Xander: Gene Targeted Metagenomics

Jordan A Fish*[1,2], Yanni Sun[2] , James M Tiedje[1,3], C. Titus Brown[2,4]James R Cole[1]

[1]Center for Microbial Ecology, Michigan State University [2]Department of Computer Science and Engineer, Michigan State University [3]Department of Plant, Soil and Microbial Sciences, Michigan State University [4]Microbiology and Molecular Genetics, Michigan State University

Email: Jordan Fish*- fishjord@msu.edu; Yanni Sun - yannisun@msu.edu; James M Tiedje - tiedjej@msu.edu; C. Titus Brown - ctb@msu.edu; James R Cole - colej@msu.edu;

*Corresponding author

Abstract

Background: Metagenomics can provide important insight in to microbial communities. It can be used to analyze entire genomes and takes full advantage of increasing sequencing capacity. However analyzing large metagenomic datasets has proven to be very computationally challenging with even modest metagenomic datasets requiring hundreads of gigabytes to terabytes of memory to assemble with traditional assembly methods. As dataset sizes continue to increase as sequencing capacity increases new methods will be required for tackling the metagenomic assembly problem. In this paper, we present a method for assembling protein coding sequences for one or more genes of interest from a metagenomic dataset. This method uses a compressible graph format and only assembles targeted data to drastically reduce the amount of memory and processing time required.

Results: Using Xander we were able to assemble contigs for two targeted gene families (rplB and nirK) from a defined community sample and a soil rihzosphere metagenomic sample. From the defined community the assembled contigs matched the expected protein coding sequences. Using the contigs assembled from the metagenomic sample existing nirK primers were evaluated for coverage on uncultured organisms.

Conclusions: Gene targeted assembly enables the use gene targeted sequence analysis techniques on metagenomic datasets without requiring whole organism genomes be assembled. Gene-targeted assembly can also aid in primer development by providing an expanded set of organisms, including uncultured ones, on which to test primer sensistivity and specificity.

Background

Metagenomics has the potential to help answer many questions but has faced scalability challenges stemming from the amount of raw sequencing data generated by metagenomic experiments [1, 2]. Metagenomic assembly has been an area of growing interest in the past decade, with early datasets assembled using single genome assembly methods that had difficulty with metagenomic samples [3, 4]. The tendency for single genome assemblers to only assemble a few dominant organisms has been an impetus to develop metagnomic specific assembly methods [5].

We propose a gene targeted approach for assembling metagenomic datasets called Xander. Xander is a De Bruijn Graph [6] assembler [7] that uses external information to perform a guided, instead of exhaustive, traversal of the assembly graph. Xander uses profile Hidden Markov Models (HMM) [8] to guide traversal of the assembly graph. Using an HMM, the paths most likely to code for the target gene can be extended first thus limiting the portion of the assembly graph that must be explored. In addition to limiting the graph traversal HMMs provide a measure of how likely the resulting assembled contig comes from the supplied model. Using a gene-targeted assembly approach allows for the use of existing functional-based analysis methods with metagenomic data. Xander also enables researchers to examine genes involved in biologically interesting pathways without using amplicon based sequencing approaches.

Gene targeted assembly is less resource intensive and faster than traditional whole genome metagenomic assembly. In addition to the De Bruijn Graph, only small paths relative to the graph's size must be kept in memory. Further reduction in the memory usage are achieved by using a probabilistic data structure for holding the De Bruijn Graph in memory, a Bloom filter [9, 10]. By targeting relatively small segments of the assembly graph by using an HMM to guide assembly, the amount of the graph that must be explored during assembly is constrained, providing a speed up over whole genome approaches.

Targeted assembly of metagenomic datasets has drawn research interest and generated several approaches including EMIRGE [11] and Mira [12]. EMIRGE is an expectation maximization algorithm for assembling target sequences by iterative read mapping. The Mira assembler can perform a type of targeted assembly by using mirabait to extract out all reads that overlap with a reference set then assembling that subset of

reads. In addition to targeted-gene assembly De Bruijn Graphs have also been used as a database for a blast-like search algorithm, BlastGraph [13]. Xander differs from these targeted-assembly methods, which focus on identifying reads for targeted assembly, by using a De Bruijn Graph representation of the reads. Xander is similar to the BlastGraph approach but instead of using reads to query a De Bruijn Graph representation of the reference database, Xander queries a De Bruijn Graph representation of the reads using HMMs.

Results and Discussion

HMP Mock Community

Xander's performance was evaluated using the Human Metabiome Project's (HMP) whole genome shotgun (WGS) mock community datasets with Ribosomal Protein L2 (rplB) selected as the target gene. *rplB* was selected because it is a well conserved single copy gene. Each organism has a single copy of the *rplB* and several copies in the HMP mock community overlap by one or more 21-mer. The HMP mock community consists of 22 human gut associated microorganisms with sequenced genomes listed in Table . The HMP mock community WGS datasets consisted of a total of one gigabase of 75 basepair Illumina sequences available from NCBI's Short Read Archive under accession numbers SRR172902, SRR172903 which were combined for all analyses. The annotated whole genome records for each organism were downloaded from GenBank and the Coding Sequence (CDS) annotation was extracted.

The combined dataset was quality filtered as described below, a summary of the trimming results is contained in Table . A Bloom filter was built from the combined dataset in 8 minutes 41 seconds with an estimated 0.001% false positive rate. The Bloom filter was built with a k-mer size of 21 using four hash functions and was one gigabyte in size.

To evaluate the coverage of the HMP mock community dataset the combined read set was mapped to the whole genome sequences for all the organisms using Bowtie2 [14] summarized in Fig . Two organisms were not targeted for assembly: Candida albicans is a eukaryote and hence does not have a copy of bacterial *rplB* and Listeria monocytogenes since only a hand full of reads mapped to it's *rplB* region.

The starting vertices for Xander's search were selected by first aligning the reference genomes' translated *rplB* sequences to the *rplB* model used for searching. The protein alignment was then used to align the nucleotide sequences for each reference. For each *rplB* nucleotide reference sequence, the first consecutive 21 nucleotides aligning to the rplB model (e.g. without gaps or insertions), along with the model position the first resiude occupied were used as the search start points. A total of 20 starting vertices were selected,

one from each of the 20 organisms selected.

Xander was then run on the selected starts and a summary of the results are show in Table . Using the HMP mock community dataset with the start points selected Xander was able to assemble 20 *rplB* gene sequences with an average protein-protein identity to the reference gene of 90%. About half of the resulting sequences were partial assemblies, which due to cuts in the assembly graph caused by zero coverage for parts of the reference genomes which can be seen in the average coverage data in Table and in detail in Supplemental 1. Since several of the organisms overlapped by at least one 21-mer the paths in the assembly graph crossed which combined with sequencing errors lead to the differences seen between the expected protein sequence and assembled contig.

Primer Validation with Rizosphere Metagenome

Xander was used with a soil metagenomic dataset to assembly *nirK* sequences. The metagenomic sample was taken from a GLBRC test plot at Kellogg Biological Station and sequenced with the Illumina GAII sequencer. The dataset contained approximately 530 million 100 basepair reads. The De Bruijn Graph was constructed in nine hours with a k value of 30 using a Bloom filter size of 32GB, using four hash functions, and with an estimated false positive rate of 0.5%.

These assembled sequences were then analyzed using the ProbeMatch tool (http://github.com/rdpstaff/ProbeMatch) to test the sensitivity and specificity of an existing set of nirK primers (CITE). The sequence data was obtained after quality trimming was performed.

Cuts in the Assembly Graph

In addition to producing partial assemblies, cuts in the assembly graph can stop the search from terminating in a reasonable amount of time. If there is no path to the end of the model the A* search devolves in to an exhaustive traversal of the graph, something we specifically want to avoid. To handle these cases, two heuristic pruning methods were developed based on the log odds ratio comparing the probability the current path was generated by the HMM or a null model described in the Methods below. The effects of the different pruning methods can be seen in Fig . By making the pruning heuristic more strict Xander considers far fewer nodes enabling faster searching and discarding non-target path segments.

Conclusions

Xander is more sensitive than whole genome assembly methods and more specific than individual read-based approaches for functional analysis. Using an assembly based approach provides more context from which to make a classification decision as to whether a stretch of sequence belongs to the target gene family or not. Using HMM probabilities to guide local assembly helps to ensure the most relevant paths are explored to assemble sequences most likely to code for the target gene.

Methods

Graph Structure

A novel graph structure was created that combined a De Bruijn Graph (DG) and HMM together in a single combined assembly graph (CG) for assembling genes of interest. A vertex in CG is created for every pair of vertices u, v in DG and HMM:

$$\forall(u,v)\, u \in \text{DG},\ v \in \text{HMM}$$

each vertex in CG combines the information in u and v. The total number of vertices in CG will be

$$|V(DG)| * |V(HMM)|$$

where V(G) is the vertex set of the graph G. Vertices in CG are generated as needed to reduce the memory requirements.

The edge set $E(CG)$ was defined as follows: suppose w_i, and $w_j \in V(CG)$ and were made by combining vertices v_i with u_i and v_j with u_j respectively with v vertices from the De Bruijn Graph and u vertices from the HMM.

$$\overrightarrow{w_i w_j} \in E(\text{CG}) \leftrightarrow \overrightarrow{v_i v_j} \in E(\text{DG}) \text{ and } \overrightarrow{u_i u_j} \in E(\text{HMM})$$

. That is, an edge exists in CG if and only if an edge connects the vertices they were created by combining. The weight of an edge $\overrightarrow{uv}$ in CG are the defined as sum of the transition and emission probabilities taken from the HMM.

$$weight(\overrightarrow{uv}) = P_{transition}(u \to v) + P_{emission}(v)$$

The emission symbol is the unique character in the K-mer contained in v.

The De Bruijn Graph is constructed in nucleotide space regardless of whether the HMM is modeling protein or nucleotide sequences. When searching with a protein HMM the De Bruijn Graph is traversed in protein space by walking three nodes in any one direction at a time. The emission symbol then becomes

the three unique characters at the end of the K-mer translated to protein. The codon reading frame is fixed based on the vertex chosen to begin graph traversal.

Seed Identification

Xander includes two ways to identify seed kmers from which to start searching. Both methods use a representative reference set of aligned sequences from the target gene family. The k-mer and the model position from the aligned reference, and implicit match HMM state are combined to form a search starting vertex in CG.

The first seed identification approach is an exact seed matching approach. The reference sequences were broken up in to K-mers and stored in a hash table. Each read was then decomposed into K-mers that were then looked up in the hash of the references K-mers. For use with a protein HMM a seed length of $\lfloor K/3 \rfloor$ was used and input reads were translated in to all six reading frames. When assembling multiple target gene families the reference sets can be combined together into a single hash so that potential search starts can be identified in a single pass over the reads.

The second seed identification approach combines the seed identification and searching in to one step. The aligned reference file is provided to the BasicSearch program which then decomposes all the reference reads in to k-mers. When protein gene family is being targeted the references are broken up in to $k/3$ length words and back translated in to all possible nucleotide k-mers that translate to the reference protein word. The Bloom filter is then queried for each reference k-mer, any k-mers identified are then used as a starting point for a search.

Assembly Approach

Assemblies in Xander are done using the A* search algorithm [15] for finding paths through the CG. The A* implementation in Xander was modified to find the highest scoring path instead of the lowest cost path. The set of goal vertices is defined as any vertex in the last model position that is in the match or delete state. The scoring function for a path P is defined as:

$$S(P) = \sum_{i=0}^{|P|} w(P_i P_{i+1})$$

where w(...) is the weight of the edge between two vertices in P. By using the transition and emission probabilities from the HMM and selecting the highest scoring path through the graph the search is analagous to the Viterbi Decoding [16] algorithm.

The heuristic cost function for a vertex v is defined as:

$$h(v) = P_{v_{state} \to match} + \sum_{i=v_{modelposition}+1}^{M} P_{match \to match}(i, i+1)$$

the sum of the most likely state transitions from a v's state to the end of the model. Where P is the probability of the given transition and M is the length of the HMM.

To ensure The log-odds edge weights used by the heuristic score and scoring function were monotonic the following transformation is applied to every edge in CG:

$$w(\overrightarrow{uv}) = w(\overrightarrow{uv}) - max(P_{emission}(v_{HMMstate})) \quad u_{HMMstate} \neq i$$

Since this heuristic score will never overestimate the actual score it meets the admissibility criteria for A*, and additionally since the scoring function is monotonic a closed set is not required.

Since search starting vertices can be in any model position, not just the beginning of the model, a second HMM is built from the reverse of the seed alignment used to build the forward HMM. Using this reverse model Xander can traverse paths in both directions from a starting vertex. The contigs generated by each search direction are reported seperately; a tool is included with Xander to combine the two contigs fragments in to a single contig.

K Shortest Paths

A Kth shortest path algorithm [17] to find multiple high scoring paths from a single starting vertex. Yen's algorithm iteratively finds the shortest path, then 2nd shortest path to the kth shortest path. This is sped up by the observation that the ith shortest path in the sequence must branch from one of the $i-1$ shortest paths already identified. Yen's algorithm can be further improved by the observation that the ith shortest path must branch from it's parent j after the point j branched from it's parent [18]. The version of Yen's algorithm implemented in Xander was modified to discard paths that did not contain unseen kmers. In this way each of the K shortest paths found contain new information.

Prunning Unproductive Paths

Xander implements a path pruning heuristics to remove paths that are unlikely to yield contigs that match the model well. When a node is opened the probability of the path to that point is calculated and compared to the probability of the path being generated from a null model (cite hmmer2 null model) and the node is discarded if the log odds ratio is below a threshold value θ. In the event a search terminates

before reaching the end of the model, the intermediate node with the highest bits saved score is returned. This heuristic pruning is done in addition to the A* search. The log-odds-ratio threshold can be tuned using a command line switch to balance the trade-off between sensitivity and running time.

Quality Filtering

Read quality filtering was performed by trimming reads at quality score 2 as recommended by Illumina (CASAVA1.7 User Guide) using the SequenceTrimmer tool in the ReadSeq package (http://github.com/rdpstaff/ReadSeq.

HMM Construction

The HMMs were built using the seed sequences from the Functional Gene Repository (cite frontiers article). These seed sequences were used to build an HMM for each gene using a modified version of HMMER3 using the

`--enone`

option to disable sequence weighting. HMMER3's default settings were tuned for detecting remote paralogs [19] where Xander is targeting close homologs. The default priors sometimes caused extensive searching of nonproductive insert and delete paths. HMMER3's source code was modified to change the prior probabilities for the $delete \rightarrow match$ and $insert \rightarrow match$ transitions to 95% probability, $delete \rightarrow delete$ and $insert \rightarrow insert$ transitions to 5% probability. The modifications to HMMER3 are available as a patch file against version 3.0.

Implementation

Xander was implemented in the Java programming language and is distributed under the terms of the GPLv3 License available from https://github.com/rdpstaff/Xander-HMMgs. Xander uses a Bloom filter to store a compressed representation of the De Bruijn Graph. Xander consists of three primary tools; one for building a Bloom filter De Bruijn Graph, a tool for identifying starting positions, the core search tool. Support programs and scripts are also included with Xander for manipulating file formats, combining contig fragments and filtering Xander results.

Any of the tools can be replaced with a 3^{rd} party tool using difference heuristics as long as the resulting file matches the expected format. For example the starting vertex identification can be replaced with a 3^{rd} party tool so long as the resulting file contains the starting kmer and starting model position.

Authors contributions

Text for this section ...

Acknowledgements

Text for this section ...

References

1. Pop M, Phillippy A, Delcher AL, Salzberg SL: **Comparative genome assembly**. *Briefings in bioinformatics* 2004, **5**(3):237–248, [http://bib.oxfordjournals.org/content/5/3/237.short].
2. Treangen TJ, Salzberg SL: **Repetitive DNA and next-generation sequencing: computational challenges and solutions**. *Nature Reviews Genetics* 2011, [http://www.nature.com/doifinder/10.1038/nrg3117].
3. Venter JC, Remington K, Heidelberg JF, Halpern AL, Rusch D, Eisen JA, Wu D, Paulsen I, Nelson KE, Nelson W, Fouts DE, Levy S, Knap AH, Lomas MW, Nealson K, White O, Peterson J, Hoffman J, Parsons R, Baden-Tillson H, Pfannkoch C, Rogers YH, Smith HO: **Environmental Genome Shotgun Sequencing of the Sargasso Sea**. *Science* 2004, **304**(5667):66–74, [http://www.sciencemag.org/content/304/5667/66]. [PMID: 15001713].
4. Qin J, Li R, Raes J, Arumugam M, Burgdorf KS, Manichanh C, Nielsen T, Pons N, Levenez F, Yamada T, Mende DR, 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 JM, Hansen T, Le Paslier D, Linneberg A, Nielsen HB, 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, Doré J, Guarner F, Kristiansen K, Pedersen O, Parkhill J, Weissenbach J, Antolin M, Artiguenave F, Blottiere H, Borruel N, Bruls T, Casellas F, Chervaux C, Cultrone A, Delorme C, Denariaz G, Dervyn R, Forte M, Friss C, Guchte Mvd, Guedon E, Haimet F, Jamet A, Juste C, Kaci G, Kleerebezem M, Knol J, Kristensen M, Layec S, Roux KL, Leclerc M, Maguin E, Minardi RM, Oozeer R, Rescigno M, Sanchez N, Tims S, Torrejon T, Varela E, Vos Wd, Winogradsky Y, Zoetendal E, Antolin M, Artiguenave F, Blottiere H, Borruel N, Bruls T, Casellas F, Chervaux C, Cultrone A, Delorme C, Denariaz G, Dervyn R, Forte M, Friss C, Guchte Mvd, Guedon E, Haimet F, Jamet A, Juste C, Kaci G, Kleerebezem M, Knol J, Kristensen M, Layec S, Roux KL, Leclerc M, Maguin E, Minardi RM, Oozeer R, Rescigno M, Sanchez N, Tims S, Torrejon T, Varela E, Vos Wd, Winogradsky Y, Zoetendal E, Bork P, Ehrlich SD, Wang J: **A human gut microbial gene catalogue established by metagenomic sequencing**. *Nature* 2010, **464**(7285):59–65, [http://www.nature.com/nature/journal/v464/n7285/full/nature08821.html].
5. Namiki T, Hachiya T, Tanaka H, Sakakibara Y: **MetaVelvet: an extension of Velvet assembler to de novo metagenome assembly from short sequence reads**. *Nucleic Acids Research* 2012, **40**(20):e155–e155, [http://nar.oxfordjournals.org/content/40/20/e155]. [PMID: 22821567].
6. de Bruijn NG, Erdos P: **A combinatorial problem**. *Koninklijke Nederlandse Akademie v. Wetenschappen* 1946, **49**(49):758–764.
7. Compeau PEC, Pevzner PA, Tesler G: **How to apply de Bruijn graphs to genome assembly**. *Nature Biotechnology* 2011, **29**(11):987–991, [http://www.nature.com/nbt/journal/v29/n11/full/nbt.2023.html].
8. Eddy SR: **What is a hidden Markov model?** *Nature Biotechnology* 2004, **22**(10):1315–1316, [http://www.nature.com/nbt/journal/v22/n10/full/nbt1004-1315.html].
9. Bloom BH: **Space/time trade-offs in hash coding with allowable errors**. *Commun. ACM* 1970, **13**(7):422–426, [http://doi.acm.org/10.1145/362686.362692].
10. Pell J, Hintze A, Canino-Koning R, Howe A, Tiedje JM, Brown CT: **Scaling metagenome sequence assembly with probabilistic de Bruijn graphs**. *Proceedings of the National Academy of Sciences* 2012, [http://www.pnas.org/content/early/2012/07/25/1121464109]. [PMID: 22847406].

11. Miller CS, Baker BJ, Thomas BC, Singer SW, Banfield JF: **EMIRGE: reconstruction of full-length ribosomal genes from microbial community short read sequencing data**. *Genome Biology* 2011, **12**(5):R44, [http://genomebiology.com/2011/12/5/R44/abstract]. [PMID: 21595876].

12. Chevreux B, Pfisterer T, Drescher B, Driesel AJ, Müller WEG, Wetter T, Suhai S: **Using the miraEST Assembler for Reliable and Automated mRNA Transcript Assembly and SNP Detection in Sequenced ESTs**. *Genome Research* 2004, **14**(6):1147–1159, [http://genome.cshlp.org/content/14/6/1147]. [PMID: 15140833].

13. Holley G, Peterlongo P: **BlastGraph: intensive approximate pattern matching in string graphs and de-Bruijn graphs**. In *PSC 2012*, Prague, Czech Republic 2012[http://hal.inria.fr/hal-00711911].

14. Langmead B, Salzberg SL: **Fast gapped-read alignment with Bowtie 2**. *Nature Methods* 2012, **9**(4):357–359, [http://www.nature.com/nmeth/journal/v9/n4/full/nmeth.1923.html].

15. Hart P, Nilsson N, Raphael B: **A Formal Basis for the Heuristic Determination of Minimum Cost Paths**. *IEEE Transactions on Systems Science and Cybernetics* 1968, **4**(2):100–107.

16. Viterbi A: **Error bounds for convolutional codes and an asymptotically optimum decoding algorithm**. *IEEE Transactions on Information Theory* 1967, **13**(2):260–269.

17. Yen JY: **Finding the K Shortest Loopless Paths in a Network**. *Management Science* 1971, **17**(11):712–716, [http://www.jstor.org/stable/2629312]. [ArticleType: research-article / Issue Title: Theory Series / Full publication date: Jul., 1971 / Copyright © 1971 INFORMS].

18. Lawler EL: **A Procedure for Computing the K Best Solutions to Discrete Optimization Problems and Its Application to the Shortest Path Problem**. *Management Science* 1972, **18**(7):401–405, [http://www.jstor.org/stable/2629357]. [ArticleType: research-article / Issue Title: Theory Series / Full publication date: Mar., 1972 / Copyright © 1972 INFORMS].

19. Johnson S: **DIVISION OF BIOLOGY AND BIOMEDICAL SCIENCES**. *PhD thesis*, Washington University 2006, [http://selab.janelia.org/publications/Johnson06/Johnson06-phdthesis.pdf].

Figures

Figure 1 - HMP Mock Community Read Mapping

Percentage of reads mapped (unnormalized) to the reference organism. For organisms with more than one chromosome only the reads mapping to the chromosome containing the rplB gene were counted.

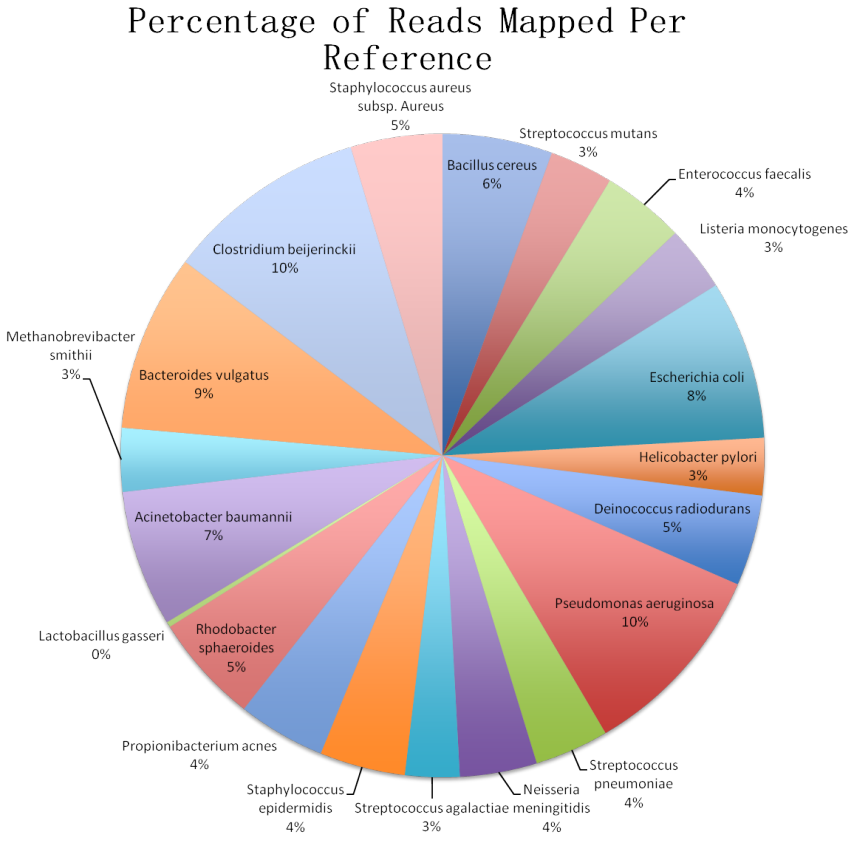

Percentage of Reads Mapped Per Reference
Staphylococcus aureus subsp. Aureus 5%
Streptococcus mutans 3%
Bacillus cereus 6%
Enterococcus faecalis 4%
Listeria monocytogenes 3%
Clostridium beijerinckii 10%
Escherichia coli 8%
Methanobrevibacter smithii 3%
Bacteroides vulgatus 9%
Helicobacter pylori 3%
Deinococcus radiodurans 5%
Acinetobacter baumannii 7%
Pseudomonas aeruginosa 10%
Lactobacillus gasseri 0%
Rhodobacter sphaeroides 5%
Propionibacterium acnes 4%
Streptococcus pneumoniae 4%
Staphylococcus epidermidis 4%
Streptococcus agalactiae 3%
Neisseria meningitidis 4%

Figure 2 - Comparision of pruning heursitics

Effect of cuts in the assembly graph on resulting contig plus effectiveness of different heuristic path pruning techniques.

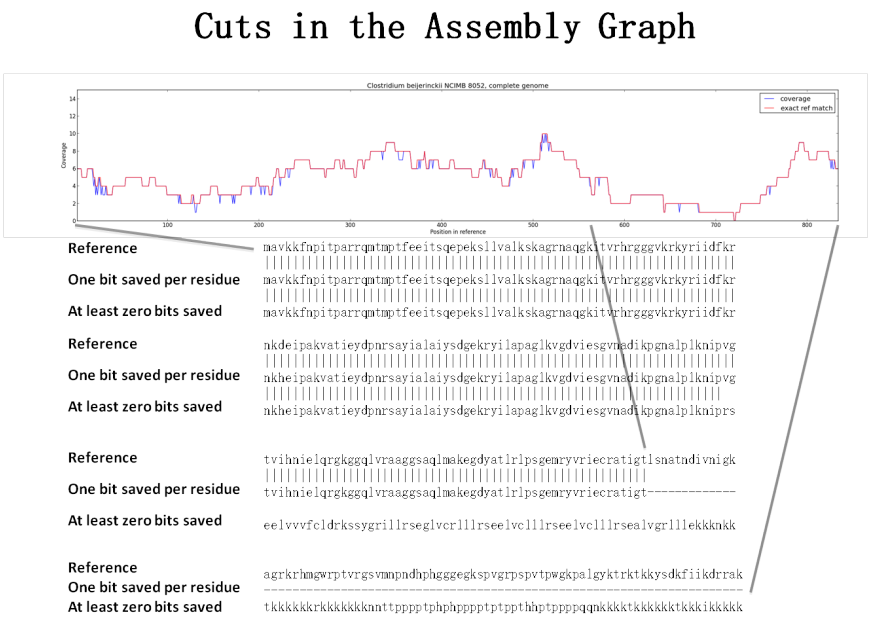


Figure 3 - Combined Graph Structure

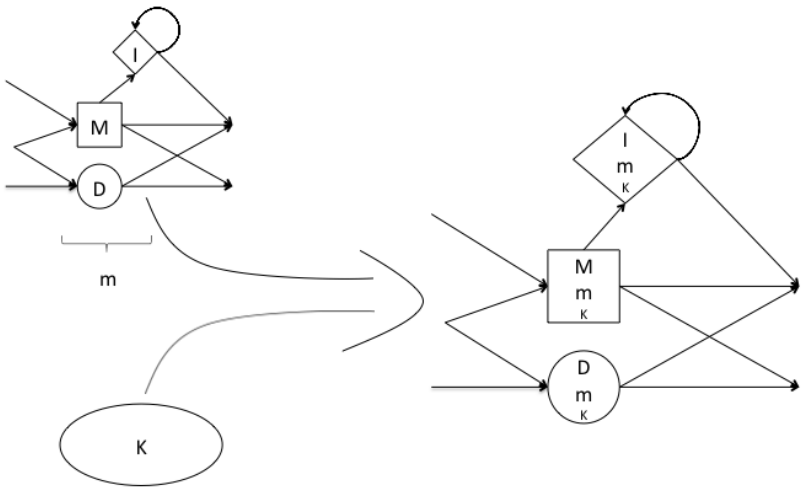
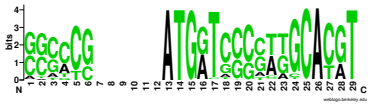
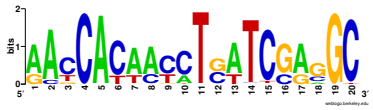


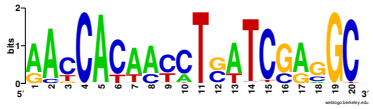
Figure 4 - Sequence conservation of nirK primer regions

Sequence conservation logos for the primer regions of the *nirK* sequences assembled using Xander from the MSR2 dataset. a) nirK1F b) nirK5R c) F1aCu d) R3Cu.

a.


b.


c.


d.


Tables

Table 1 - HMP Mock Community Composition

Organisms in the HMP mock community and accession number of the GenBank record from which annotations were harvested.

† indicates genome records with incomplete annotations, annotations from another assembly of a synonmous strain (the accession number in parathesis) were used instead. * indicates genomes that were removed from the final analysis.

Organism Name	Strain	Accession Number
Streptococcus mutans	NN2025 DNA	NC_003028 (AP010655) †
Listeria monocytogenes*	L99 serovar 4a	NC_003210 (FM211688)†
Acinetobacter baumannii	ATCC 17978	NC_009085.1
Acinetobacter baumannii	ATCC 17978 plasmid pAB1	NC_009083.1
Acinetobacter baumannii	ATCC 17978 plasmid pAB2	NC_009084.1
Actinomyces odontolyticus	ATCC 17982 Scfld020 & Scfld021	DS264586.1
Actinomyces odontolyticus	ATCC 17982 Scfld020	DS264585.1
Bacillus cereus	ATCC 10987	AE017194.1
Bacillus cereus	ATCC 10987 plasmid pBc10987	NC_005707.1
Bacteroides vulgatus	ATCC 8482	NC_009614.1
Candida albicans*	SC5314 Assembly 21	N/A
Clostridium beijerinckii	NCIMB 8052	NC_009617.1
Deinococcus radiodurans	R1 chromosome 1	NC_001263.1
Enterococcus faecalis	OG1RF chromosome	ABPI01000001.1
Escherichia coli	K12	NC_000913.2
Helicobacter pylori	26695	NC_000915.1
Lactobacillus gasseri	ATCC 33323	NC_008530.1
Methanobrevibacter smithii	ATCC 35061	NC_009515.1
Neisseria meningitidis	MC58	NC_003112.2
Propionibacterium acnes	KPA171202	NC_006085.1
Pseudomonas aeruginosa	PAO1	NC_002516.2
Rhodobacter sphaeroides	2.4.1 chromosome 1	NC_007493.1
Rhodobacter sphaeroides	2.4.1 chromosome 2	NC_007494.1
Rhodobacter sphaeroides	2.4.1 plasmid A, partial sequence	NC_009007.1
Rhodobacter sphaeroides	2.4.1 plasmid B	NC_007488.1
Rhodobacter sphaeroides	2.4.1 plasmid C	NC_007489.1
Rhodobacter sphaeroides	2.4.1 plasmid D	NC_007490.1
Rhodobacter sphaeroides	2.4.1 plasmid E	NC_009008.1
Staphylococcus aureus subsp. aureus	USA300_TCH1516	NC_010079.1
Staphylococcus aureus subsp. aureus	USA300_TCH1516 plasmid pUSA300HOUMR	NC_010063.1
Staphylococcus aureus subsp. aureus	USA300_TCH1516 plasmid pUSA01-HOU	NC_012417.1
Staphylococcus epidermidis	ATCC 12228	NC_004461.1
Staphylococcus epidermidis	ATCC 12228 plasmid pSE-12228-06	NC_005003.1
Staphylococcus epidermidis	ATCC 12228 plasmid pSE-12228-05	NC_005004.1
Staphylococcus epidermidis	ATCC 12228 plasmid pSE-12228-04	NC_005005.1
Staphylococcus epidermidis	ATCC 12228 plasmid pSE-12228-03	NC_005006.1
Staphylococcus epidermidis	ATCC 12228 plasmid pSE-12228-02	NC_005007.1
Staphylococcus epidermidis	ATCC 12228 plasmid pSE-12228-01	NC_005008.1
Streptococcus agalactiae	2603V/R	NC_004116.1
Streptococcus pneumoniae	TIGR4	NC_003028.3

Table 2 - HMP Trimming summary

Summary of the quality filtering results on the HMP Mock Community. Half the reads had at least one base removed from the 3' end. 17403 reads had no bases left after trimming and were removed.

	Total sequences	Total Bases	Average Length
Before Trimming	14494884	1037 MB	75
After Trimming	14477481	905 MB	65.5

Table 3 - Summary of Xander results for the HMP Mock community

Output of running Xander on the HMP mock community WGS dataset. The forward and reverse searches for each reference are combined in a single row in the table with the total time for each direction's search reported. The fragment nats (probability log base e) represent the score A* was optimizing on while the bits saved reflects the probability the sequence comes from the HMM. The average coverage of the *rplB* region computed by bowtie mapping for each reference is also reported.

Reference Organism	Reference Length	Starting State	Protein Length	left nats	left bits saved	right nats	rightbits saved	Search Time (s)	Average coverage (by bowtie2 mapping)
Bacillus cereus	274	250	31	-14.744	20	-8.216	68.66	0.123	1.07
Streptococcus mutans	263	256	283	-14.55	62.36	-66.953	815.82	1.933	26.84
Enterococcus faecalis	277	270	284	-11.49	123.61	-69.908	859.67	1.758	2.08
Escherichia coli	274	271	61	-3.263	25.38	-21.885	151.62	1.749	10.16
Helicobacter pylori	277	271	66	-1.742	27.57	-42.965	141.71	0.005	6.7
Deinococcus radiodurans	276	271	282	-3.17	725.5	-113.486	802.01	6.192	55.86
Pseudomonas aeruginosa	274	271	41	-1.983	27.22	-20.531	86.73	0.005	3.8
Streptococcus pneumoniae	278	270	284	-15.275	18.16	-72.054	856.43	7.52	
Neisseria meningitidis	278	271	23	-4.065	24.22	-8.74	50.93	0.001	6.43
Streptococcus agalactiae	263	270	284	-14.878	18.73	-71.657	857	1.983	1.78
Staphylococcus epidermidis	278	271	284	-1.652	27.7	-73.829	858.71	4.571	41.23
Propionibacterium acnes	279	271	282	-1.108	28.49	-106.232	812.45	4.509	7.19
Rhodobacter sphaeroides	280	271	281	-4.922	22.98	-174.453	714.31	3.515	33.66
Lactobacillus gasseri	277	193	2	-21.946	15.45	-26.169	51.91	0.001	0.09
Acinetobacter baumannii	243	238	284	-32.284	84.36	-61.043	771.09	0.732	12.44
Methanobrevibacter smithii	242	177	17	-22.914	12.43	-13.467	24.65	0.001	13.43
Bacteroides vulgatus	274	271	281	-2.035	27.15	-173.271	717.96	1.022	9.72
Clostridium beijerinckii	278	271	201	-2.702	26.19	-82.677	552.85	42.109	4.87
Staphylococcus aureus subsp. aureus	278	269	284	-5.063	34.49	-74.807	853.08	3.192	50.18

Table 4 - nirK Primer Coverage

Coverage of published *nirK* primers nirK1F, nirK5R, F1aCu and R3Cu on *nirK* sequences assembled from the MSR2 soil rizosphere metagenomic dataset and FunGene Repository (7,643 *nirK* sequences from release 7.2) allowing up to two mismatches.

		FunGene Repository		MSR2		
Primer	Sequence	Number of Hits	Coverage	Covering region	Matching Primer	Coverage
nirK1F	GGMATGGTKCCSTGGCA	1084	14%	113	1	1%
nirK5R	GCCTCGATCAGRTTGTGGTT	3656	48%	118	67	57%
F1aCu	ATCATGGTSCTGCCGCG	3247	42%	125	19	15%
R3Cu	GCCTCGATCAGRTTGTGGTT	3656	48%	118	67	58%

Additional Files

Additional file 1 — Sample additional file title

Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.