a pre-normalized result. A manuscript on the metabolic analysis of these microbes based on their genome content is in preparation.

3.1 Personnel/Work summary and breakdown

Dr. C. Titus Brown (the PI) will supervise all software development, research, and outreach aspects of this grant. We also request support for one graduate RA from Computer Science, who will lead the theoretical and computational aspects of the research as well as coordinate the outreach effort. For years 1-2 this will be **Jason Pell**, who has done most of the graph theory and percolation analysis for our compressible graph data structure, and also worked on digital normalization. For years 3-5, we will recruit a new CSE graduate student.

3.2 Timeline

Year 1: Read-to-graph alignment, use in diginorm, and initial error correction (1a).
Year 2: Long-read error correction (1b) and combining diginorm and partitioning (2a).
Year 3: Online partitioning (2b) and graph analysis (2c).
Year 4: Online partitioning broken by abundance variation (2d) and read retrieval (2e).
Year 5: Filtering incoming sequence and partitioning optimization (2f and onwards).

4 Outreach plan

I propose to target the bioinformatics gap that exists between computer science and molecular biology/genomics by building an undergraduate summer research program that combines education with research in biology, genomics, and bioinformatics. There are two gaps to be closed. First, there is a substantial underrepresentation of women and minorities in bioinformatics. This is, at heart, a "supply" problem: the dearth of women and minorities in computer science undergraduate studies leads to an underrepresentation in CS-derived courses of study. Second, there is a lack of students conversant in both CS and biology.

These gaps introduce an opportunity for training undergraduates in practical research issues. There are many basic computational research tasks that are readily accessible with little training and some hands-on guidance. These research tasks include assembling and annotating microbial genomes, BLASTing large sequence collections for matches to given proteins, and mapping short reads to existing genomes for resequencing analysis. With appropriate mentoring, these tasks provide an easy opportunity for undergraduates to engage in real research.

I propose to use my unique position as a bridge between the CSE and Microbiology departments, and my role as an educator in the BEACON NSF Science and Technology Center on the Study of Evolution in Action, to address these gaps. My teaching and education activities at MSU have already focused on this at both the undergraduate and graduate level. I was hired to help bridge the biology and computational programs here, and have already initiated a number of courses at the graduate level or above. I developed an interdisciplinary graduate seminar course in bioinformatics that achieved an approximately 50/50 split between 25 students from bio and CS majors in its

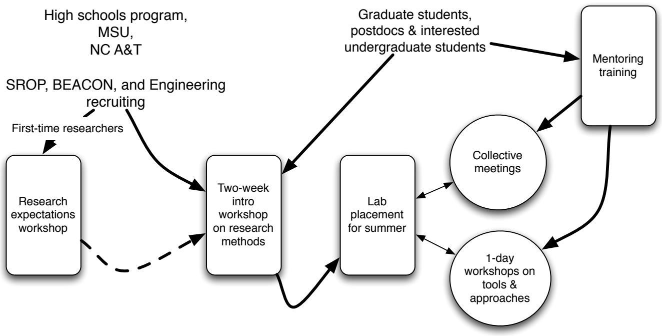
High schools program,
MSU,
NC A&T
SROP, BEACON, and Engineering recruiting
First-time researchers
Graduate students, postdocs & interested undergraduate students
Mentoring training
Collective meetings
Research expectations workshop
Two-week intro workshop on research methods
Lab placement for summer
1-day workshops on tools & approaches

Figure 5: Proposed outreach plan.

second year; I also developed "Computational Science for Evolutionary Biologists", a grad course taught across three of the BEACON campuses (two via teleconference, UT Austin and UW Seattle) two years ago, and across four campuses (U. Idaho) this last year. I am also the founding course director for a summer workshop for career biologists on Analyzing Next-Generation Sequencing Data that attracted over 130 applicants (including 15 faculty, 50 postdocs, and 50 graduate students) in 2011, and over 165 applicants in 2012, including 22 tenure-line faculty. Both years, the student body was more than 60% female. In all of these educational situations I have developed approaches to gently introduce biologists to computation, and have done so with consistently high student evaluations.

I will develop a three-tier program in mentoring. First, I will create a summer program that recruits sophomore women from the Genomics and Molecular Genetics undergraduate major at MSU and sophomore students from NC A&T. This summer program will start with a two-week workshop on basic computational research skills. Second, I will place students in biology labs with computational projects. During this they will be co-mentored by members of my lab. Third, in a summer series of 1-day workshops, we will delve further into bioinformatics. In doing so we hope to interest them in additional computational or mathematical education in their coursework, e.g. a CS minor. Finally, graduate students and postdocs from my lab – most of whom have transitioned between disciplines themselves – will be given formal mentoring training and will engage in mentoring and teaching activities, providing role models for the undergraduates and gaining valuable career experience.

By giving women and other underrepresented minorities a strongly guided research experience in bioinformatics the summer before they become juniors, we will position them well to capitalize on their experience through the next summer, when they will be exploring career options in research or industry. Moreover, in year 2, we expect to invite interested and successful "graduates" from the first summer program to participate in both the education and research programs in the second summer. Note that one potential source of students is a high school training effort that just started this past summer; the BEACON program is developing follow-on programs and has asked me to help coordinate them.

Implementation: The BEACON NSF STC and the Summer Research Opportunities Program (SROP) will provide administrative support and help coordinate administration and funding. Several of my graduate students, including Jason Pell, and my postdoc Adina Howe, will coordinate and teach the two-week workshop, based on exercises derived from my existing fall course. Once the summer undergraduate students are placed in research labs, we will hold weekly meetings collectively to discuss progress and tackle problems as a group. We will also conduct a number of 1-day intensive compute workshops to tackle specific analysis issues. These workshops will be adapted from source materials already available from our two-week workshop, adapted to provide more scientific background.

Evaluation: BEACON NSF STC and SROP already conduct evaluations of students in their summer programs, and we will participate in these to assess the effectiveness of our training program. Moreover, the PI is engaged in developing novel evaluation materials for both the Analyzing Next Generation Sequencing Data grant and his BEACON-related teaching; these will be adapted to undergraduates to investigate the degree and extent of learning of bioinformatic- and computation-specific content during the summer courses.

References

[Angiuoli et al., 2011] Angiuoli, S., White, J., Matalka, M., White, O. and Fricke, W. (2011). Resources and costs for microbial sequence analysis evaluated using virtual machines and cloud computing. PLoS One *6*, e26624.

[Bonomi et al., 2006] Bonomi, F., Mitzenmacher, M., Panigrahy, R., Singh, S. and Varghese, G. (2006). Bloom Filters via d-left Hashing and Dynamic Bit Reassignment. In Proceedings of the Allerton Conference on Communication, Control and Computing.

[Brown et al., 2012] Brown, C., Howe, A., Zhang, Q., Pyrkosz, A. and Brom, T. (2012). A reference-free algorithm for computational normalization of shotgun sequencing data. In review at PLoS One, July 2012; Preprint at http://arxiv.org/abs/1203.4802.

[Brown and Tiedje, 2011] Brown, C. and Tiedje, J. (2011). Metagenomics: the paths forward, vol. Handbook of Molecular Microbial Ecology II: Metagenomics in Different Habitats,. Wiley.

[Compeau et al., 2011] Compeau, P., Pevzner, P. and Tesler, G. (2011). How to apply de Bruijn graphs to genome assembly. Nat Biotechnol *29*, 987–91.

[Conway and Bromage, 2011] Conway, T. and Bromage, A. (2011). Succinct data structures for assembling large genomes. Bioinformatics *27*, 479–86.

[Cormode and Muthukrishnan, 2005] Cormode, G. and Muthukrishnan, S. (2005). An improved data stream summary: the count-min sketch, and its applications. Journal of Algorithms *55*.

[Eid et al., 2009] Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., Peluso, P., Rank, D., Baybayan, P., Bettman, B., Bibillo, A., Bjornson, K., Chaudhuri, B., Christians, F., Cicero, R., Clark, S., Dalal, R., Dewinter, A., Dixon, J., Foquet, M., Gaertner, A., Hardenbol, P., Heiner, C., Hester, K., Holden, D., Kearns, G., Kong, X., Kuse, R., Lacroix, Y., Lin, S., Lundquist, P., Ma, C., Marks, P., Maxham, M., Murphy, D., Park, I., Pham, T., Phillips, M., Roy, J., Sebra, R., Shen, G., Sorenson, J., Tomaney, A., Travers, K., Trulson, M., Vieceli, J., Wegener, J., Wu, D., Yang, A., Zaccarin, D., Zhao, P., Zhong, F., Korlach, J. and Turner, S. (2009). Real-time DNA sequencing from single polymerase molecules. Science *323*, 133–8.

[Gans et al., 2005] Gans, J., Wolinsky, M. and Dunbar, J. (2005). Computational improvements reveal great bacterial diversity and high metal toxicity in soil. Science *309*, 1387–90.

[Gibson et al., 2008] Gibson, D., Benders, G., Axelrod, K., Zaveri, J., Algire, M., Moodie, M., Montague, M., Venter, J., Smith, H. and 3rd Hutchison CA (2008). One-step assembly in yeast of 25 overlapping DNA fragments to form a complete synthetic Mycoplasma genitalium genome. Proc Natl Acad Sci U S A *105*, 20404–9.

[Gilbert et al., 2010a] Gilbert, J., Meyer, F., Antonopoulos, D., Balaji, P., Brown, C., Brown, C., Desai, N., Eisen, J., Evers, D., Field, D., Feng, W., Huson, D., Jansson, J., Knight, R., Knight, J., Kolker, E., Konstantindis, K., Kostka, J., Kyrpides, N., Mackelprang, R., McHardy, A., Quince, C., Raes, J., Sczyrba, A., Shade, A. and Stevens, R. (2010a). Meeting report: the terabase metagenomics workshop and the vision of an Earth microbiome project. Stand Genomic Sci *3*, 243–8.

[Gilbert et al., 2010b] Gilbert, J., Meyer, F., Jansson, J., Gordon, J., Pace, N., Tiedje, J., Ley, R., Fierer, N., Field, D., Kyrpides, N., Glockner, F., Klenk, H., Wommack, K., Glass, E., Docherty, K., Gallery, R., Stevens, R. and Knight, R. (2010b). The Earth Microbiome Project: Meeting report of the '1 EMP meeting on sample selection and acquisition' at Argonne National Laboratory October 6 2010. Stand Genomic Sci *3*, 249–53.

[Glass et al., 2010] Glass, E., Wilkening, J., Wilke, A., Antonopoulos, D. and Meyer, F. (2010). Using the metagenomics RAST server (MG-RAST) for analyzing shotgun metagenomes. Cold Spring Harb Protoc *2010*, pdb.prot5368.

[Gnerre et al., 2011] Gnerre, S., Maccallum, I., Przybylski, D., Ribeiro, F., Burton, J., Walker, B., Sharpe, T., Hall, G., Shea, T., Sykes, S., Berlin, A., Aird, D., Costello, M., Daza, R., Williams, L., Nicol, R., Gnirke, A., Nusbaum, C., Lander, E. and Jaffe, D. (2011). High-quality draft assemblies of mammalian genomes from massively parallel sequence data. Proc Natl Acad Sci U S A *108*, 1513–8.

[Grabherr et al., 2011] Grabherr, M., Haas, B., Yassour, M., Levin, J., Thompson, D., Amit, I., Adiconis, X., Fan, L., Raychowdhury, R., Zeng, Q., Chen, Z., Mauceli, E., Hacohen, N., Gnirke, A., Rhind, N., di Palma, F., Birren, B., Nusbaum, C., Lindblad-Toh, K., Friedman, N. and Regev, A. (2011). Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nat Biotechnol *29*, 644–52.

[Henry et al., 2011] Henry, C., Overbeek, R., Xia, F., Best, A., Glass, E., Gilbert, J., Larsen, P., Edwards, R., Disz, T., Meyer, F., Vonstein, V., Dejongh, M., Bartels, D., Desai, N., D'Souza, M., Devoid, S., Keegan, K., Olson, R., Wilke, A., Wilkening, J. and Stevens, R. (2011). Connecting genotype to phenotype in the era of high-throughput sequencing. Biochim Biophys Acta *1810*, 967–77.

[Hess et al., 2011] Hess, M., Sczyrba, A., Egan, R., Kim, T., Chokhawala, H., Schroth, G., Luo, S., Clark, D., Chen, F., Zhang, T., Mackie, R., Pennacchio, L., Tringe, S., Visel, A., Woyke, T., Wang, Z. and Rubin, E. (2011). Metagenomic discovery of biomass-degrading genes and genomes from cow rumen. Science *331*, 463–7.

[HMP, 2012] HMP (2012). A framework for human microbiome research. Nature *486*, 215–21.

[Iverson et al., 2012] Iverson, V., Morris, R., Frazar, C., Berthiaume, C., Morales, R. and Armbrust, E. (2012). Untangling genomes from metagenomes: revealing an uncultured class of marine Euryarchaeota. Science *335*, 587–90.

[Keegan et al., 2012] Keegan, K., Trimble, W., Wilkening, J., Wilke, A., Harrison, T., D'Souza, M. and Meyer, F. (2012). A Platform-Independent Method for Detecting Errors in Metagenomic Sequencing Data: DRISEE. PLoS Comput Biol *8*, e1002541.

[Kelley et al., 2010] Kelley, D., Schatz, M. and Salzberg, S. (2010). Quake: quality-aware detection and correction of sequencing errors. Genome Biol *11*, R116.

[Koren et al., 2012] Koren, S., Schatz, M., Walenz, B., Martin, J., Howard, J., Ganapathy, G., Wang, Z., Rasko, D., McCombie, W., Jarvis, E. and Phillippy, A. (2012). Hybrid error correction and de novo assembly of single-molecule sequencing reads. Nat Biotechnol .

[Llewellyn and Eisenberg, 2008] Llewellyn, R. and Eisenberg, D. (2008). Annotating proteins with generalized functional linkages. Proc Natl Acad Sci U S A *105*, 17700–5.

[Mackelprang et al., 2011] Mackelprang, R., Waldrop, M., DeAngelis, K., David, M., Chavarria, K., Blazewicz, S., Rubin, E. and Jansson, J. (2011). Metagenomic analysis of a permafrost microbial community reveals a rapid response to thaw. Nature *480*, 368–71.

[Melsted and Pritchard, 2011] Melsted, P. and Pritchard, J. (2011). Efficient counting of k-mers in DNA sequences using a bloom filter. BMC bioinformatics *12*, 333.

[Miller et al., 2010] Miller, J., Koren, S. and Sutton, G. (2010). Assembly algorithms for next-generation sequencing data. Genomics *95*, 315–27.

[Namiki et al., 2012] Namiki, T., Hachiya, T., Tanaka, H. and Sakakibara, Y. (2012). MetaVelvet: an extension of Velvet assembler to de novo metagenome assembly from short sequence reads. Nucleic Acids Research *40*.

[NRC, 2007] NRC (2007). The New Science of Metagenomics: Revealing the Secrets of Our Microbial Planet. National Academies Press, Washington, D.C.

[Pell et al., 2012] Pell, J., Hintze, A., Canino-Koning, R., Howe, A., Tiedje, J. and Brown, C. (2012). Scaling metagenome sequence assembly with probabilistic de bruijn graphs. Accepted at PNAS, July 2012; Preprint at http://arxiv.org/abs/1112.4193.

[Peng et al., 2011] Peng, Y., Leung, H., Yiu, S. and Chin, F. (2011). Meta-IDBA: a de Novo assembler for metagenomic data. Bioinformatics *27*, i94–101.

[Pennisi, 2011] Pennisi, E. (2011). Human genome 10th anniversary. Will computers crash genomics? Science *331*, 666–8.

[Pérez and Granger, 2007] Pérez, F. and Granger, B. E. (2007). IPython: a System for Interactive Scientific Computing. Comput. Sci. Eng. *9*, 21–29.

[Pevzner et al., 2001] Pevzner, P., Tang, H. and Waterman, M. (2001). An Eulerian path approach to DNA fragment assembly. Proc Natl Acad Sci U S A *98*, 9748–53.

[Pignatelli and Moya, 2011] Pignatelli, M. and Moya, A. (2011). Evaluating the fidelity of de novo short read metagenomic assembly using simulated data. PLoS One *6*, e19984.

[Qin et al., 2010] Qin, J., Li, R., Raes, J., Arumugam, M., Burgdorf, K., Manichanh, C., Nielsen, T., Pons, N., Levenez, F., Yamada, T., Mende, D., Li, J., Xu, J., Li, S., Li, D., Cao, J., Wang, B., Liang, H., Zheng, H., Xie, Y., Tap, J., Lepage, P., Bertalan, M., Batto, J., Hansen, T., Paslier, D. L., Linneberg, A., Nielsen, H., Pelletier, E., Renault, P., Sicheritz-Ponten, T., Turner, K., Zhu, H., Yu, C., Li, S., Jian, M., Zhou, Y., Li, Y., Zhang, X., Li, S., Qin, N., Yang, H., Wang, J., Brunak, S., Dore, J., Guarner, F., Kristiansen, K., Pedersen, O., Parkhill, J., Weissenbach, J., Bork, P., Ehrlich, S. and Wang, J. (2010). A human gut microbial gene catalogue established by metagenomic sequencing. Nature *464*, 59–65.

[Schatz et al., 2010] Schatz, M., Langmead, B. and Salzberg, S. (2010). Cloud computing and the DNA data race. Nat Biotechnol *28*, 691–3.

[Simpson and Durbin, 2012] Simpson, J. and Durbin, R. (2012). Efficient de novo assembly of large genomes using compressed data structures. Genome Res *22*, 549–56.

[Sogin et al., 2006] Sogin, M., Morrison, H., Huber, J., Welch, D. M., Huse, S., Neal, P., Arrieta, J. and Herndl, G. (2006). Microbial diversity in the deep sea and the underexplored 'rare biosphere'. Proc Natl Acad Sci U S A *103*, 12115–20.

[Tringe and Rubin, 2005] Tringe, S. and Rubin, E. (2005). Metagenomics: DNA sequencing of environmental samples. Nat Rev Genet *6*, 805–14.

[Tyson et al., 2004] Tyson, G., Chapman, J., Hugenholtz, P., Allen, E., Ram, R., Richardson, P., Solovyev, V., Rubin, E., Rokhsar, D. and Banfield, J. (2004). Community structure and metabolism through reconstruction of microbial genomes from the environment. Nature *428*, 37–43.

[Venter et al., 2004] Venter, J., Remington, K., Heidelberg, J., Halpern, A., Rusch, D., Eisen, J., Wu, D., Paulsen, I., Nelson, K., Nelson, W., Fouts, D., Levy, S., Knap, A., Lomas, M., Nealson, K., White, O., Peterson, J., Hoffman, J., Parsons, R., Baden-Tillson, H., Pfannkoch, C., Rogers, Y. and Smith, H. (2004). Environmental genome shotgun sequencing of the Sargasso Sea. Science *304*, 66–74.

[von Mering et al., 2007] von Mering, C., Hugenholtz, P., Raes, J., Tringe, S., Doerks, T., Jensen, L., Ward, N. and Bork, P. (2007). Quantitative phylogenetic assessment of microbial communities in diverse environments. Science *315*, 1126–30.

[Woyke et al., 2010] Woyke, T., Tighe, D., Mavromatis, K., Clum, A., Copeland, A., Schackwitz, W., Lapidus, A., Wu, D., McCutcheon, J., McDonald, B., Moran, N., Bristow, J. and Cheng, J. (2010). One bacterial cell, one complete genome. PLoS One *5*, e10314.

[Ye et al., 2012] Ye, C., Ma, Z., Cannon, C., Pop, M. and Yu, D. (2012). Exploiting sparseness in de novo genome assembly. BMC Bioinformatics *13 Suppl 6*, S1.

[Zerbino et al., 2009] Zerbino, D., McEwen, G., Margulies, E. and Birney, E. (2009). Pebble and rock band: heuristic resolution of repeats and scaffolding in the velvet short-read de novo assembler. PLoS One *4*, e8407.