Uncovering the fine-grained structure of the human microbiome with synthetic long reads

Supplementary material

Table of Contents

Metagenomic sample preparation

Mock microbioal DNA, HM-277D Staggered v5.2H, was obtained from BEIresources. Gut microbiome DNA was isolated from the frozen feces of a healthy subject using PowerSoil DNA Isolation Kit (MO BIO Laboratories, Inc.). Both DNA samples were sequenced using Illumina Tru-seq synthetic long reads technique (three and seven libraries for mock and gut microbiome samples, respectively) and the standard shortgun technique, with each library sequenced on one full lane of HiSeq. All libraries were prepared according the manufacturer's standard protocol. Shotgun sequence reads for both the mock and the gut metagenomic samples were subsampled at random to produce subsampled libraries containing the same amount of base pairs as the in the Tru-seq synthetic long read libraries. The results were assembled on the Illumina Basespace platform, according to standard protocol.

We used Ovation Ultralow DR Multiplex Systems 1–8 (0330-32, NuGEN Technologies, Inc.) for whole genome library preparation. Briefly, 100 ng of intact gDNA was diluted into 120 µL of 1X low EDTA-TE buffer and transferred to Covaris snap cap microtube and Fragmented to 300 bp following Covaris recommended settings. Fragmented DNA was purified using Agencourt RNAClean XP bead, provided by Nugen Library preparation kit. The sheared DNA was then subjected to end repair and adaptor ligation. Adaptor ligated libraries were purified with Agencourt RNAClean XP bead and amplified using 18 PCR cycle of 94°C for 30 sec, 60°C –for 30 sec, and 72°C for 1 min. Agencourt RNAClean XP bead was used for amplified Library Purification and libraries Fragment distribution was validated on Bioanalyzer DNA Chip 1000.

Overview of the Nanoscope pipeline

In order to facilitate the analysis of synthetic long read data for in the context of metagenomics, we have developed a bioinformatics pipeline called Nanoscope (Figure 1). Nanoscope takes as input a set of long read libraries together with optional (but highly recommended) short read libraries. It then performs a four-stage analysis of this data that includes de-novo assembly, variant calling and haplotyping, taxonomic identification, and abundance estimation.

Nanoscope starts by invoking the Soapdenovo[1] and Celera[2] assemblers to independently assemble the short and long read libraries, before merging the results using Minimus2[3]. In the next step, it invokes a variant calling and phasing algorithm called Lens to analyze the assembled contigs for strain variation. Lens reveals hundreds to thousands of sites where individual bacteria of the same strain differ from each other and then phases these variants into bacterial haplotypes. A typical contig might harbor more than a dozen different strain haplotypes, each of which may contain thousands of sequence variants. Variants and haplotypes are determined using a simple model (see the section on Lens below) that, unlike previous approaches[4,5], does not make any assumptions on the length of sequencing reads or the ploidy of the organism; we have found that these factors may confuse existing bacterial variant callers and lead to suboptimal results.

Finally, Nanoscope invokes the FCP software package[6] to assign taxonomic labels to assembled contigs and to estimate bacterial abundances. The latter task is done by mapping short reads to assembled contigs and by aggregating the coverage over all contigs assigned to the same species. Computing abundances from short reads avoids certain biases inherent to synthetic long reads; mapping reads to contigs enables estimation of abundances for bacteria whose genomes are not present in standard databases. At each stage, Nanoscope uses the popular Quast tool[7] to assess its performance and to generate reports.

Nanoscope differs from existing metagenomic pipelines[8,9] because it includes additional programs for dealing with synthetic long reads (most notably, the Celera and Minimus2 assemblers). We modified the source code of some of these packages to handle longer genomic sequences (see below); all programs used by Nanoscope have also been tuned for longer read lengths. The source code of Nanoscope is publically available in an open-source repository.

De-novo assembly

Short and long read libraries are first assembled using the Soapdenovo2 r240 (k-mer size of 51) and Celera 8.1 assemblers, respectively. The Soapdenovo 2 parameters we use are:

```
    $(SOAP) sparse_pregraph -K 51 -z 250000000 -R -s
    config.txt -p $(PROCESSORS) -o short
    $(SOAP) contig -g short -p $(PROCESSORS)
    $(SOAP) map -s config.txt -g short -p $(PROCESSORS)
    $(SOAP) scaff -g short –F -p $(PROCESSORS)
```

The configuration script we use is:

```
    [LIB]
    reverse_seq=0
    asm_flags=3
    rank=1
```

We use the following feature flags for the long read .frg library:

```
$(CELDIR)/fastqToCA \
    -nonrandom \
    -reads $< \
    -libraryname 0 \
    -technology none \
    -feature forceBOGunitigger 0 \
    -feature doNotTrustHomopolymerRuns 0 \
    -feature discardReadsWithNs 0 \
    -feature doNotQVTrim 0 \
    -feature deletePerfectPrefixes 0 \
    -feature doNotOverlapTrim 0 \
    -feature isNotRandom 0
```

We use the following spec file for the Celera assembler (inspired by the spec file used for assembling the Sagrasso sea metagenome):

```
    overlapper = ovl
    unitigger = bogart
    merSize = 14
    ovlStoreMemory = 1192
    ovlHashBits=24
    ovlHashBlockLength = 20000000
    ovlRefBlockSize =  5000000
    ovlMinLen=1000
    cnsReuseUnitigs=1
    cnsReduceUnitigs=0

    frgCorrBatchSize = 200000
    ovlCorrBatchSize = 100000
    doFragmentCorrection=0
    utgGenomeSize = 40000000
```

The resulting contigs are merged and deduplicated using a custom version of the CD-HIT 4.6.1 package; overlaps between contigs are computed with a modified Mummer 3.23 and the assemblies are merged with Minimus2.

```
    cat short.fasta long.fasta>ctgs.fasta
    $(CDHIT)/cd-hit-est –i ctgs.fasta –o
ctgs.filtered.fasta c 0.99 -M 3000
    $(AMOSDIR)/toAmos -s ctgs.filtered.fasta -o out.afg
    $(AMOSDIR)/minimus2 out -D OVERLAP 500
```

The modifications to CD-HIT 4.6.1 are a bugfix when dealing with short reads and an increase of MAX_SEQ (in cdhit-common.h) to 5000000. Mummer 3.23 was recompiled with make CPPFLAGS="-O3 -DSIXTYFOURBITS" in order to handle longer sequences.

Variant calling and haplotyping

Long reads are realigned back to the contigs using BWA-mem 0.7.5a with default parameters; the Lens algorithm then detects strain-specific variants and places them into bacterial haplotypes (see below). Default parameters were used:

```
python filter_by_cigar.py -i input.bam -o
   filtered.bam;
samtools view -b -h -q 30 filtered.bam >
   filtered.q30.bam;
python make_variants.py   -b filtered.q30.bam   -o
  variants.pos   --coverage-threshold 3   --frequency-
  threshold 0.1   --qscore-threshold 15   --indels
python make_reads.py   -b filtered.q30.bam   -v
  variants.pos   -r variants.reads   --qscore-
  threshold 15
python detect_subspecies.py -b filtered.q30.bam   -r
  variants.reads  -k haplotypes.txt  --cov-cutoff 2
  --cov-cutoff-percentage 0.75   --similarity-cutoff
  1.0   --overlap-cutoff 2
```

Species identification

Next, the LCA algorithm from the FCP package is used for assigning taxonomic labels.

```
$(FCPDIR)/BLASTN.py blastn ctgs.fasta ctgs.blastn.results
$(FCPDIR)/LCA.py ctgs.blastn.results 1e-5 15 lca.results
$(FCPDIR)/TaxonomicSummary.py ctgs.blastn.results
lca.results lca.summary
```

The reference database consists of all finished bacterial and archaeal genomes in the NCBI RefSeq database (ftp://ftp.ncbi.nih.gov/genomes/Bacteria/all.gbk.tar.gz). Given a taxonomic rank, Nanoscope defines the taxa present in the sample as all ones to which at least at least 5 contigs and at least 40,000 bp of sequence were assigned. This is implemented in the script *select_present_taxa.py*.

Abundance estimation

Finally, short reads are aligned to assembled contigs with BWA-mem 0.7.5a with parameters '–T 30' and the average coverage for each contig is computed. The coverage of each contig was computed using *samtools idxstats*. The coverage of a taxon that was determined to be present in the sample is defined to be the average coverage of all contigs greater than 10,000 bp that have been assigned to it. The abundance of a taxon is its normalized coverage. Abundance estimation is implemented in *determine_abundances.py*.

```
$(PYTHON) $(BIN)/select_present_taxa.py \
       --tax lca.summary \
       --res 'SPECIES' \
       --bp-cutoff 40000  \
       --ctg-cutoff 5 \
       --present taxa.suspected
$(SAMTOOLS) idxstats short-read-to-ctgs.bam >
       contg.coverages
$(PYTHON) $(BIN)/determine_abundances.py \
       --taxa lca.summary \
       --ctgs lca.reults \
       --cov ctg.coverages \
       --res 'SPECIES' \
       --abundances taxa.abundances
```

The Lens haplotyper and variant caller

Lens is a new variant calling and phasing tool specialized for metagenomes and synthetic long reads. It is based on algorithms that, unlike previous approaches[4,5], do not make any assumptions on the length of sequencing reads or the ploidy of the organism; we have found that these factors may confuse existing bacterial variant callers and lead to suboptimal results (see below). At a high level, Lens does two things: starting from an alignment of long reads to assembled contigs (or to bacterial reference genomes), it first determines positions at which the reads and the reference differ; these positions are indicative of multiple closely related strains of the same bacterium. Then, Lens phases these variants into long haplotypes, each haplotype being defined in this context as a set of variants that co-occur within the same bacterial substrain.

The Lens haplotyper leverages the fact that each long read originates from a single organism, and therefore all variants within a read must belong to the same substrain. By connecting reads at their overlapping variants, Lens places the variants into multi-kilobase-long haplotypes in a process that is reminiscent of single-individual haplotyping (SIH) techniques[10]. In our setting, the number of true haplotypes is an unknown parameter that may be greater than two, making the phasing problem considerably more difficult. Although there exist well-known phasing algorithms for polyploid genomes (e.g. plants or cancer genomes), they all assume a fixed, known ploidy[11,12], with the notable exception of some recent methods developed while this paper was under review [13,14]; Lens on the other hand infers the ploidy directly from the data. More precisely, Lens assembles haplotypes using an approximate greedy procedure (see below); this choice is in part due to the fact that the SIH problem (of which ours is a generalization) is computationally intractable[15]. In brief, Lens sorts aligned reads from left to right and in turn uses each read to either extend an existing haplotype or to form a new one, depending on the read-haplotype overlap and on the cost of forming a new cluster (both are tunable parameters for the algorithm). Our high-level approach may in principle have applications outside metagenomics, such as in cancer genome phasing.

The Lens phasing algorithm

The Lens program starts with an alignment of long reads to contigs in BAM format. Reads with alignment scores lower than 30 are discarded. Lens first builds a list of variants that have high support from the reads: it selects all variants that occur in at least three reads and in at least 5% of all reads that map to that position. Then, for every read, Lens records which of the variants identified above it contains; variants having a q-score less than 15 are not recorded. Finally, Lens uses a greedy heuristic to assemble long reads into bacterial haplotypes. It iterates over all reads (sorted by the their starting positions), and at each step either adds a read to an existing haplotype (if the read and the haplotype overlap at two positions or more and agree completely on their overlap), or is used to initialize a new haplotype:

```
• Let $O = H = \emptyset$ be (respectively) the set of output haplotypes and the current (working) set of haplotypes.
• For every variant position $p$ (in increasing order):
    o Let $R_p$ be the set of reads that spans position $p$.
    o If $R_p \cap R_{p-1} = \emptyset$, then $O \leftarrow O \cup H$ and $H \leftarrow \emptyset$.
    o For $r \in R_p$:
        ▪ If $\exists h \in H$ such that read $r$ overlaps haplotype $h$ at $v$ variant sites without error (if the 'cautious' flag is set, require h to be unique):
            • Use $r$ to extend $h$.
        ▪ Else:
            • Let $h' = \{r\}$. Add the new haplotype $h'$ to $H$.
• Let $O \leftarrow O \cup H$. Return $O$.
```

At the end, all haplotypes in O that have less than q% of their sequence covered by reads at a coverage of c are discarded. All other haplotypes are reported. The parameters v, q, c are set via command-line flags; their default values are respectively 2 variants, 75% and 2X. The 'cautious' flag is not set by default'.

Running Lens

Lens consists of a series of four scripts that are run one after the other. Lens takes as input a .bam file with long reads aligned to a reference genome or to an assembled genomic contig. We used 'bwa mem –T 30' for generating this file (using BWA 0.7.5a).

BAM filtering. We ran Lens only on reads with high mapping scores (>=30). We also used reads that align across their entire length (that had fewer than 500 bp in clipped regions, as defined by the CIGAR string). We filtered for such reads using:

```
python filter_by_cigar.py -i input.bam -o
    filtered.bam;

samtools view -b -h -q 30 filtered.bam >
    filtered.q30.bam
```

Variant calling. Our first step was to call variants in the reference:

```
python make_variants.py   -b filtered.q30.bam   -o
variants.pos   --coverage-threshold 3   --frequency-
threshold 0.1   --qscore-threshold 15   --indels
```

This will finds all variants (including indels) that have at least 3 reads supporting them and have an allele frequency of at least 0.1 (these are the default parameters). Also, only reads that have a qscore of 15 or more at a given position are used for calculating support.

Variant calling in reads. Next, we compiled the reads that support the variants we have just identified. The result of this step is a file that lists for each read all the variants that the read covers.

```
python make_reads.py \   -b filtered.q30.bam \   -v
     variants.pos \   -r variants.reads \   --qscore-
     threshold 15
```

The format of variants.reads is: <ctg id> <read id> <pos1:allele1> <pos2:allele2> ...

Assembling reads into haplotypes. Finally, we assembled the reads at their overlapping variants into bacterial haplotypes.

```
python detect_subspecies.py \   -b filtered.q30.bam \
-r variants.reads \   -k haplotypes.txt \   --cov-
cutoff 2 \   --cov-cutoff-percentage 0.75 \   --
similarity-cutoff 1.0 \   --overlap-cutoff 2
```

This reports only haplotypes that have a coverage of at least two over at least 75% of their length. Two reads will be connected into a haplotype if they overlap at at least two variants and are identical at both of these positions.The format of haplotypes.txt is: <ctg id>   <start pos>   <end pos>   <coverage>   <variants>.

Quality control

Assessing the bias of Tru-seq synthetic long reads

We assessed the accuracy of the synthetic long read libraries on the mock metagenome by mapping them to known reference genomes using Mummer 3.23. The number of misassemblies was determined using the Quast 2.3 package. To compare the coverage biases of Tru-seq synthetic long reads to traditional Illumina short reads, we aligned both kinds of reads to the reference using BWA-mem 0.7.5a with default parameters. We used bedtools 2.17.0 with options ‘genomecov –ibam <bam> -bga > genome.cov’ to compute the coverage at each position; the fraction of a genome covered (Supplementary Figure 3) was defined as the percentage of base pairs with non-zero coverage. We used ‘samtools idxstats <bam>’ to compute the coverage at each contig; species abundances (Supplementary Figure 4) were defined as the normalized coverage of each contig.

Evaluating the assembly of the mock and human gut metagenomes

The assessment of accuracy and the prediction of genes on the gut metagenome was done using Quast 2.3 and Metagenemark 2.8 (packaged with Quast). We used as true reference for the mock metagenome the following Fasta sequences downloaded from NCBI (ftp://ftp.ncbi.nlm.nih.gov/genomes/Bacteria/):

```
1	>gi|10957398|ref|NC_000958.1| Deinococcus radiodurans R1 plasmid MP1, complete sequence
2	>gi|10957530|ref|NC_000959.1| Deinococcus radiodurans R1 plasmid CP1, complete sequence
3	>gi|15644634|ref|NC_000915.1| Helicobacter pylori 26695 chromosome, complete genome
4	>gi|15805042|ref|NC_001263.1| Deinococcus radiodurans R1 chromosome 1, complete sequence
5	>gi|15807672|ref|NC_001264.1| Deinococcus radiodurans R1 chromosome 2, complete sequence
6	>gi|16802048|ref|NC_003210.1| Listeria monocytogenes EGD-e, complete genome
7	>gi|22536185|ref|NC_004116.1| Streptococcus agalactiae 2603V/R chromosome, complete genome
8	>gi|27466918|ref|NC_004461.1| Staphylococcus epidermidis ATCC 12228 chromosome, complete genome
9	>gi|32470520|ref|NC_005003.1| Staphylococcus epidermidis ATCC 12228 plasmid pSE-12228-06, complete sequence
10	>gi|32470532|ref|NC_005004.1| Staphylococcus epidermidis ATCC 12228 plasmid pSE-12228-05, complete sequence
11	>gi|32470555|ref|NC_005005.1| Staphylococcus epidermidis ATCC 12228 plasmid pSE-12228-04, complete sequence
12	>gi|32470572|ref|NC_005006.1| Staphylococcus epidermidis ATCC 12228 plasmid pSE-12228-03, complete sequence
13	>gi|32470581|ref|NC_005007.1| Staphylococcus epidermidis ATCC 12228 plasmid pSE-12228-02, complete sequence
14	>gi|32470588|ref|NC_005008.1| Staphylococcus epidermidis ATCC 12228 plasmid pSE-12228-01, complete sequence
15	>gi|42740913|gb|AE017194.1| Bacillus cereus ATCC 10987, complete genome
16	>gi|44004339|ref|NC_005707.1| Bacillus cereus ATCC 10987 plasmid pBc10987, complete sequence
17	>gi|49175990|ref|NC_000913.2| Escherichia coli str. K-12 substr. MG1655, complete genome
18	>gi|50841496|ref|NC_006085.1| Propionibacterium acnes KPA171202 chromosome, complete genome
19	>gi|77358697|ref|NC_003112.2| Neisseria meningitidis MC58 chromosome, complete genome
20	>gi|77404592|ref|NC_007488.1| Rhodobacter sphaeroides 2.4.1 plasmid B, complete sequence
21	>gi|77404693|ref|NC_007489.1| Rhodobacter sphaeroides 2.4.1 plasmid C, complete sequence
22	>gi|77404776|ref|NC_007490.1| Rhodobacter sphaeroides 2.4.1 plasmid D, complete sequence
23	>gi|77461965|ref|NC_007493.1| Rhodobacter sphaeroides 2.4.1 chromosome 1, complete sequence
24	>gi|77464988|ref|NC_007494.1| Rhodobacter sphaeroides 2.4.1 chromosome 2, complete sequence
25	>gi|110645304|ref|NC_002516.2| Pseudomonas aeruginosa PAO1 chromosome, complete genome
```

26 >gi|116628683|ref|NC_008530.1| Lactobacillus gasseri ATCC 33323 chromosome, complete genome
27 >gi|125654605|ref|NC_009007.1| Rhodobacter sphaeroides 2.4.1 plasmid A, partial sequence
28 >gi|125654693|ref|NC_009008.1| Rhodobacter sphaeroides 2.4.1 plasmid E, partial sequence
29 >gi|126640097|ref|NC_009083.1| Acinetobacter baumannii ATCC 17978 plasmid pAB1, complete sequence
30 >gi|126640109|ref|NC_009084.1| Acinetobacter baumannii ATCC 17978 plasmid pAB2, complete sequence
31 >gi|126640115|ref|NC_009085.1| Acinetobacter baumannii ATCC 17978 chromosome, complete genome
32 >gi|148337902|gb|DS264586.1| Actinomyces odontolyticus ATCC 17982 Scfld021 genomic scaffold,
33 >gi|148337903|gb|DS264585.1| Actinomyces odontolyticus ATCC 17982 Scfld020 genomic scaffold
34 >gi|148642060|ref|NC_009515.1| Methanobrevibacter smithii ATCC 35061 chromosome, complete genome
35 >gi|150002608|ref|NC_009614.1| Bacteroides vulgatus ATCC 8482 chromosome, complete genome
36 >gi|150014892|ref|NC_009617.1| Clostridium beijerinckii NCIMB 8052 chromosome, complete genome
37 >gi|161508266|ref|NC_010079.1| Staphylococcus aureus subsp. aureus USA300_TCH1516 chromosome
38 >gi|161510924|ref|NC_010063.1| Staphylococcus aureus subsp. aureus USA300_TCH1516 plasmid pUSA300HOUMR
39 >gi|194172857|ref|NC_003028.3| Streptococcus pneumoniae TIGR4 chromosome, complete genome
40 >gi|225631039|ref|NC_012417.1| Staphylococcus aureus subsp. aureus USA300_TCH1516 plasmid pUSA01-HOU
42 >Ca21chr1_C_albicans_SC5314 (3188548 nucleotides)
43 >Ca21chr2_C_albicans_SC5314 (2232035 nucleotides)
44 >Ca21chr3_C_albicans_SC5314 (1799406 nucleotides)
45 >Ca21chr4_C_albicans_SC5314 (1603443 nucleotides)
46 >Ca21chr5_C_albicans_SC5314 (1190928 nucleotides)
47 >Ca21chr6_C_albicans_SC5314 (1033530 nucleotides)
48 >Ca21chr7_C_albicans_SC5314 (949616 nucleotides)
49 >Ca21chrR_C_albicans_SC5314 (2286389 nucleotides)
50 >gi|347750429|ref|NC_004350.2| Streptococcus mutans UA159 chromosome, complete genome
51 >gi|384511964|ref|NC_017316.1| Enterococcus faecalis OG1RF chromosome, complete genome

On the mock metagenome, we supplied Quast with gene and operon coordinates; gene coordinates were also obtained from NCBI (ftp://ftp.ncbi.nlm.nih.gov/genomes/Bacteria/); the following files were downloaded:

NC_000913.gff NC_000915.gff NC_000958.gff NC_000959.gff NC_001263.gff NC_001264.gff NC_002516.gff NC_003028.gff NC_003210.gff NC_003909.gff NC_004116.gff NC_004461.gff NC_005003.gff NC_005004.gff NC_005005.gff NC_005006.gff NC_005007.gff NC_005008.gff NC_005707.gff NC_006085.gff NC_007488.gff NC_007489.gff NC_007490.gff NC_007493.gff NC_007494.gff NC_008530.gff NC_009007.gff NC_009008.gff NC_009083.gff NC_009084.gff NC_009085.gff NC_009515.gff NC_009614.gff NC_009617.gff NC_010063.gff NC_010079.gff NC_012417.gff

Operon coordinates were obtained from the ProOpDB database (http://operons.ibt.unam.mx/OperonPredictor/) for the following 17 organisms: *A. baumanii, E.coli, M.smithii, R.shpaeroides, S.pneumoniae, B.cereus, H.pylori, N. meningitidis, S.agalactiae, C.beijerinckii, L. gasseri, P. acnes, S.aureus, D.radiodurans, L.monocytogenes, P.aeruginosa, S.epidermidis*

These genes and operons were fed to Quast using the –G and –O options.

We used the above genes and operons when evaluating the mock metagenome. The genes reported (predicted) for the gut metagenomes are obtained by Metagenemark, which is invoked by Quast.

Analyzing strain haplotypes

Variants were determined and assembled into strain haplotypes using Lens. We define the span of a haplotype as the interval between the first and last variant in that haplotype. Two haplotypes are said to intersect if their spans intersect. A haplotype graph G is defined over haplotypes and two haplotypes are connected in G if their spans intersect. Genomic regions harboring haplotypes were defined as the union of the haplotype spans within a connected component of G (with one region per connected component). We tested for intersection with gene intervals obtained via Metagenemark as described above. All these calculations were performed using bedtools 2.17.0. Genomic regions were computed using *bedtools merge* on the list of haplotype spans. Intersection with ORFs was computed similarly using *bedtools intersect*.

To evaluate whether the gut microbiome haplotypes correspond to bacterial strains, we determined whether they satisfy perfect phylogeny[16]. A phylogenetic tree suggests how bacterial strains (represented by haplotypes here) might have evolved from one another; a tree satisfies the perfect phylogeny property if all strains evolved from a common ancestor and during this process, each position mutated at most once. When perfect phylogeny is not met (such as when certain positions have mutated twice), it is possible to measure the extent to which it is violated by estimating the number of positions that can be excluded to make perfect phylogeny hold. To assess whether variants were in perfect phylogeny, we used the four-gametes characterization of perfect phylogeny; this condition says that a binary phylogenetic matrix admits a perfect phylogeny if and only if there are no two columns and four rows such that the alleles at these four rows over the two columns are (0,0), (0,1), (1,0), (1,1). We found 227 regions with four our more haplotypes. A small number of genomic regions could not be expressed as a binary matrix, and we discarded 10 such regions from our analysis. Of the remaining 217, 185 satisfied the four-gametes condition. To determine the number of positions to be corrected to ensure perfect phylogeny, we used a greedy heuristic, where the position with the most conflicts (i.e. for which the number of column pairs that does not satisfy the four gametes condition was the highest) was discarded until the region was in perfect phylogeny.

We downloaded essential genes from the OGEEdb database for all bacterial organisms that were also found in the mock gut metagenome, and combined them with E. coli essential genes downloaded from http://www.genome.wisc.edu/Gerdes2003/supplementary_table.html. We determined essential genes contained in assembled gut metagenomic contigs by mapping them with Mummer 3.26, choosing matches that spanned the entire length of the gene with 99% or more similarity. We determined the overlapping variants using *bedtools intersect*. In the mock metagenome, we determined genome coordinates directly from the .gff files described in the assembly subsection above. We called variants in the E. Coli reference genome (taken from the mock metagenome reference) using Lens as described earlier. We used SNPEff 4.0 with default parameters to predict the deleteriousness of each variant. We used the following list of essential genes: http://www.genome.wisc.edu/Gerdes2003/supplementary_table.html and we took their coordinates in E. Coli from the .gff file described above. Intersections between variants and gene intervals was computed using *bedtools intersect*.

Evaluating the detection quality of low-abundance bacteria

We used the LCA and Epsilon-NB algorithms from the FCP 1.0.5 package as representatives of homology and composition-based methods. The former involves aligning contigs to a reference database using BLASTN 2.2.25+ (parameters are chosen automatically by FCP 1.0.5) and requires two parameters, which we set to (1e-5, 15). We used a parameter of 1e10 with Epsilon-NB. True labels were found by aligning contigs to the mock metagenome reference with Mummer; contigs having a single match larger than 1 kbp were assigned a label. To compare our abundance estimation accuracy with Illumina, we mapped short reads to the reference using BWA (alignment score > 30) and computed the average coverage for each organism. Our abundance estimation script *detect_abundance.py* admits two regimes. In the first regime, the abundance of a taxonomic label is defined by number of short read base pairs that map to all contigs having that taxonomic label (normalized by the total number of short reads basepairs that map to some contig). In the second regime, we compute for each taxonomic label the median coverage of all contigs that have that label; the abundance each taxon is obtained by normalizing these median values. We use the first regime by default; we found the second regime to be more accurate when estimating abundance at the species level. In our experiments, we use the second regime when dealing with species and the first regime in all other cases.

Validation of bacterial haplotypes using short-read data

We used Illumina 101-bp short read sequencing to verify the accuracy of the haplotypes identified in both the mock and the gut samples. We first generated a Fasta sequence for each haplotype produced by Lens using the *make_fasta.py* script in the Lens package; this script takes the Fasta sequence of the contig in which a haplotype is reported and replaces the base pairs at which there are putative variants so as to make them consistent with the given haplotype. We measure our accuracy over alleles present in haplotypes at variant positions in the haplotype. Thus, if at position *p* there is support for two bases (A and T) and three haplotypes cover that position (two of which carry A, and one of which carries T), we say that there are three alleles, and we consider all of them in our verification process. To perform verification, we generate a modified fasta sequence by taking the interval from the first to the last variant in the haplotype, plus a 75 bp window on each side. We then align all available short sequencing reads against these haplotypes using Bowtie 2.2.5 with the –k 5 flag (report up to 5 of the best alignments) and all other parameters set to default. Note that we use all available shotgun sequencing reads at this step, and not only ones subsampled for the main analysis (see "Metagenomic sample preparation"). We then examine whether shotgun reads align to regions with haplotype variants in a way that confirms these variants. For SNVs, we say that a read supports a variant if it aligns to its position and carries the haplotype allele at its matching base pair with a PHRED qscore of at least 20. Similarly, we say that a read supports an insertion if it contains a subsequence that matches perfectly the inserted sequence. A read supports a deletion if it aligns to the surrounding base pairs without an insertion. We say that a haplotype variant is supported by short reads if it has at least 2 supporting short reads. We implemented this verification procedure in the scripts *verify_snps.py* and *verify_indels.py* that are part of Nanoscope and executed it on the haplotypes found in both the gut and the mock metagenomes. Overall, we found very high concordance between short and long SNVs (97%) as well as high concordance over indels (99%); see Supplementary Table 27. We explain the high accuracy over indels by the fact that the criterion for calling an indel (which requires three perfectly aligning reads that carry that indel) is relatively more stringent than it is for SNVs. Also, note that some haplotypes may not be covered by short reads by chance and the above numbers are lower bounds on the true accuracy. However, they are close to the estimates reported by Schlossnig et al. (2012) for a very similar variant calling method (these authors reported approximately 2% accuracy for calling SNPs). Finally note that this strategy may be implemented on the long reads themselves instead of the haplotypes; however that would be very computationally demanding as each read will have a large number of possible matches (depending on the long read coverage) and modern BWT-based aligners are not suited to handle such input.

Comparison to alternative methods

Comparing Lens to existing bacterial variant callers

We compared the Lens to the PILON bacterial variant caller on the task of finding SNPs and short indels with respect the mock metagenome reference using our three synthetic long read libraries. We did not call variants using short reads, as we only use variants visible via long reads in our downstream haplotyping analysis.

We aligned the three long read libraries to the mock metagenome reference fasta using bwa mem 0.7.5a with default parameters and used the resulting bam files as input for both Lens and PILON. The Lens variant caller (using default parameters) called 335 variants from this file. We ran PILON 1.11 on the same input file with the –diploid and –min-depth 3 flag (the latter, to be comparable with the Lens cutoff for calling a variant), and with all other parameters set to default. PILON called more than 5775 variants on the same dataset, and 1876 of these had the PASS flag, indicating that they passed quality control.

We proceeded to compare the variants called by each method. Of the 335 variants called by Lens, 217 were among the 1876 called by PILON, and 117 were not. Manually inspecting 10 random variants in IGV revealed that 9 of these 10 variants were surrounded by a large number of other variants (10 or more SNPs in a 500bp window around the analyzed variant). A total of 6 of these 9 windows contained indels in addition to SNPs. Furthermore 5 of the ten analyzed variants were in regions with a 50% or more drop in coverage compared to the average level in a 30Kbp window around that variant.

This suggests that the PILON variant caller models the error profile of Illumina reads when making its variant calls, and this profile is not compatible with that of long reads.

To verify the quality of the variants called by Lens we devised a verification scheme in which we align short reads to the bacterial haplotypes obtained from these variants (see the section on short read validation below). We found that more than 99% of the alleles carried by the bacterial haplotypes at the variant positions could also be confirmed by short reads (Supplementary Table 29). This suggests that PILON makes false negative calls on long read data.

Next, we examined in more detail the PILON variants not reported by Lens. Of these, the vast majority was supported by a single long read. Since we require that our haplotypes be supported by at least two reads over 75% of their SNPs, we discarded all such positions from the 1876 reported by PILON. This left 325 positions with support from at least two long reads that were also tagged with the "PASS" flag. We ran the Lens haplotyping algorithm over these variants (we ran the scripts *make_reads.py* and *detect_subspecies.py* using default parameters), which yielded only 17 haplotypes harboring at least one non-reference allele. In comparison, Lens produced 77 such haplotypes from the 335 positions that it found. We then extracted a list of positions in the haplotypes that carried a non-reference allele and found that each of the 17 haplotypes carried exactly one non-reference allele. Further inspection revealed that 81% of reads overlapped only one of the 325 variant positions reported by PILON, and thus could not be used for haplotype assembly. It thus appears that PILON does not call variants using long reads that align to the reference with more than 2-3 mismatches; this appears to be again due to the fact that PILON models the error profile of short reads.

Finally, we used PILON 1.11 with –diploid and with default parameters to call variants in the real metagenomic dataset. We found that PILON had high memory requirements and we could not get it to run with 90G of RAM. Since PILON used 25G of RAM on the mock metagenomic data, and our real data contains substantially more base pairs in very long contigs (the longest of which are >4X longer than those in the mock data), we estimate that PILON may take >100G of RAM to run on this dataset.

Another potential alternative to Lens is the program MaryGold, which calls variants in bacterial genomes that directly from an assembly graph. However, this method is not applicable to our setting; although Celera and SOAPdenovo can individually output their assembly graphs in AMOS format, we cannot provide MaryGold with an assembly graph of both short and long reads. Furthermore, in order to call variants using long reads, we would need the assembly graph of the raw short reads used to construct the long reads via subassembly, and such a graph is not available to us in principle, since the subassembly software is run within Illumina Basespace.

Comparison to a long read SMRT mock metagenomic dataset

To further study the effectiveness of long reads for metagenomic analyses, we compared our results based on Tru-seq synthetic long reads to a publicly available dataset of SMRT[17] long reads that were generated for the mock metagenomic community. This dataset was generated for an even abundance mock metagenomic sample; in our sample, bacterial species had uneven abundances. Nonetheless, several bacteria were covered by both types of reads, which allowed us to perform a comparison.

We used the PBCR pipeline (part of the Celera assembler 8.3rc1) with the MHAP[18] alignment module to self-correct and assemble 70X of SMRT long reads from the even mock metagenomic sample (we downloaded the data from https://github.com/PacificBiosciences/DevNet/wiki/Human_Microbiome_Project_MockB_Shotgun):

```
wgs-8.3rc1/Linux-amd64/bin/PBcR \
  -libraryname pb-hmp-mock-even \
  -fastq pb-mock-even.fastq \
  -length 500 \
  -threads 40 \
  -partitions 500 \
  genomeSize=83000000
```

We also aligned the raw PacBio reads to our assembled contigs using pbalign, and performed Quiver error correction on the contigs:

```
pbalign raw-reads.bas.h5 asm.ctg.fasta alignment.cmp.h5  --
nproc 16 –forQuiver;
```

```
python variantCaller.py --algorithm=quiver -j8
alignment.cmp.h5 -r asm.ctg.fasta -o out.gff -o out.fasta
```

Assembly quality

Compared to the synthetic long read assembly, the resulting contigs had a lower rate of misassemblies and point mutations (105 misassemblies in 53 Mbp of MHAP contigs, compared to 121 misassemblies in 46 Mbp of contigs assembled from synthetic long reads; Supplementary Table 28). The normalized misassembly rate for MHAP was 1.98/Mbp, compared to 2.63/Mbp for synthetic long reads (a 25% reduction in error). It thus appears that the additional subassembly step increases the total amount of misassemblies; however, the misassembly rate remains on the same order of magnitude as that of the SMRT technology, and it therefore seems like the assembler is able to discard most locally misassembled reads (recall from Supplementary Table 4 that there were about 1800 misassemblies in the raw reads). The MHAP contigs had also a longer N50 length (142 Kbp, compared to 91 Kbp), although this can be in part attributed to the even bacterial abundances in the SMRT sample.

We also attempted to estimate the quality of our assembly on the more complex *bona fide* human gut microbiome sample. We first attempted to use REAPR, an existing state-of-the-art reference-free assembly validation tool that can assess the quality of assembled contigs. REAPR uses the distance between aligned pair-end reads to detect misassemblies, and therefore works best with long mate-pair reads. Because such reads were not available for our dataset, we instead used REAPR with standard paired-end reads. REAPR did not detect any misassemblies based on this input, which is likely due to a small insert size. We are unaware of any existing tools that can perform reference-free assembly validation based on long read data, and this remains an open area for further research as long read technologies become more popular.

Variant calling

The error rate for indels, however, was significantly higher in the MHAP contigs, which is an artifact of the SMRT error model and of the subsequent error correction process. We then mapped the error-corrected SMRT long reads to the known mock metagenome using bwa mem 0.7.5a (with default parameters) and used Lens to identify genomic variants and place them into haplotypes. Because of the high indel error rate of the reads, we only considered SNVs; default Lens parameters were used for SNV calling.

We verified the validity of the SMRT-derived haplotypes using short reads following a very similar strategy to the one we used for validating haplotypes derived from synthetic long reads (see above). The only difference was that we used the true mock reference genomes in place of the assembled contigs; this allowed us to examine the variant calling performance of the two methods in isolation from their ability to perform de-novo assembly. In total, 57469 alleles specific to a bacterial strain were present in the SMRT haplotypes; of these, 20258 were located in regions with more than 20x coverage in shotgun reads (Supplementary Table 29). However, only 2663 of these 20258 alleles carried a variant base-pair with respect to the mock metagenome reference. Thus most SMRT variants could not be assembled into haplotypes that met our quality threshold of 2x support over >75% of positions, a first indication of the lower quality of these haplotypes. We further analyzed the quality of the haplotypes by mapping to them the Illumina shotgun reads according to the same strategy that we used for verifying haplotypes derived from long reads. We found that only 17915 variants (88%) could be supported by short reads; more problematically, only 389 of the 2663 alleles carrying a non-reference base-pair (15%) could be confirmed.

For comparison, we repeated the same analysis using our three long-read libraries. More than 99% of all variants could be confirmed, and the absolute numbers of confirmed alleles that carrier a non-reference base-pair (271) was similar to that for SMRT reads (389; note however that the regions with SMRT and synthetic long read coverage are not exactly identical). This suggests that a lot of SNVs found using SMRT reads are artifacts of the error-correction process. We conclude that SMRT reads are useful for constructing draft contigs from bacteria, but have difficulty in identifying bacterial mutations and haplotypes.

Comparison of the merged assembly strategy to a joint assembly using SPAdes

To assess the effectiveness of our assembly merging strategy, we used SPAdes 3.5.0 to assemble long and short reads jointly. In all experiments, we set K=127 and all other parameters to default. On the mock metagenome, SPAdes produced fewer errors and longer contigs than our merging strategy (Supplementary Table 30). However, it assembled a much smaller fraction of the mock metagenome than the combined strategy (28 Mbp, compared to 46 Mbp) and the amount of genes and operons assembled by SPADES was 50% that of the assembly merging approach.

Curiously, the amount of sequence assembled by SPAdes (28Mbp) is comparable with that of Soapdenovo2 (26 Mbp). We explain this as follows: a De Brujn graph assembler like SPADES breaks long read data into k-mers, thus partly discarding contiguity information. It can use this information indirectly by threading through repeat structures in the DBG graph; however, it is difficult to resolve short tandem repeats via threading using 10 Kbp reads. Therefore, SPAdes performs poorly in regions where it does not have short read (mate-pair) data to resolve such repeats; in regions where it has both types of data however, it can resolve both long and short repeats and assembles the reads very well. On the gut metagenome, SPAdes again assembled reads into longer contigs than the merging strategy; however these contigs spanned 50% fewer base pairs and contained 50% fewer ORFs as predicted by Quast and Metagenemark (Supplementary Table 31).

Finally, we evaluated SPAdes on another recent metagenomic dataset also sequenced using Tru-seq synthetic long reads (see “Comparison to the study of Sharon et al.” below). This study had very limited amounts of both short and long read sequencing for each sample. On that dataset, SPAdes produced contigs that were 2x longer than ones from the assembly merging approach, while assembling only 15% fewer sequence; this indicates that there may be cases where SPAdes has advantages over assembly merging. Because of the many tradeoffs that arise from using SPAdes, we include it as an optional assembler in Nanoscope that can be activated via a command-line parameter.

Comparison to the study of Schlossnig et al.

Schlossnig et al.[19] used shotgun sequencing to analyze the strain-specific variants present in the gut microbiomes of a large cohort of human subjects. We performed similar population genetics analyses using reference bacterial genomes listed by these authors. Based on their criterion that bacterial species should have at least 40% of their genomic sequences covered by reads, we extracted 10 reference genomes for downstream analysis. This smaller number of genomes is likely due to our lower sequencing depth compared to a pooling dataset of 252 microbiome samples studied by Schlossnig et al. To specifically compare the population metrics evaluated in their paper, we calculated SNP density and nucleotide diversity (π), which accounts for all possible variations at each base of the genome. We found that they are highly correlated and comparable to the results in Schlossnig et al. (Supplementary Figure 14). We used these two parameters because they comprehensively summarize the population diversity within a population of closely related bacterial species. Specifically, we observed that the SNP density (measured in SNPs/kb) fell in the range of 7.6-46.0, while the nucleotide diversity π as between 10^(-3) and 10^(-2). We opted not to calculate pN/pS like Schlossnig et al., as we feel the above two parameters confirmed our findings sufficiently. Our results indicate that our SNP calling procedures are in line with the methods previously described, yielding comparable population genetics parameters.

Comparison to the study of Greenblum et al.

Recently, Greenblum et al.[20] examined strain-level copy number variation in the human gut metagenome. Their analysis differs from ours in several respects. First, their pooling approach is unable to consistently produce clusters that represent a single bacterial species. Indeed, the clusters of reads that they report (which should represent individual bacterial species) actually contain sequence that maps to multiple genera using standard metagenomic binning programs (for example, cluster no. 96 – despite being labeled as “Escherichia coli” across many figures in the paper – actually covers three genera: Salmonella, Shigella, and Escherichia). Our haplotypes on the other hand, belong to a single bacterial contig, which can only belong to a single species. Furthermore, the deconvolution approach used by the authors on two genomic clusters only works when these clusters contain a lot of species (as indicated by the authors) and that also have a lot of reference genomes sequenced, which is the case of *E. coli,* but not so much for other less sequenced genera. Our approach helps circumvent these issues by directly recovering genomes via de-novo assembly. This assembly-based approach can uncover previously undescribed genomes and the resulting de-novo contigs can be used as a reference to analyze SNPs and CNVs (potentially using the method of Greenblum et al.). Finally, we would like to point out that the biological significance of bacterial CNV has not yet been established either in traditional microbiology genetic studies or in the setting of microbiome. We therefore choose not to analyze the CNVs in our genome and focus on SNVs and indels instead.

Comparison to the study of Sharon et al.

While our study was under review, a related study was published by Sharon et al.[21]; this study used Tru-Seq synthetic long reads to analyze a soil environmental sample. Their work differs from ours in several respects. First, Sharon et al. use a limited amount of long-read sequencing per sample (about 500 Mbp; one long read library produced almost 1 Gbp of sequence in our experiments). This amount of sequencing was not sufficient to assess all of the benefits of using long reads: our work used 3-7 Gbp per sample, which led to substantial improvements in the reconstruction of bacterial genomes (compared to alternative approaches) that were not reported by Sharon et al. Sharon et al. therefore conclude that Truseq synthetic long reads are a tool primarily for performing species identification and for assessing the diversity of metagenomic communities. Our work shows that with as little as 3 long read libraries, long reads outperform most existing methods on practically all aspects of whole-metagenome analysis, including a recent method that pooled hundreds of samples (Nielsen et al.[22]). Our positive results may change how future microbiome studies are conducted.

More precisely, Sharon et al. perform two types of analyses that are also present in our paper: de-novo assembly and species identification. We propose superior techniques for doing each type of analysis and report better results.

De-novo assembly

First, we demonstrate that long reads can recover megabase-long contigs. For example, we assembled a 2 Mbp contig that was previously identified as a new species using hundreds of pooled samples and recovered as a cluster of short reads (Supplementary Figure 12); our work assembled it from only one metagenomic sample.

This improved assembly quality is not solely due to having more coverage: we have also developed a computational pipeline and an assembly strategy that is more effective than that of Sharon et al. Although two of our tools (Minimus2 and Celera) were also used by Sharon et al., they require substantial tuning to be effective. For example, Sharon et al. could not get Celera (using mostly default parameters) to produce any assemblies from long reads; our version (using custom parameters; see Supplementary Methods) assembled their dataset to an N50 of 13 Kbp; further analysis using SPAdes increased this to 22 Kbp. Furthermore, Minimus2 with default parameters (as proposed by Sharon et al.) did not run at all on our dataset, and had to be recompiled with increased memory buffer sizes (see Supplementary Methods) in order to handle contigs of ~500 Kbp or more.

We now give more details on how performed our comparison. We ran Nanoscope on the 4m soil dataset of Sharon et al., the data for which is available on the SRA. Using the optional SPAdes assembler in Nanoscope (enabled with the *–spades* flag), we assembled 441 Mbp of sequence into contigs with an N50 length of 22 Kbp; Sharon et al. report contig N50 lengths of ~8 Kbp (which correspond to the unassembled long reads; Supplementary Table 32). The contigs we assembled spanned 43,000 genes and contained 19,000 positions with genomic variants over which we could define 94 bacterial haplotypes (Supplementary Table 32).

Species identification

Furthermore, because our method assembled the Sharon et al. data into much longer contigs, it was also easier to perform species identification. Our mapping-based species identification method is more flexible than one based on marker genes (the latter is only a slight generalization of 16S RNA sequencing); using this method, we identified the phylum *Firmicutes* among the top 10 most abundant, while Sharon et al. did not detect this phylum.

In brief, Sharon et al. reported the following top phyla in the 4m sample: Proteobacteria, Chloroflexi, Nitrospirae, Crenarchaeota, Actinobacteria, Bacteroidetes, Planctomycetes, Euryarchaeota, Spirochaetes (in addition to some candidate phyla). The FCP package within Nanoscope (see Supplementary Methods for exact parameters) determined the top 16 phyla (defined as ones to which with >75 Kbp can be mapped) to be Proteobacteria, Chloroflexi, Nitrospirae, Firmicutes, Euryarchaeota, Bacteroidetes, Actinobacteria, Acidobacteria, Deinococcus-Thermus, Cyanobacteria, Planctomycetes, Crenarchaeota, Deinococcus-Thermus, Spirochaetes, Chlorobi (see Supplementary methods). All of the phyla identified by Sharon et al. can be found in this list. The phylum Firmicutes is the 4-th most abundant in terms of number of base pairs that map to it (79 contigs, 1.7 Mbp in total), but the approach of Sharon et al. cannot detect it. See Supplementary Table 33 for more details.

These improved results compared to Sharon et al. represent only part of our contributions. Our paper also introduces important new analyses that can only be done with synthetic long reads.

Bacterial genome phasing

We identify hundreds of thousands of SNVs in the bacterial metagenome and show they are affected by purifying selection and by evolutionary constraints. We then phase these SNVs into long haplotypes that sometimes span > 100 Kbp. This is a unique novel capability of synthetic long reads, compared to Illumina and PacBio; it is made possible by a new algorithm called Lens. Although Sharon et al. also report finding SNVs within a small set of marker genes, they do not investigate further the nature and the properties of these variants. For instance, they do not present evidence that these SNVs are of biological significance and that they are not simply sequencing artifacts.

All of the above analyses are made possible by a standalone tool called Nanoscope that will be made available to the metagenomics community to carry out studies similar to ours.

Implications to human health

Finally and perhaps most importantly, our study is the first to consider the human microbiome and show potential applications of synthetic long reads to the study of human health. For example, our work showed how we can assemble many long flagellar operons, which are strongly associated with pathogenicity.

In conclusion, Sharon et al. used one library for 3 samples as well as an analysis methodology that did not fully explore the benefits offered by synthetic long reads. Our study used a reasonable amount of sequencing (2-7 Gbp) and demonstrated dramatic improvements in every aspect of metagenomic analysis as well as in new types of analyses (e.g. bacterial haplotyping).

Comparison to previous metagenomic analysis methods

Our proposed approach is able to recover bacterial species from a metagenome at a level of quality comparable to that of previous methods involving hundreds of human subjects (Nielsen et al.[22]), multiple DNA extraction methods (Albertsen et al.[23]), or tetranucleotide binning with a mix of Sanger and mate-pair sequencing (Iverson et al.[24]). See Supplementary Table 34 for a high-level overview.

Comparison to Nielsen et al.

A key aspect of our method that is also shared with that of Nielsen et al. is that it can recover bacterial strains from the metagenome; however, our two approaches differ in the way strains are represented. The method of Nielsen et al. outputs clusters of scaffolds (39 Kbp N50; 700 Kbp max.) that belong to the same genome (each cluster encompasses several Mbp of sequence). Our method assembles contigs *de-novo*, and assigns taxonomic labels using a mapping-based approach. These contigs can be much longer than the scaffolds of Nielsen et al.: we report more than ten contigs of >1 Mbp in length, with the longest one being 3.9 Mbp. Furthermore, we find a 2.2 Mbp contig belonging to a species newly discovered by Nielsen et al. and reported as a cluster of dozens of short sequences; these sequences map to our single long contig. However, because we only have one sample, we cannot further cluster our contigs by species like Nielsen et al. In other words, our method offers better contiguity (i.e. longer contigs), but lower completeness (i.e. we cannot identify all the contigs that belong to the same bacterial strain).

The microbial variants and the resulting haplotypes that we uncover offer a different and complementary type of information about microbial strains, compared to that of Nielsen et al. Bacteria from a given strain are constantly evolving and will differ from each other at multiple genomic positions; our method can accurately find these variants and thus provides resolution at the *sub-strain* level. This represents an even higher level of resolution than that of Nielsen et al. In a sense, our method enables one to take a look at evolution as it unfolds and as strains mutate into other strains.

To summarize, both of our approaches potentially recover full bacterial genomes and identify strains; our method has higher assembly contiguity than that of Nielsen et al. (i.e. higher contig N50 length), but lower completeness. While our method cannot cluster contigs into strains, it can further resolve a contig of a given strain into bacterial haplotypes specific to that strain. Finally and very importantly, it should be noted that our method requires only a single sample, as opposed to hundreds.

Comparison to Albertsen et al.

The methods of Albertsen et al. and Iverson et al. have shortcomings with respect to both our method and that of Nielsen et al. The method of Albertsen et al. sequences a sample using two different methods; species with different GC content will have different coverages with each method; this fact can be used to cluster contigs into groups associated with a given type of bacterium. Although this approach recovers about a dozen of (almost) complete genomes, it does not apply to microbes having similar GC content (e.g. similar strains).

Comparison to Iverson et al.

The method of Iverson et al. performs tetranucleotide binning of contigs before assembling them using mate-pairs and Sanger sequencing; although they fully recover a small number of genomes, this approach again cannot disentangle related strains, and furthermore may not be entirely accurate.

Bibliography

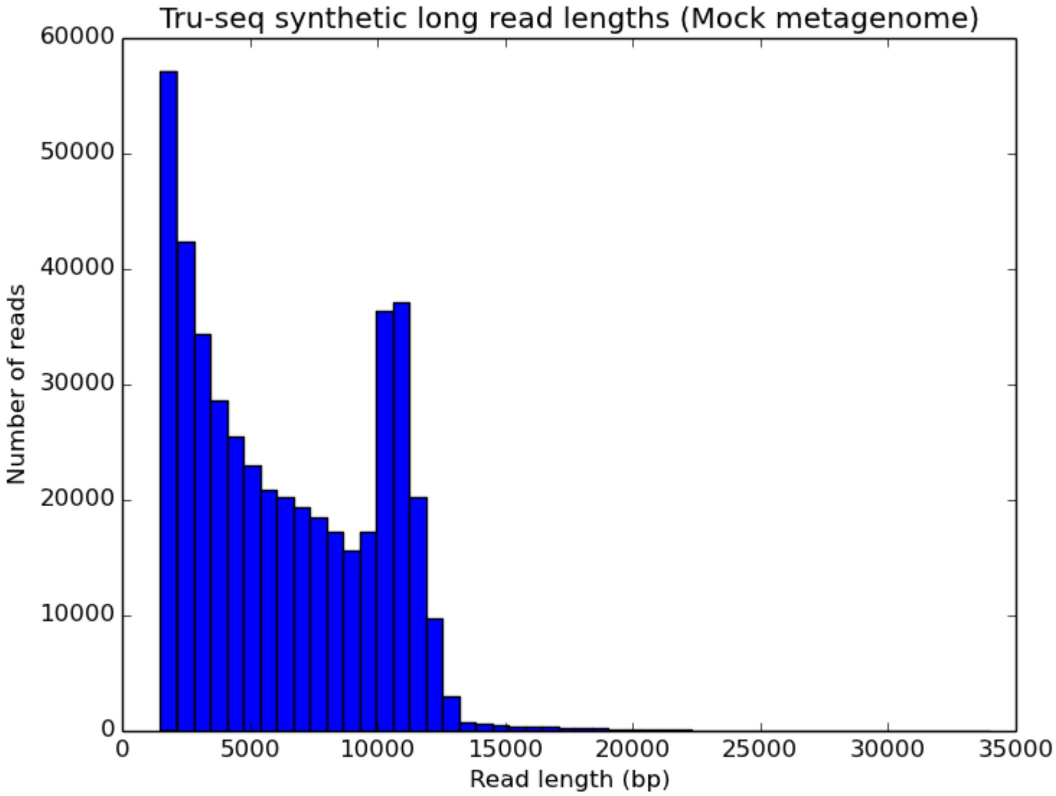
1. Li, R. et al. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Res.* (2009).doi:10.1101/gr.097261.109
2. Myers, E.W. et al. A Whole-Genome Assembly of Drosophila. *Science* **287**, 2196-2204 (2000).
3. Sommer, D., Delcher, A., Salzberg, S. & Pop, M. Minimus: a fast, lightweight genome assembler. *BMC Bioinformatics* **8**, 64 (2007).
4. Walker, B.J. et al. Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. *PLoS ONE* **9**, e112963 (2014).
5. Nijkamp, J.F., Pop, M., Reinders, M.J.T. & de Ridder, D. Exploring variation-aware contig graphs for (comparative) metagenomics using MaryGold. *Bioinformatics* **29**, 2826-2834 (2013).
6. Parks, D., MacDonald, N. & Beiko, R. Classifying short genomic fragments from novel lineages using composition and homology. *BMC Bioinformatics* **12**, 328 (2011).
7. Gurevich, A., Saveliev, V., Vyahhi, N. & Tesler, G. QUAST: quality assessment tool for genome assemblies. *Bioinformatics* **29**, 1072-1075 (2013).
8. Treangen, T. et al. MetAMOS: a modular and open source metagenomic assembly and analysis pipeline. *Genome Biology* **14**, R2 (2013).
9. Schloss, P.D. et al. Introducing mothur: Open-Source, Platform-Independent, Community-Supported Software for Describing and Comparing Microbial Communities. *Applied and Environmental Microbiology* **75**, 7537-7541
10. Duitama, J. et al. Fosmid-based whole genome haplotyping of a HapMap trio child: evaluation of Single Individual Haplotyping techniques. *Nucleic Acids Res.* **40**, 2041-2053 (2012).
11. Berger, E., Yorukoglu, D., Peng, J. & Berger, B. HapTree: a novel Bayesian framework for single individual polyplotyping using NGS data. *PLoS Comput. Biol.* **10**, e1003502 (2014).
12. Aguiar, D. & Istrail, S. Haplotype assembly in polyploid genomes and identical by descent shared tracts. *Bioinformatics* **29**, i352-60 (2013).
13. Niklas, N. et al. cFinder: definition and quantification of multiple haplotypes in a mixed sample. *BMC Res Notes* **8**, 422 (2015).
14. Pulido-Tamayo, S. et al. Frequency-based haplotype reconstruction from deep sequencing data of bacterial populations. *Nucleic Acids Res.* **43**, e105 (2015).
15. Gusfield, D. Inference of haplotypes from samples of diploid populations: complexity and algorithms. *J. Comput. Biol.* **8**, 305-323 (2001).
16. Gusfield, D. Efficient algorithms for inferring evolutionary trees. *Networks* **21**, 19-28 (1991).
17. Eid, J. et al. Real-Time DNA Sequencing from Single Polymerase Molecules. *Science* **323**, 133-138 (2009).
18. Berlin, K. et al. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nat Biotechnol* **33**, 623-630 (2015).
19. Schloissnig, S. et al. Genomic variation landscape of the human gut microbiome. *Nature* **493**, 45-50 (2013).
20. Greenblum, S., Carr, R. & Borenstein, E. Extensive strain-level copy-number

variation across human gut microbiome species. *Cell* **160**, 583-594 (2015).

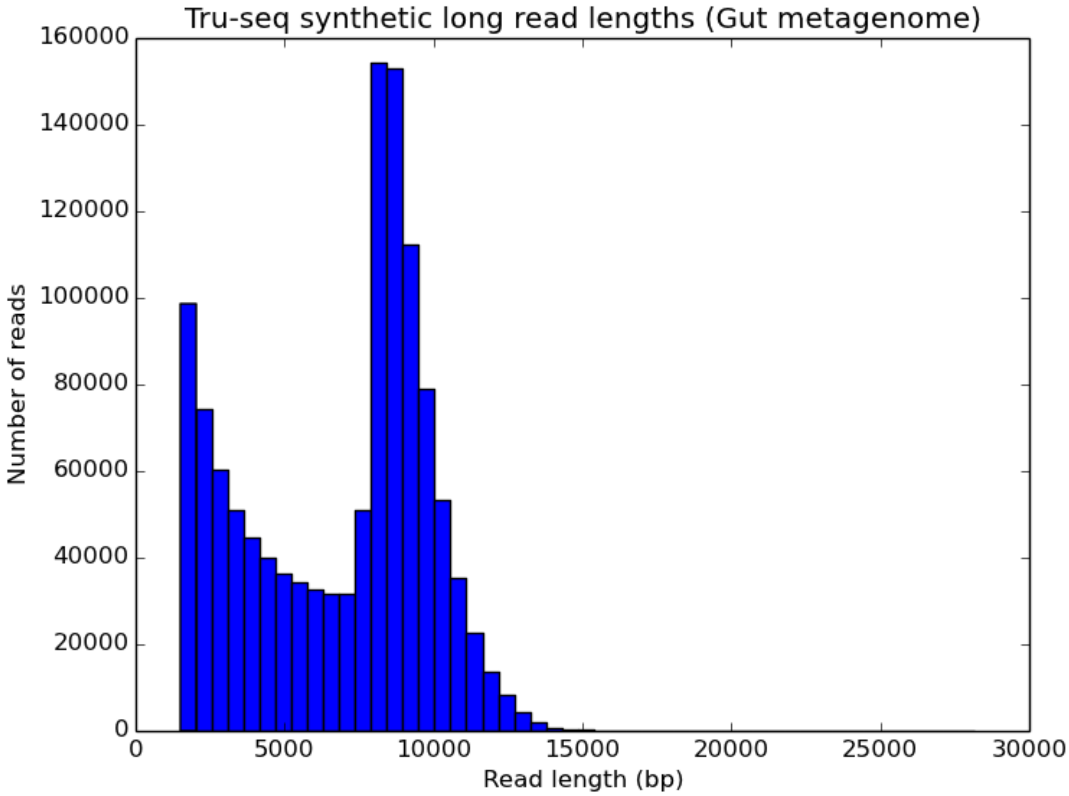
21. Sharon, I. et al. Accurate, multi-kb reads resolve complex populations and detect rare microorganisms. *Genome Res*. **25**, 534-543 (2015).
22. Nielsen, H.B. et al. Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes. *Nat Biotech* **32**, 822-828
23. Albertsen, M. et al. Genome sequences of rare, uncultured bacteria obtained by differential coverage binning of multiple metagenomes. *Nat Biotech* **31**, 533-538
24. Iverson, V. et al. Untangling Genomes from Metagenomes: Revealing an Uncultured Class of Marine Euryarchaeota. *Science* **335**, 587-590 (2012).

Supplementary Figures

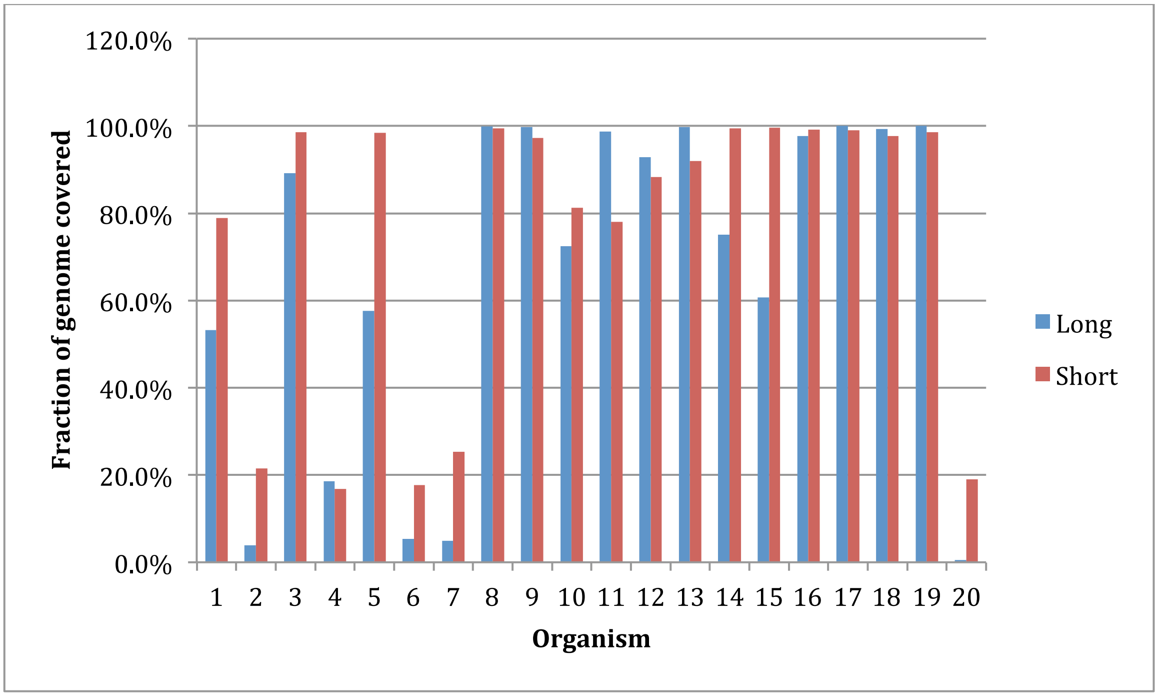
We present below Supplementary Figures 1-14.



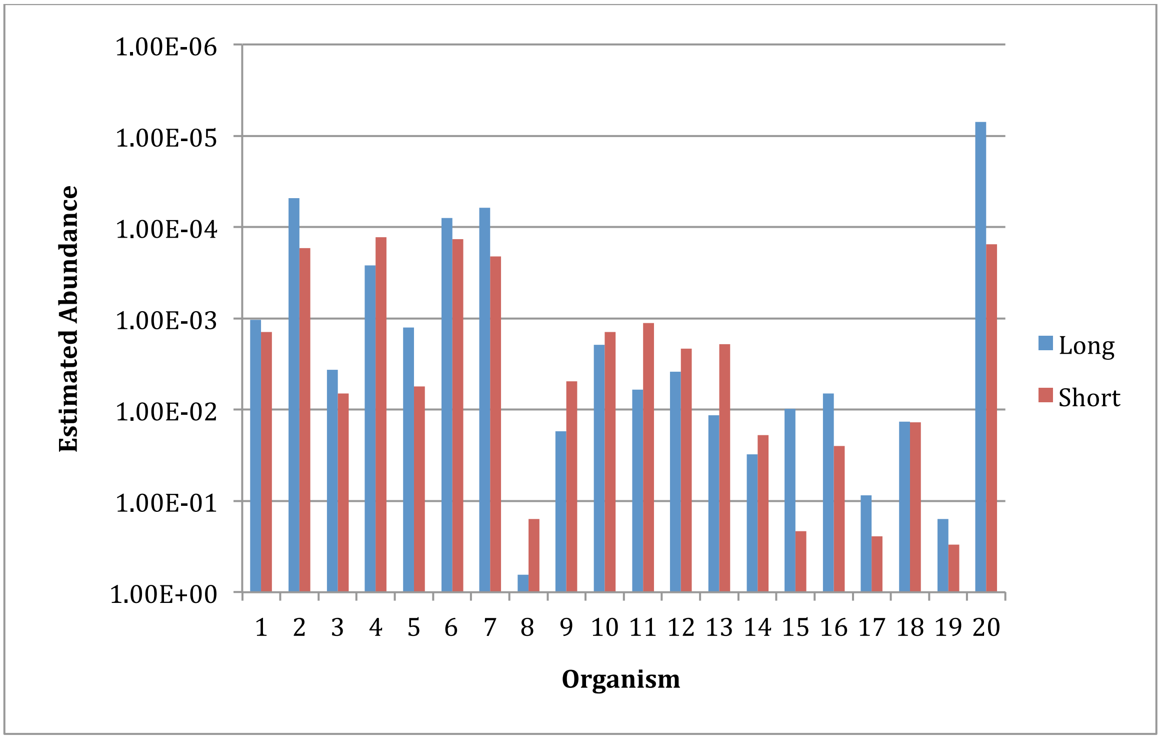
Supplementary Figure 1: Histogram of long read lengths for the mock metagenome



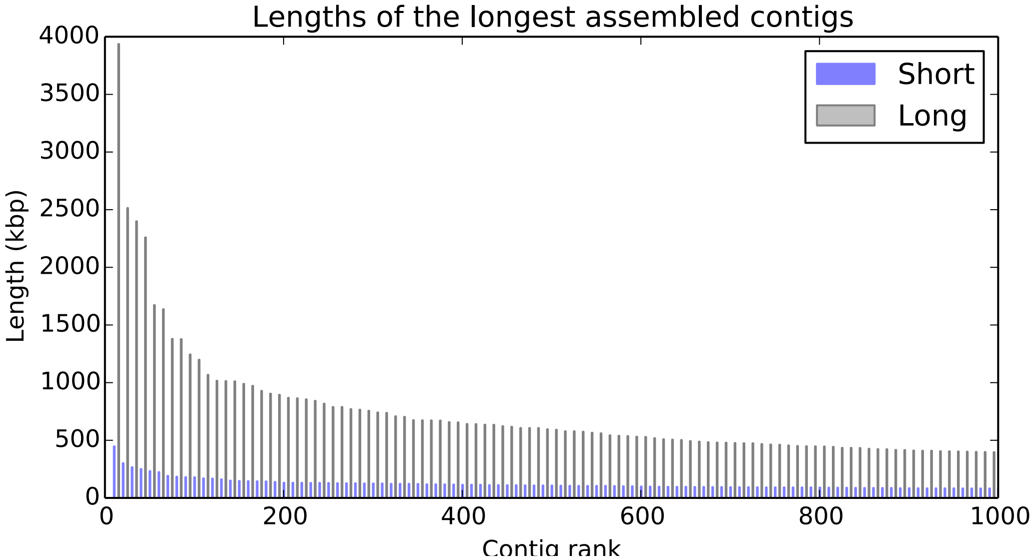
Supplementary Figure 2: Histogram of long read lengths for the real metagenome



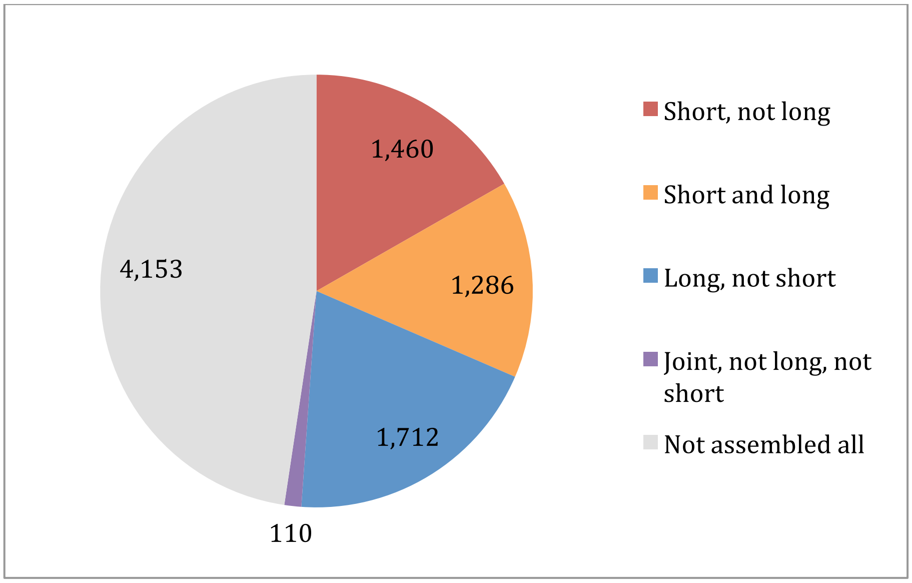
Supplementary Figure 3: Fraction of genome covered with short and long reads, per organism, given an equal number of bases sequenced with each technology. For several organisms, the % coverage greatly varies between the two technologies, indicating different types of bias.



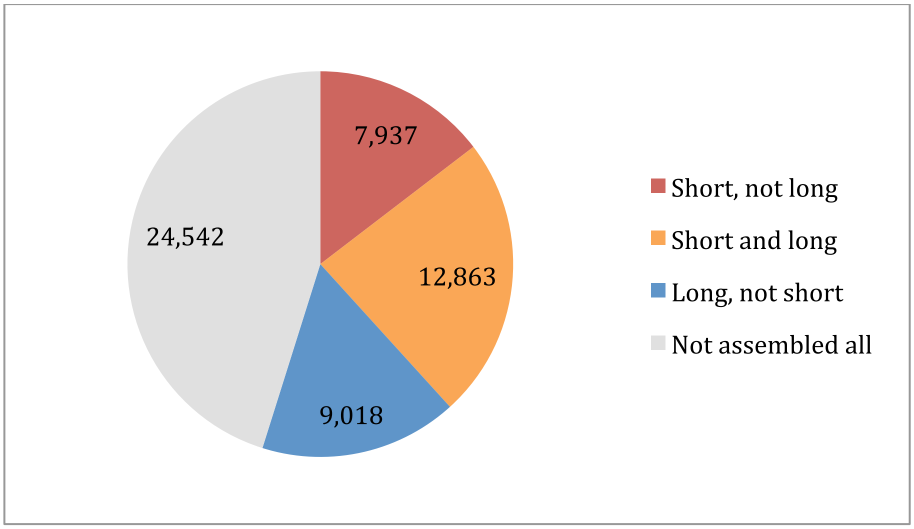
Supplementary Figure 4: Estimated abundance using short and long reads. For several organisms, the estimated abundances vary significantly.



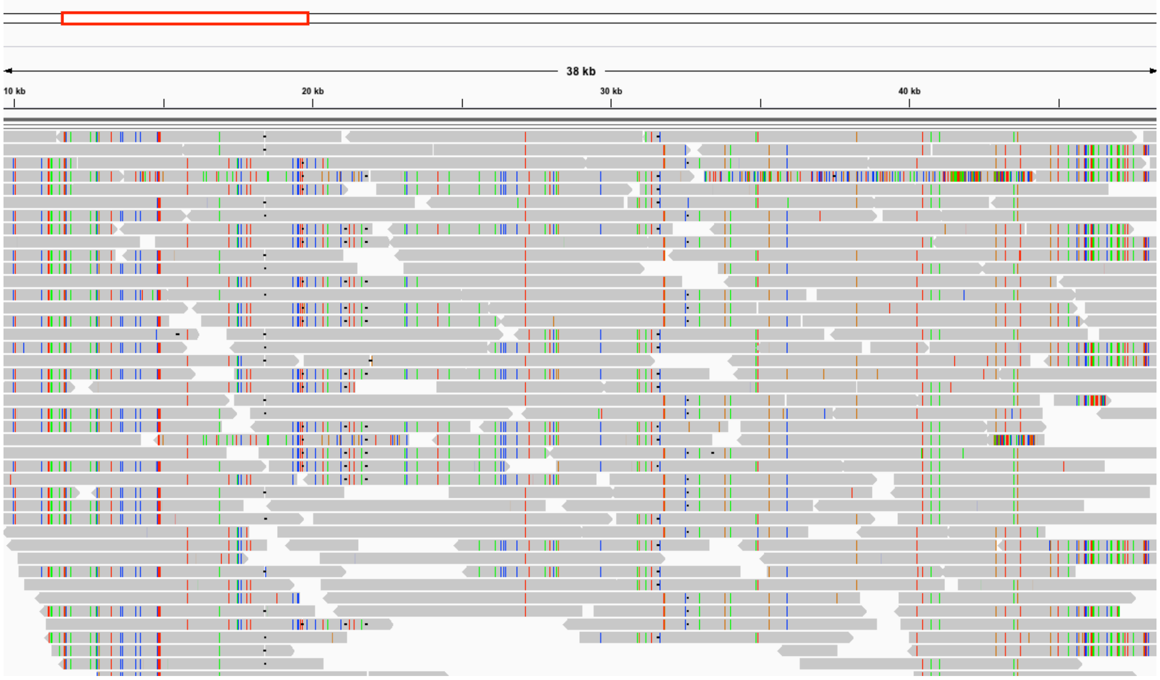
Supplementary Figure 5: Comparison of contig lengths obtained from short and long sequencing (real metagenome). About twenty contigs obtained from long read sequencing are longer than 1 Mbp.



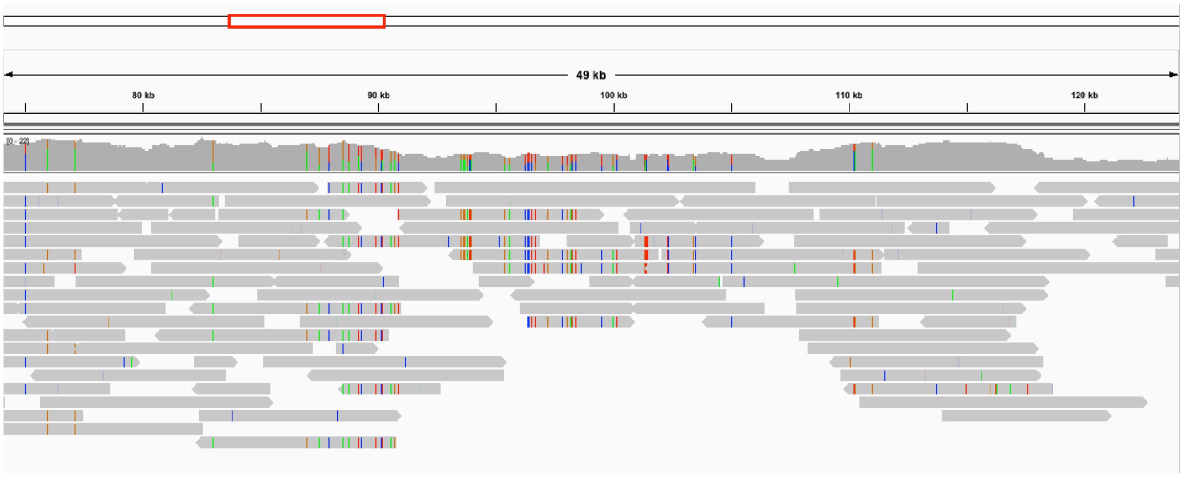
Supplementary Figure 6: Recovery of operons from the assemblies obtained from short reads, long reads, and from the joint assembly (mock metagenome). Short reads were assembled using Soapdenovo2, long reads were assembled with Celera; the two were merged with Minimus2. The joint assembly recovers more than half of all operons, and twice more than only short reads. Interestingly, long and short reads seem to recover different types of operons.



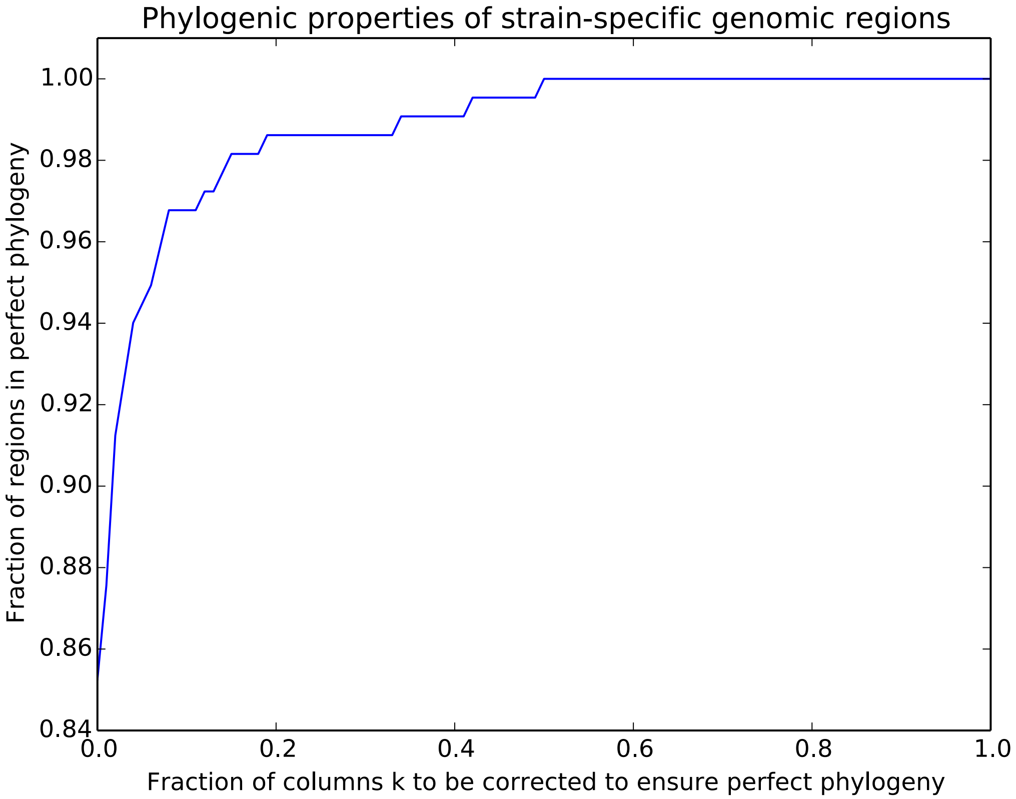
Supplementary Figure 7: Recovery of genes from the assemblies obtained from short reads, long reads, and from the joint assembly (mock metagenome). Short reads were assembled using Soapdenovo2, long reads were assembled with Celera; the two were merged with Minimus2. The joint assembly recovers more than half of all genes, and twice more than only short reads. Interestingly, long and short reads seem to recover different types of genes.



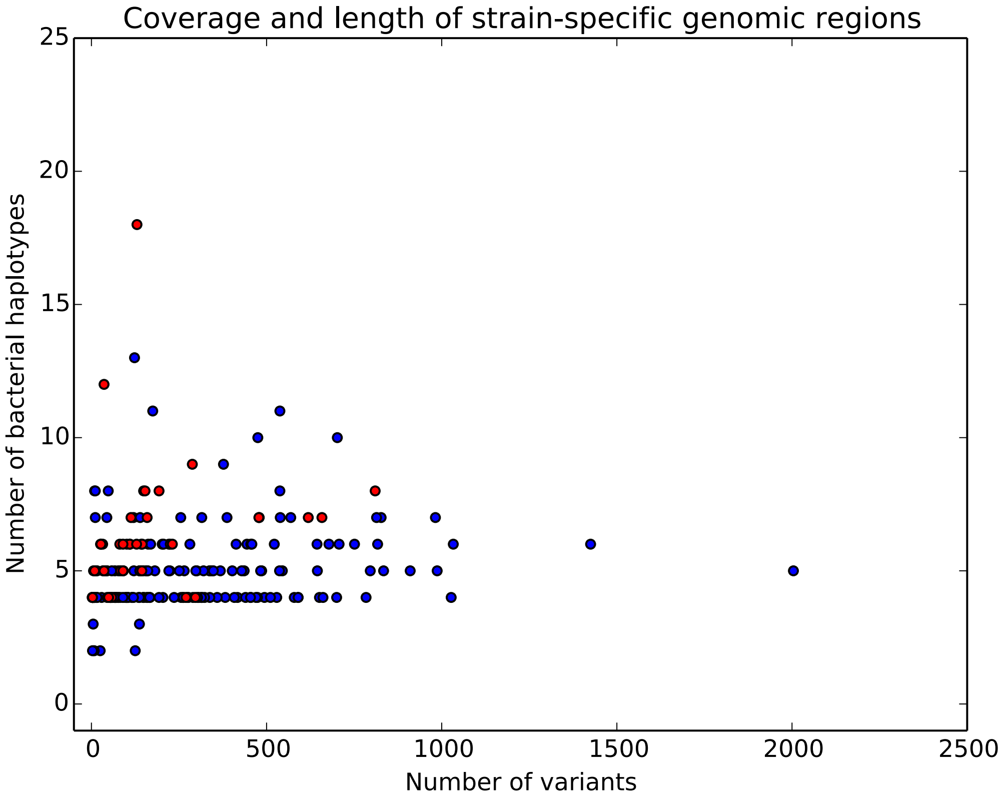
Supplementary Figure 8: Fragment of 110 kbp genomic region in which there is variation between several bacterial subspecies. The contig belongs to the bacterium *Parabacteroides distasonis*.



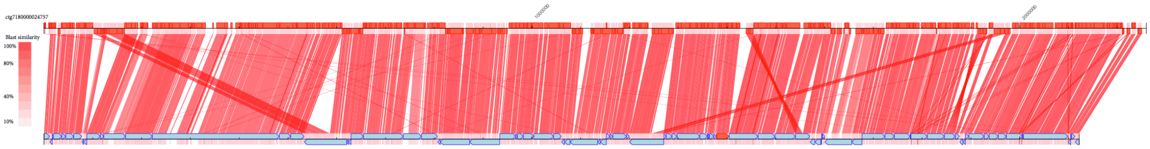
Supplementary Figure 9: Genomic region 50 kbp in length in which there is variation between several bacterial subspecies. The contig belongs to the bacterium *Odoribacter splanchnicus*.



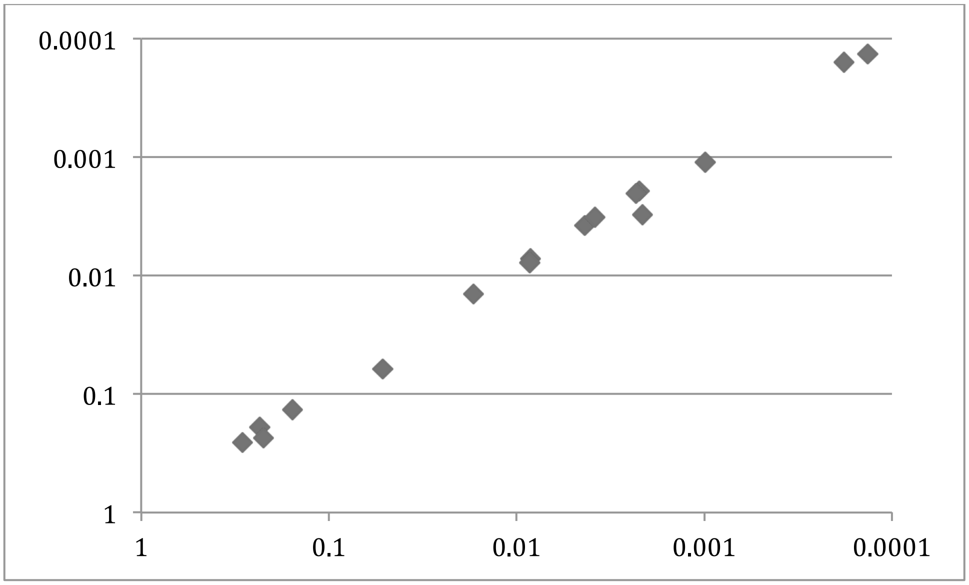
Supplementary Figure 10: Percentage of genomic regions where all haplotypes are in perfect phylogeny, as a function of the percentage of positions that have to be corrected to ensure phylogeny. More than 85% of positions are in perfect phylogeny, and by correcting less than 5% of positions, we can increase this number to more than 92%.



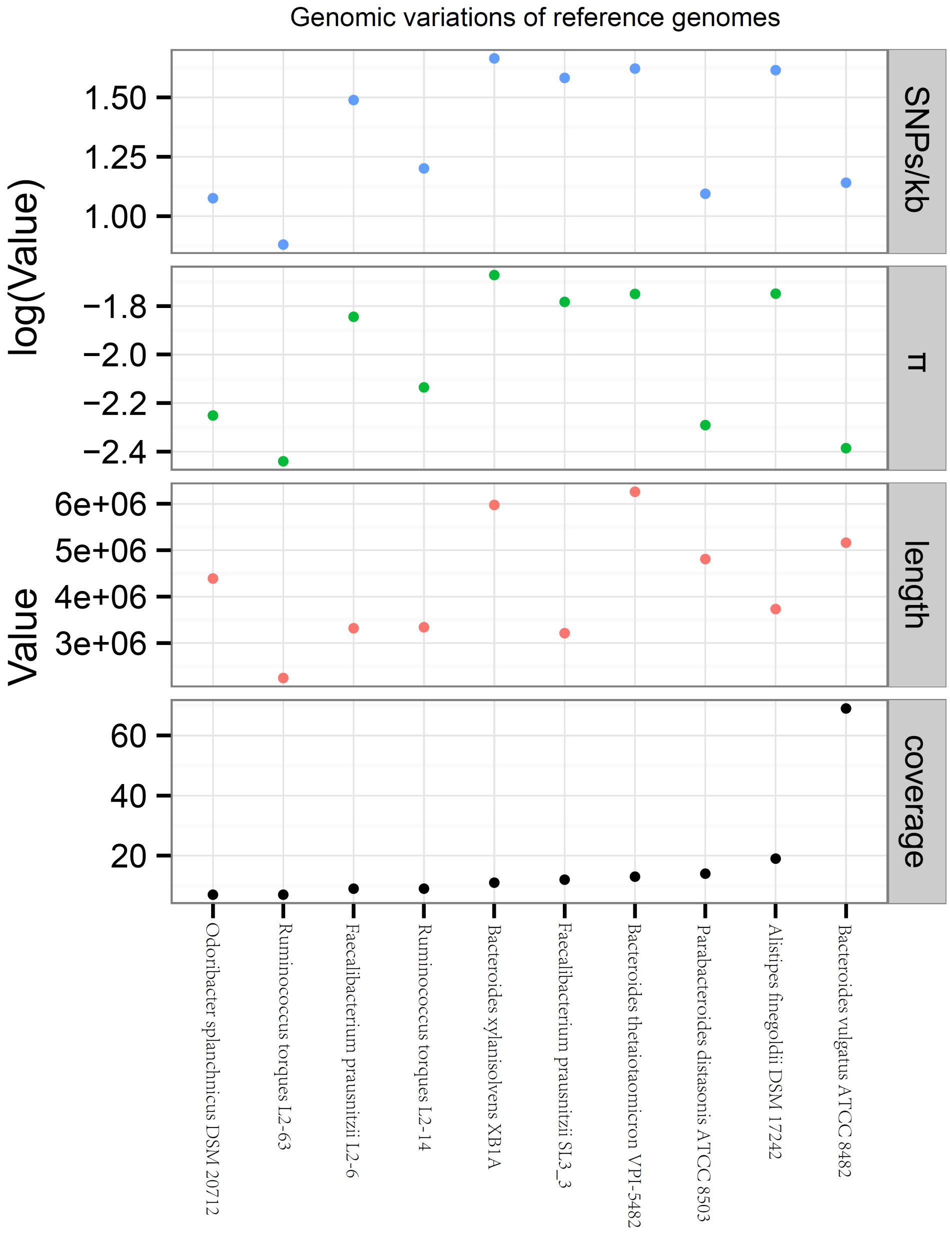
Supplementary Figure 11: Summary of the length and depth of genomic regions at which there is variation among bacteria. Blue regions are in perfect phylogeny, and red regions are not.



Supplementary Figure 12: Recovery of a 2.3 Mbp long contig from a species belonging to the genus Acinetobacter for which no finished genome was previously available. We mapped contigs from an earlier fragmented assembly (bottom) to a 2.3 Mbp contig that we assembled (top). Most of the long contig appears to be covered by shorter contigs from the fragmented assembly.



Supplementary Figure 13: Abundance estimates in the mock metagenome obtained from Nanoscope, compared to the abundances obtained from mapping short reads to the 20 known genome references.



Supplementary Figure 14: Genomic variation statistics for 10 gut microbial species selected from our gut metagenome sample (at least 40% genomes were covered by reads). There is no obvious correlation between genome size/coverage and SNP density and π, which may be due to limited number of genomes analyzed.

Supplementary Tables

We present below Supplementary Tables 1-34.

Species	Genome size (bp)	GC fraction
Acinetobacter baumannii	4,001,456	0.39
Actinomyces odontolyticus	2,393,758	0.65
Bacillus cereus	5,432,260	0.35
Bacteroides vulgatus	5,163,189	0.42
Clostridium beijerinckii	6,000,632	0.30
Deinococcus radiodurans	3,284,061	0.67
Enterococcus faecalis	2,739,625	0.38
Escherichia coli	4,639,675	0.51
Helicobacter pylori	1,667,825	0.39
Lactobacillus gasseri	1,894,360	0.35
Listeria monocytogenes	2,944,528	0.38
Neisseria meningitidis	2,272,360	0.52
Propionibacterium acnes	2,560,265	0.60
Pseudomonas aeruginosa	6,264,404	0.67
Rhodobacter sphaeroides	4,602,949	0.69
Staphylococcus aureus	2,903,080	0.33
Staphylococcus epidermidis	2,564,615	0.32
Streptococcus agalactiae	2,160,267	0.36
Streptococcus mutans	2,032,925	0.37
Streptococcus pneumoniae	2,160,842	0.40

Supplementary Table 1: Composition of the HMP mock metagenomic community. DNA from twenty organisms with known reference genomes was put together in this sample.

Library	# reads	bp	N50 length (bp)
Long 1	149,375	959,367,910	9,241
Long 2	154,443	971,635,090	9,090
Long 3	147,218	940,325,694	9,208
Long (total)	451,036	2,871,328,694	9,189
Short	30,793,158	3,140,902,116	101

Supplementary Table 2: Sequencing libraries for the mock metagenome

Library	# reads	bp	N50 length (bp)
Long 1	173,055	1,144,582,563	8,549
Long 2	170,269	1,146,386,491	8,586
Long 3	180,145	1,207,605,902	8,591
Long 4	172,566	1,160,996,971	8,582
Long 5	175,772	1,211,278,430	8,644
Long 6	177,640	1,266,272,031	8,712
Long 7	177,473	1,258,217,400	8,701
Long (total)	1,226,920	8,395,339,788	8,612
Short 1	41,945,090	4,236,454,090	101
Short 2	38,410,786	3,879,489,386	101
Short (total)	80,355,876	8,115,943,476	101

Supplementary Table 3: Sequencing libraries for the real metagenome

	Long-1	Long-2	Long-3
# contigs (>= 0 bp)	149375	154443	147218
# contigs (>= 1000 bp)	149375	154443	147218
Total length (>= 0 bp)	959218535	971480647	940178476
Total length (>= 1000 bp)	959218535	971480647	940178476
# contigs	149375	154443	147218
Largest contig	32269	33975	30905
Total length	959218535	971480647	940178476
Reference length	83861393	83861393	83861393
GC (%)	48.94	48.88	48.86
Reference GC (%)	43.64	43.64	43.64
N50	9241	9089	9208
NG50	12414	12435	12431
N75	5835	5664	5790
NG75	12055	12081	12074
L50	43747	44462	42822
LG50	2876	2871	2853
L75	75930	77789	74489
LG75	4592	4584	4567
# misassemblies	671	696	684
# misassembled contigs	624	656	645
Misassembled contigs length	3737230	3651499	3720717
# local misassemblies	2409	2448	2405
# unaligned contigs	355 + 2015 pt.	336 + 2112 pt.	318 + 1885 pt.
Unaligned length	1415081	1451166	1324686
Genome fraction (%)	41.09	41.578	40.812
Duplication ratio	27.805	27.83	27.441
# N's per 100 kbp	0.17	0.17	0.19
# mismatches per 100 kbp	12.85	12.93	13.34
# indels per 100 kbp	2.3	2.32	2.18
Largest alignment	32269	33975	30905
NA50	9222	9074	9189
NGA50	12407	12430	12428
NA75	5809	5637	5764
NGA75	12049	12077	12070
LA50	43781	44491	42850
LGA50	2883	2874	2857
LA75	76058	77920	74617
LGA75	4600	4587	4570

Supplementary Table 4: Quality control for the mock metagenome long read libraries.

N	Species	Coverage	Estim. Abund.	% covered	% GC
1	Acinetobacter baumannii	0.826602	1.04E-03	53.2%	38.9%
2	Actinomyces odontolyticus	0.038524	4.84E-05	3.9%	65.4%
3	Bacillus cereus	2.906793	3.65E-03	89.2%	35.5%
4	Bacteroides vulgatus	0.210122	2.64E-04	18.6%	42.2%
5	Clostridium beijerinckii	0.997614	1.25E-03	57.6%	29.9%
6	Deinococcus radiodurans	0.063593	7.99E-05	5.4%	66.6%
7	Enterococcus faecalis	0.048978	6.15E-05	4.9%	37.8%
8	Escherichia coli	514.979955	6.47E-01	100.0%	50.8%
9	Helicobacter pylori	13.812709	1.74E-02	99.8%	38.9%
10	Lactobacillus gasseri	1.548396	1.95E-03	72.4%	35.3%
11	Listeria monocytogenes	4.835126	6.07E-03	98.7%	38.0%
12	Neisseria meningitidis	3.035695	3.81E-03	92.9%	51.5%
13	Propionibacterium acnes	9.174239	1.15E-02	99.8%	60.0%
14	Pseudomonas aeruginosa	24.522923	3.08E-02	75.1%	66.6%
15	Rhodobacter sphaeroides	7.798898	9.80E-03	60.8%	68.8%
16	Staphylococcus aureus	5.3144	6.68E-03	97.7%	32.7%
17	Staphylococcus epidermidis	69.370668	8.71E-02	100.0%	32.0%
18	Streptococcus agalactiae	10.844528	1.36E-02	99.4%	35.6%
19	Streptococcus mutans	125.456057	1.58E-01	100.0%	36.8%
20	Streptococcus pneumoniae	0.005563	6.99E-06	0.6%	39.7%

Supplementary Table 5: Results of mapping long reads to known reference genomes (mock metagenome). Each organism varies in in its coverage with long reads, its relative abundance estimated from the coverage, and the fraction of the genome covered by long reads.

N	Species	Coverage	Estim. Abund.	% covered	% GC
1	Acinetobacter baumannii	2.195619	1.40E-03	79.0%	38.9%
2	Actinomyces odontolyticus	0.265852	1.70E-04	21.5%	65.4%
3	Bacillus cereus	10.39775	6.64E-03	98.5%	35.5%
4	Bacteroides vulgatus	0.203252	1.30E-04	16.8%	42.2%
5	Clostridium beijerinckii	8.645627	5.52E-03	98.5%	29.9%
6	Deinococcus radiodurans	0.212691	1.36E-04	17.6%	66.6%
7	Enterococcus faecalis	0.325533	2.08E-04	25.3%	37.8%
8	Escherichia coli	245.819059	1.57E-01	99.4%	50.8%
9	Helicobacter pylori	7.701225	4.92E-03	97.3%	38.9%
10	Lactobacillus gasseri	2.212187	1.41E-03	81.3%	35.3%
11	Listeria monocytogenes	1.766114	1.13E-03	78.1%	38.0%
12	Neisseria meningitidis	3.34195	2.14E-03	88.3%	51.5%
13	Propionibacterium acnes	3.008918	1.92E-03	92.0%	60.0%
14	Pseudomonas aeruginosa	29.686944	1.90E-02	99.4%	66.6%
15	Rhodobacter sphaeroides	336.527637	2.15E-01	99.6%	68.8%
16	Staphylococcus aureus	39.216208	2.51E-02	99.2%	32.7%
17	Staphylococcus epidermidis	380.161488	2.43E-01	99.1%	32.0%
18	Streptococcus agalactiae	21.615019	1.38E-02	97.6%	35.6%
19	Streptococcus mutans	471.457497	3.01E-01	98.6%	36.8%
20	Streptococcus pneumoniae	0.239976	1.53E-04	19.0%	39.7%

Supplementary Table 6: Results of mapping short reads to known reference genomes (mock metagenome). Each organism varies in in its coverage with short reads, its relative abundance estimated from the coverage, and the fraction of the genome covered by short reads.

N	Species	Short	Long	Both	Only one	% GC
1	Acinetobacter baumannii	78.98%	53.17%	42.02%	48.11%	38.9%
2	Actinomyces odontolyticus	21.49%	3.85%	0.84%	23.66%	65.4%
3	Bacillus cereus	98.52%	89.25%	87.80%	12.16%	35.5%
4	Bacteroides vulgatus	16.75%	18.58%	3.14%	29.06%	42.2%
5	Clostridium beijerinckii	98.45%	57.63%	56.41%	43.26%	29.9%
6	Deinococcus radiodurans	17.63%	5.40%	1.27%	20.48%	66.6%
7	Enterococcus faecalis	25.27%	4.90%	1.34%	27.49%	37.8%
8	Escherichia coli	99.44%	99.98%	99.44%	0.55%	50.8%
9	Helicobacter pylori	97.27%	99.81%	97.10%	2.87%	38.9%
10	Lactobacillus gasseri	81.29%	72.39%	58.32%	37.05%	35.3%
11	Listeria monocytogenes	78.07%	98.73%	77.11%	22.57%	38.0%
12	Neisseria meningitidis	88.34%	92.92%	82.71%	15.84%	51.5%
13	Propionibacterium acnes	92.02%	99.82%	91.87%	8.11%	60.0%
14	Pseudomonas aeruginosa	99.42%	75.08%	74.67%	25.15%	66.6%
15	Rhodobacter sphaeroides	99.60%	60.76%	60.39%	39.58%	68.8%
16	Staphylococcus aureus	99.18%	97.70%	96.88%	3.11%	32.7%
17	Staphylococcus epidermidis	99.09%	100.00%	99.09%	0.91%	32.0%
18	Streptococcus agalactiae	97.65%	99.37%	97.14%	2.73%	35.6%
19	Streptococcus mutans	98.61%	100.00%	98.61%	1.39%	36.8%
20	Streptococcus pneumoniae	18.99%	0.56%	0.14%	19.27%	39.7%

Supplementary Table 7: Coverage statistics for species present in the mock metagenome. For each species, we indicate the percent of the genome covered by short reads, long reads, by both, and by only one technology. For many organisms, short and long reads appear to cover different regions of the genome.

	Short	Long	Joint
# contigs	92247	24199	34786
Largest contig	628113	3936002	3936002
Total length	232738162	609559313	656202352
GC (%)	45.84	46.85	46.85
N50	8687	37319	49208
N75	1635	16435	18127
L50	4263	2832	2367
L75	21837	9466	8301
# N's per 100 kbp	243.51	0	116.82
# predicted genes (unique)	274600	523358	552680
# predicted genes (>= 0 bp)	275200	570360	623203
# predicted genes (>= 300 bp)	214963	493158	533061
# predicted genes (>= 1500 bp)	25898	85342	90978
# predicted genes (>= 3000 bp)	3300	10404	11163

Supplementary Table 8: Assembly metrics for the real metagenome. Short and long read libraries were assembled with Soapdenovo2 and the Celera assemblers, respectively. The results were merged using Minimus2 to produce a joint assembly. We report quality control metrics from the QUAST package.

	Short	Long	Joint
# contigs (>= 0 bp)	6732	1412	3092
# contigs (>= 1000 bp)	3107	1412	1668
Total length (>= 0 bp)	26883707	33907364	46410922
Total length (>= 1000 bp)	24478720	33907364	45482193
# contigs	6732	1412	3092
Largest contig	413959	1687316	1687331
Total length	26883707	33907364	46410922
Reference length	83861393	83861393	83861393
GC (%)	50.2	44.38	46.67
Reference GC (%)	43.64	43.64	43.64
N50	19122	43803	91895
N75	4369	16870	25235
L50	227	128	102
L75	1038	454	364
# misassemblies	29	89	121
# misassembled contigs	23	75	95
Misassembled contigs length	185760	3960997	6541944
# local misassemblies	553	79	651
# unaligned contigs	332 + 54 part	3 + 13 part	290 + 58 part
Unaligned length	262197	53073	265150
Genome fraction (%)	31.583	39.626	52.986
Duplication ratio	1.005	1.022	1.041
# N's per 100 kbp	101.44	0	55.5
# mismatches per 100 kbp	3.9	8.92	7.9
# indels per 100 kbp	2.24	2.54	2.69
# genes	17753 + 3917 part	18909 + 1594 part	27224 + 2485 pt.
# operons	2746 + 1563 part	2998 + 953 part	4567 + 1079 part
Largest alignment	413959	1662515	1662529
NA50	18948	42994	83121
NA75	4324	16131	23510
LA50	229	131	107
LA75	1047	468	383

Supplementary Table 9: Assembly metrics for the mock metagenome. Short and long read libraries were assembled with Soapdenovo2 and the Celera assemblers, respectively. The results were merged using Minimus2 to produce a joint assembly. We report quality control metrics from the QUAST package.

	Genes	Operons
Short	17,753	2,746
Long	18,909	2,998
Joint	27,224	4,567
Short, not long	7,937	1,460
Short and long	12,863	1,286
Long, not short	9,018	1,712
Joint, not long, not short	0	110
Not assembled all	24,542	4,153
Total	51,766	8,720

Supplementary Table 10: Recovery of genes and operons from the assemblies obtained from short reads, long reads, and from the joint assembly (mock metagenome). Short reads were assembled using Soapdenovo2, long reads were assembled with Celera; the two were merged with Minimus2. The joint assembly recovers more than half of all operons, and twice more than only short reads. Interestingly, long and short reads seem to recover different types of operons.

Operon ID	Gene ID	Genomic coordinates
e-coli-551	flgA	1129427 1130086
e-coli-551	flgM	1129058 1129351
e-coli-551	flgN	1128637 1129053
e-coli-552	flgB	1130241 1130657
e-coli-552	flgC	1130661 1131065
e-coli-552	flgD	1131077 1131772
e-coli-552	flgE	1131797 1133005
e-coli-552	flgF	1133025 1133780
e-coli-552	flgG	1133952 1134734
e-coli-552	flgH	1134787 1135485
e-coli-552	flg	1135497 1136594
e-coli-552	flgJ	1136594 1137535
e-coli-552	flgK	1137601 1139244
e-coli-552	flgL	1139256 1140209

Supplementary Table 11: Both flagellar operons in E. Coli are assembled using the joint dataset (mock metagenome). The longer operon is 10 Kbp long and contains more than 11 genes.

Operon ID	Length (bp)
e-coli-552	1449
e-coli-551	9968
r-shpaeroides-647	4788
r-shpaeroides-26	13961
r-shpaeroides-656	2918
r-shpaeroides-658	766
p-aeruginosa-627	18422
r-shpaeroides-17	2148
r-shpaeroides-18	714
r-shpaeroides-1960	2489
p-aeruginosa-1933	2511

Supplementary Table 12: Flagellar operons fully assembled within the mock metagenome. The joint assembly enables us to recover entire operons responsible for bacterial motility, an important factor in determining the whether a microbe will be infectious. We are able to recover about half of all such operons present in the data, including all the ones present in E. Coli and R. Sphaeroides. Remaining operons could not be recovered mainly due to the low abundance of their organisms and to gaps in coverage.

Contig ID	Contig len.	Genus	# of operons	# of genes	Max. genes/operon
25887	1.2 Mbp	Eubacterium	3	13	5
48450	1.2 Mbp	Azospirillum	5	20	5
48971	0.9 Mbp	Butyrivibrio	1	3	3
24757	2.3 Mbp	Acinetobacter	10	50	11

Supplementary Table 13: Flagellar operons in human gut metagenome. Out of 16 contigs longer than 0.8 Mbp, four contained flagellar genes. The longest such contig contained 10 operons, the longest of which contained 11 genes.

	Human gut		Mock metagenome	
	Base pairs	Variants	Base pairs	Variants
Essential genes	65,332	11	973,673	19
Non-essential genes	5,272,922	1,863	4,529,662	203
% Essential	1.22%	0.59%	17.69%	8.56%

Supplementary Table 14: Frequency of strain variants in the human gut and mock metagenomes across essential and non-essential genes. Essential genes from the OGEEDB essential genes database were mapped to assembled gut metagenome contigs using Mummer. We found 30 contigs with matches, and their total length was 5,338,254 bp. A total of 65,332 bp mapped to essential genes; all other regions within the mapped contigs were considered non-essential. The essential genes contained 11 variants out of 1874 in total within the mapped contigs. This represents a statistically significant enrichment ($p < 0.02$ using the χ^2 test). The frequency of strain variants in the mock metagenome across essential and non-essential genes was also significant ($p < 1e-3$ using the χ^2 test).

Cutoff threshold	2	3	4
Variant positions	1626	1304	1210

Supplementary Table 15: Number of positions with strain variants identified by Lens in the mock metagenome as a function of the coverage cutoff. All other parameters were set to default: the minimum allele frequency was set to 0.1 and the minimum PHRED qscore to 15. In all cases, small numbers of variants were found in the mock metagenome.

SNPs	In essential genes	In all genes
Low effect	10	271
Moderate effect	7	210
High effect	0	6

Supplementary Table 16: Predicted effects of strain-specific variants found in the mock metagenome (*E. Coli* only). Effects were predicted using the SNPEff software package. The difference in the number of moderate effect SNPs is statistically significant (p < 1e-2; chi-squared test).

	Short	Long
contigs	271,825	24,199
contigs classified by LCA	126,272	14,858
contigs classified by NB	29,880	21,722
% contigs classified by LCA	46.45%	61.40%
% contigs classified by NB	10.99%	89.76%
total bp	175,111,648	609,559,313
bp classified by LCA	113,235,079	398,131,752
bp classified by NB	94,155,101	582,671,628
% bp classified by LCA	64.66%	65.31%
% bp classified by NB	53.77%	95.59%

Supplementary Table 17: Results of assigning taxonomic labels to contigs using the FCP classification package at the species level on the real metagenomic sample.

	Short	Long
ctg class. correctly	1.000000 (2987/2987)	1.000000 (1331/1331)
bp class. correctly	1.000000 (23495094/23495094)	1.000000 (29986824/29986824)
ctgs unclassified	0.006651 (20/3007)	0.020603 (28/1359)
bp unclassified	0.008193 (194097/23689191)	0.007295 (220350/30207174)

Supplementary Table 18: Accuracy of the LCA algorithm at assigning taxonomic labels at the species level, assessed on the mock metagenomic sample.

	Short	Long
ctg class. correctly	1.000000 (2998/2998)	1.000000 (1358/1358)
bp class. correctly	1.000000 (23669958/23669958)	1.000000 (30188607/30188607)
ctgs unclassified	0.002993 (9/3007)	0.000736 (1/1359)
bp unclassified	0.000812 (19233/23689191)	0.000615 (18567/30207174)

Supplementary Table 19: Accuracy of the LCA algorithm at assigning taxonomic labels at the genus level, assessed on the mock metagenomic sample.

	Short	**Long**
ctg class. correctly	0.957891 (2434/2541)	0.960120 (1276/1329)
bp class. correctly	0.969226 (22118466/22820762)	0.979460 (29341659/29956984)
ctgs unclassified	0.154972 (466/3007)	0.022075 (30/1359)
bp unclassified	0.036659 (868429/23689191)	0.008282 (250190/30207174)

Supplementary Table 20: Accuracy of the Naïve Bayes algorithm at assigning taxonomic labels at the species level, assessed on the mock metagenomic sample.

	Short	**Long**
ctg class. correctly	0.964002 (2544/2639)	0.968657 (1298/1340)
bp class. correctly	0.976197 (22484253/23032496)	0.984523 (29579871/30044876)
ctgs unclassified	0.122381 (368/3007)	0.013981 (19/1359)
bp unclassified	0.027721 (656695/23689191)	0.005373 (162298/30207174)

Supplementary Table 21: Accuracy of the Naïve Bayes algorithm at assigning taxonomic labels at the genus level, assessed on the mock metagenomic sample.

N	Length (bp)	Genus
1	3,936,007	Odoribacter
2	2,514,024	Bacteroides
3	2,399,651	Bacteroides
4	2,260,142	unclassified
5	1,912,083	unclassified
6	1,672,821	Ruminiclostridium
7	1,636,963	Bacteroides
8	1,428,920	unclassified
9	1,379,634	Bacteroides
10	1,377,560	Acidaminococcus

Supplementary Table 22: Genus-level taxonomic labels assigned to the longest contigs in the real metagenome.

N	Length (bp)	FCP Classification
1	2,259,571	Bacteria;unclassified;unclassified;unclassified;unclassified;unclassified;unclassified;unclassified;
2	1,379,633	Bacteria;Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides;unclassified;unclassified;
3	1,377,544	Bacteria;Firmicutes;Negativicutes;Selenomonadales;Acidaminococcaceae;Acidaminococcus;unclassified;unclassified;
4	1,244,739	Bacteria;Proteobacteria;Alphaproteobacteria;unclassified;unclassified;unclassified;unclassified;unclassified;
5	1,067,321	Bacteria;Bacteroidetes;Bacteroidia;Bacteroidales;unclassified;unclassified;unclassified;unclassified;
6	972,590	Bacteria;Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides;unclassified;unclassified;
7	904,049	Bacteria;Bacteroidetes;Bacteroidia;Bacteroidales;unclassified;unclassified;unclassified;unclassified;
8	868,360	Bacteria;unclassified;unclassified;unclassified;unclassified;unclassified;unclassified;unclassified;
9	855,447	Bacteria;Firmicutes;Clostridia;Clostridiales;unclassified;unclassified;unclassified;unclassified;
10	842,182	Bacteria;Bacteroidetes;Bacteroidia;Bacteroidales;Bacteroidaceae;Bacteroides;unclassified;unclassified;

Supplementary Table 23: Longest contigs in the real metagenome that could not be assigned a taxonomic label at the species level or higher.

N	Length (bp)
1	3,223
2	6,604
3	16,998
4	5,689
5	13,365
6	31,566
7	8,889
8	5,715
9	7,439
10	10,426

Supplementary Table 24: Longest BLAST match from each of the above contigs to a database of known finished genomes.

Bacterium	Abundance
Lawsonia intracellularis	0.13209
Chitinophaga pinensis	0.048586
Clostridium novyi	0.043447
[Eubacterium] eligens	0.036388
[Eubacterium] siraeum	0.032927
Bacteroides helcogenes	0.027292
Marinitoga piezophila	0.027192
Alistipes finegoldii	0.026972
Bacteroides thetaiotaomicron	0.025505
Acholeplasma brassicae	0.025332
Bacteroides fragilis	0.022607
Peptoclostridium difficile	0.021531
Streptobacillus moniliformis	0.020873
Bacteroides salanitronis	0.019356
Bacteroides vulgatus	0.018029
Clostridium botulinum	0.016666
Acholeplasma palmae	0.016552
Odoribacter splanchnicus	0.01433
Eubacterium rectale	0.012471
Bacteroides xylanisolvens	0.012289
Alistipes shahii	0.012105
unclassified	0.011699
Ignavibacterium album	0.011551
Parabacteroides distasonis	0.010399
Akkermansia muciniphila	0.010303
Clostridium cellulovorans	0.009233
Clostridiales genomosp. BVAB3	0.009028
Roseburia intestinalis	0.007073
butyrate-producing bacterium SM4/1	0.00693
Treponema succinifaciens	0.006739
Faecalibacterium prausnitzii	0.005786
Campylobacter jejuni	0.005555
Fusobacterium nucleatum	0.0054
Bifidobacterium longum	0.005242
Streptomyces cattleya	0.005136
Filifactor alocis	0.005085
Lactobacillus ruminis	0.005025
Gottschalkia acidurici	0.005019
Caldilinea aerophila	0.004892
Coprococcus catus	0.004863
Fibrobacter succinogenes	0.00485
Sphaerochaeta globosa	0.004656

[Ruminococcus] torques	0.004506
Slackia heliotrinireducens	0.00447
Megamonas hypermegale	0.004444
Clostridium ljungdahlii	0.004427
Clostridium acetobutylicum	0.004251
Desulfovibrio vulgaris	0.003889
Fretibacterium fastidiosum	0.003739
Ruminococcus bromii	0.003736
Oscillibacter valericigenes	0.003693
Lachnoclostridium phytofermentans	0.003615
butyrate-producing bacterium SS3/4	0.003595
Prevotella sp. oral taxon 299	0.003478
Propionibacterium freudenreichii	0.003268
Clostridium beijerinckii	0.003136
Elusimicrobium minutum	0.003126
[Clostridium] saccharolyticum	0.003079
Cryptobacterium curtum	0.003068
Lactobacillus johnsonii	0.003066
Achromobacter xylosoxidans	0.003026
Clostridium tetani	0.003025
Dehalococcoides mccartyi	0.002996
Enterococcus faecium	0.002933
Paenibacillus mucilaginosus	0.002785
[Clostridium] stercorarium	0.002769
Butyrivibrio proteoclasticus	0.002719
Desulfovibrio desulfuricans	0.002676
Desulfomicrobium baculatum	0.002648
Ruminococcus sp. SR1/5	0.002645
Coprococcus sp. ART55/1	0.002637
Pseudomonas resinovorans	0.002628
Bifidobacterium bifidum	0.002611
Roseburia hominis	0.002605
Ruminococcus albus	0.002589
Streptococcus suis	0.002549
Enterococcus faecalis	0.002542
Butyrivibrio fibrisolvens	0.002511
Clostridium sp. SY8519	0.002491
Denitrovibrio acetiphilus	0.002481
Olsenella uli	0.002479
[Ruminococcus] obeum	0.002401
Ruminococcus champanellensis	0.002399
Coriobacterium glomerans	0.002363
Clostridium saccharoperbutylacetonicum	0.002341

Streptococcus pyogenes	0.002292
[Eubacterium] cylindroides	0.002249
butyrate-producing bacterium SSC/2	0.002242
Desulfitobacterium hafniense	0.002234
Acidaminococcus fermentans	0.002169
Adlercreutzia equolifaciens	0.002151
Clostridium perfringens	0.002133
Ethanoligenens harbinense	0.002119
Erysipelothrix rhusiopathiae	0.002103
Lactococcus lactis	0.002045
Veillonella parvula	0.002025
Clostridium pasteurianum	0.001978
Treponema pedis	0.001961
Gordonibacter pamelaeae	0.001899
Syntrophobotulus glycolicus	0.001881
Desulfovibrio magneticus	0.001877
Streptococcus pasteurianus	0.001848
Symbiobacterium thermophilum	0.001819
Treponema denticola	0.001812
Selenomonas ruminantium	0.001798
Paenibacillus sp. Y412MC10	0.001752
Selenomonas sputigena	0.001744
Finegoldia magna	0.001734
Clostridium saccharobutylicum	0.001725
Spirochaeta thermophila	0.001706
Eggerthella lenta	0.001704
Halobacteroides halobius	0.001687
Desulfitobacterium dichloroeliminans	0.001648
Syntrophomonas wolfei	0.00162
Treponema primitia	0.00158
Mahella australiensis	0.001575
Candidatus Saccharimonas aalborgensis	0.001567
Bifidobacterium thermophilum	0.001567
Desulfosporosinus orientis	0.001552
Prevotella denticola	0.001495
Cellulosilyticum lentocellum	0.00137
Acidaminococcus intestini	0.00135
Tannerella forsythia	0.001348
Desulfotomaculum ruminis	0.001335
Janthinobacterium sp. Marseille	0.001327
Clostridium sp. BNL1100	0.00132
Paenibacillus sp. JDR-2	0.001259
Treponema brennaborense	0.001236

Eubacterium limosum	0.0012
Herminiimonas arsenicoxydans	0.001179
Lactobacillus delbrueckii	0.001174
Spirochaeta smaragdinae	0.001159
[Clostridium] clariflavum	0.00115
Prevotella intermedia	0.001067
Thermobacillus composti	0.001063
Opitutus terrae	0.001054
Methanobrevibacter smithii	0.00105
Eggerthella sp. YY7918	0.000962
Bifidobacterium animalis	0.000962
Prevotella dentalis	0.000954
Sphaerochaeta pleomorpha	0.000918
Thermacetogenium phaeum	0.000901
Thermaerobacter marianensis	0.00087
Gardnerella vaginalis	0.000861
Actinobacillus succinogenes	0.00085
Ruminiclostridium thermocellum	0.000811
Alkaliphilus metalliredigens	0.000808
Haemophilus parasuis	0.000807
Treponema azotonutricium	0.0008
[Clostridium] cellulolyticum	0.000792
Escherichia coli	0.000786
Mannheimia haemolytica	0.00078
Actinobacillus pleuropneumoniae	0.000756
Acetobacterium woodii	0.000733
Megasphaera elsdenii	0.000698
Desulfotomaculum gibsoniae	0.000683
Desulfomonile tiedjei	0.000649
Porphyromonas gingivalis	0.000627
Prevotella melaninogenica	0.000608
Bacillus coagulans	0.000529
Candidatus Methanomassiliicoccus intestinalis	0.000516
Desulfosporosinus meridiei	0.000482
Thermanaerovibrio acidaminovorans	0.000477
Streptococcus intermedius	0.000476
[Clostridium] sticklandii	0.000467
Desulfotomaculum acetoxidans	0.000463
Thermoanaerobacterium thermosaccharolyticum	0.00043
Bifidobacterium dentium	0.000396
Desulfotomaculum kuznetsovii	0.000387
Porphyromonas asaccharolytica	0.00038
Alkaliphilus oremlandii	0.000378

Heliobacterium modesticaldum	0.000352
Bifidobacterium adolescentis	0.000308
Bacteroidales bacterium CF	0.000305
Campylobacter hominis	0.000272
Clostridium kluyveri	0.000228
Riemerella anatipestifer	0.000165
Weeksella virosa	0

Supplementary Table 25: Abundance estimates for the bacteria in the gut metagenome.

	Nanoscope	Reference
Acinetobacter baumannii	0.004316	0.003776
Actinomyces odontolyticus	0	0.000143
Bacillus cereus	0.00222	0.00192
Bacteroides vulgatus	0.00018	0.000158
Clostridium beijerinckii	0.008438	0.007205
Deinococcus radiodurans	0	0.002793
Enterococcus faecalis	0	0.000063
Escherichia coli	0.05176	0.060983
Helicobacter pylori	0.002308	0.002017
Lactobacillus gasseri	0.000135	0.000134
Listeria monocytogenes	0.00099	0.001104
Neisseria meningitidis	0.002127	0.003039
Propionibacterium acnes	0.003828	0.003204
Pseudomonas aeruginosa	0.016951	0.014213
Rhodobacter sphaeroides	0.23266	0.189467
Staphylococcus aureus	0.287555	0.25594
Staphylococcus epidermidis	0.22212	0.235984
Streptococcus agalactiae	0.008532	0.0078
Streptococcus mutans	0.155881	0.136271
Streptococcus pneumoniae	0	0.000206
Correlation		0.97509635

Supplementary Table 26: Abundance estimates for the bacteria in the mock metagenome obtained from Nanoscope, compared to estimates obtained by mapping short reads to the known 20 reference genomes.

	Mock SNPs	Mock indels	Gut SNPs	Gut indels
Number of distinct alleles across haplotypes	3940	121	464652	5328
In regions with > 20X short reads cov.	3880	82	141027	1925
That are also confirmed by short reads	3791	82	136072	1895
Concordance	97.70%	100%	96.49%	98.44%

Supplementary Table 27: Validation of bacterial haplotypes using shotgun sequencing. Shotgun reads were aligned to fasta sequences corresponding to each bacterial haplotype. We say that a variant within a haplotype is supported by a short read if the reads aligns perfectly to the variant. Because short and long reads have different coverage profiles, we only confirm variants that fall in regions with >20X short read coverage.

	Joint (staggered sample)	SMRT (even sample)
# contigs (>= 0 bp)	3092	1121
# contigs (>= 1000 bp)	1668	1121
Total length (>= 0 bp)	46410922	52622925
Total length (>= 1000 bp)	45482193	52622925
# contigs	3092	1121
Largest contig	1687331	2954570
Total length	46410922	52622925
Reference length	83861393	83861393
GC (%)	46.67	46.55
Reference GC (%)	43.64	43.64
N50	91895	142836
N75	25235	47598
L50	102	60
L75	364	225
# misassemblies	121	105
# misassembled contigs	95	64
Misassembled contigs length	6541944	8100339
# local misassemblies	651	66
# unaligned contigs	290 + 58 part	9 + 3 part
Unaligned length	265150	126875
Genome fraction (%)	52.986	60.466
Duplication ratio	1.041	1.036
# N's per 100 kbp	55.5	0
# mismatches per 100 kbp	7.9	3.08
# indels per 100 kbp	2.69	14.11
Largest alignment	1662529	2954569
NA50	83121	126638
NA75	23510	40835
LA50	107	66
LA75	383	245

Supplementary Table 28: Assembly of the mock metagenome using our joint long+short read sequencing strategy, compared to an assembly of SMRT reads using PBCR with MHAP. The SMRT reads were generated for the even mock metagenomic sample at a uniform depth of 70X, while our reads were generated for the staggered sample. SMRT reads generate long contigs, but have a very high indel error rate.

	Mock SNPs	SMRT SNPs
Number of distinct alleles across haplotypes	760	57469
In regions with > 20X short reads cov.	709	20258
That are also confirmed by short reads	708	17915
Distinct variant alleles across haplotypes	295	7689
In regions with > 20X short reads cov.	272	2663
That are also confirmed by short reads	271	389
Concordance over all alleles	99.86%	88.43%
Concordance over variant alleles	99.63%	14.61%

Supplementary Table 29: Validation of haplotypes obtained from SMRT reads using shotgun sequencing and comparison to the LR validation. Shotgun reads were aligned to fasta sequences corresponding to each bacterial haplotype. We say that a variant within a haplotype is supported by a short read if the reads aligns perfectly to the variant. Because various reads have different coverage profiles, we only confirm variants that fall in regions with >20X of short read coverage. We also only look at SNVs because of the SMRT reads' high error rate. Overall, the vast majority of variants identified by SMRT are not confirmed by short reads.

	Ass. merging	**SPAdes**
# contigs (>= 0 bp)	3092	397
# contigs (>= 1000 bp)	1668	254
Total length (>= 0 bp)	46410922	28875392
Total length (>= 1000 bp)	45482193	28839102
# contigs	3092	265
Largest contig	1687331	1382885
Total length	46410922	28846434
Reference length	83861393	83861393
GC (%)	46.67	45.38
Reference GC (%)	43.64	43.64
N50	91895	252756
N75	25235	91715
L50	102	27
L75	364	75
# misassemblies	121	71
# misassembled contigs	95	53
Misassembled contigs length	6541944	8282145
# local misassemblies	651	66
# unaligned contigs	290 + 58 part	5 + 0 part
Unaligned length	265150	22316
Genome fraction (%)	52.986	34.385
Duplication ratio	1.041	1.001
# N's per 100 kbp	55.5	0
# mismatches per 100 kbp	7.9	9.46
# indels per 100 kbp	2.69	1.85
# genes	27224 + 2485 part	12834 + 264 part
# operons	4567 + 1079 part	2120 + 163 part
Largest alignment	1662529	1131235
NA50	83121	201356
NA75	23510	80326
LA50	107	32
LA75	383	90

Supplementary Table 30: Comparison of the assembly merging strategy to assembly short and long reads together with SPAdes on the mock metagenomic sample. SPAdes achieves much lower error and higher N50, but assembles much fewer base pairs as well as 50% fewer genes and operons.

	Joint	SPAdes
# contigs	34786	14709
Largest contig	3936002	1046061
Total length	656202352	268877838
GC (%)	46.85	46.33
N50	49208	73993
N75	18127	38564
L50	2367	989
L75	8301	2262
# N's per 100 kbp	116.82	0
# predicted genes (unique)	552680	247095
# predicted genes (>= 0 bp)	623203	247761
# predicted genes (>= 300 bp)	533061	215917
# predicted genes (>= 1500 bp)	90978	39183
# predicted genes (>= 3000 bp)	11163	5198

Supplementary Table 31: Comparison of the assembly merging strategy to assembly short and long reads together with SPAdes on the real metagenomic sample. SPAdes achieves much lower error and higher N50, but assembles much fewer base pairs and the contigs that it produces contain 50% fewer predicted gene ORFs.

	SPAdes	Asm. Merging
Number of conitgs	2354	5040
Contigs N50	23847	13452
Longest contig	471336	197804
Total bp assembled	44152518	52147096
Genes predicted	41858	49892
Number of variants	18085	10534
Number of bacterial haplotypes	94	50

Supplementary Table 32: Metagenomic analysis of the 4m soil metagenome from Sharon et al. We ran the Nanoscope pipeline on data downloaded from SRA using both the standard assembly strategy (Asm. Merging) and using the optional SPAdes assembler. SPAdes assembled about 85% as much sequence as the merging approach into contigs that were much longer. More variants and haplotypes were found in the SPAdes assembly. Both methods over an improvement over the results of Sharon et al., which assembled sequence into contigs of less than <10 kbp.

Phylum	# contigs	% contigs	# bp	% bp
unclassified	1075	0.456669499	16128086	0.365281228
Proteobacteria	643	0.273152082	15314326	0.346850569
Chloroflexi	221	0.093882753	4341316	0.098325445
Nitrospirae	83	0.035259133	2384163	0.053998347
Firmicutes	79	0.033559898	1748190	0.039594344
Bacteroidetes	50	0.021240442	970529	0.021981283
Euryarchaeota	50	0.021240442	767823	0.017390243
Acidobacteria	23	0.009770603	502788	0.011387527
Actinobacteria	27	0.011469839	293005	0.006636201
Deinococcus-Thermus	8	0.003398471	168542	0.003817268
Crenarchaeota	9	0.00382328	157085	0.003557781
Cyanobacteria	14	0.005947324	147908	0.003349934
Chlorobi	6	0.002548853	131811	0.002985356
Spirochaetes	7	0.002973662	129599	0.002935257
Planctomycetes	9	0.00382328	112873	0.002556434
Thermodesulfobacteria	5	0.002124044	108882	0.002466043
Ignavibacteriae	5	0.002124044	78134	0.001769639

Supplementary Table 33: Top phyla identified in the 4m soil metagenome from Sharon et al. We ran the Nanoscope pipeline on data downloaded from SRA using both the standard assembly strategy (Asm. Merging) and using the optional SPAdes assembler. The FCP program identified 16 phyla to which >75 Kbp of sequence could be mapped. These include all the standard phyla reported by Sharon et al. (i.e. we choose not to consider candidate phyla); however, we also find new phyla, such as Firmicutes. This is the 4-th most abundant phylum, but it is not reported by Sharon et al.

	Our method	**Nielsen et al.**	**Albertsen et al.**	**Iverson et al.**
Sample type	Gut microbiome	Gut microbiome	Environmental	Environmental
# of samples	1	18-396	2	1
Seq. platform	Tru-seq SLR	Illumina WGS	Illumina WGS	SOLiD mate-pairs Sanger sequencing
Seq. amount	8 Gbp (long reads) x40 (subassembly)	4.5 Gbp/sample	86 Gbp	59 Gbp
Software	Nanoscope, Lens	Canopy clustering	Multi-metagenome	SEAStAR
Analysis type	De-novo assembly; Phasing	Clustering across subjects; assembly	Binning across extraction methods	De-novo assembly; Nucleotide binning
Softw. availability	Yes	Yes	Yes	No longer online
Taxa detected	178	741	31	47
Resolution	Individual SNV	Strain	Species with diff. GC content	Family
Longest scaffold	3.9 Mbp	733 Kbp	3.6 Mbp	2.2 Mbp
Scaffold N50	49 Kbp	39 Kbp[1]	4.1 Kbp overall ~100 Kbp for top species	6.8 Kbp
Bases assembled	656 Mbp	45 Mbp (genes) 35 Gbp (total)	423 Mbp	300 Mbp
# Variants	200K	n/a	n/a	n/a
# Haplotypes	5K	n/a	n/a	n/a

Supplementary Table 34: Comparison of Tru-seq synthetic long reads to alternative metagenomic analysis techniques. In brief, our approach produces similar results to alternative techniques that used hundreds of pooled samples (Nielsen et al.) or potentially inaccurate binning approaches (Albertsen et al., Iverson et al.). Furthermore, we analyze strains at a higher resolution than before by detecting strain-specific SNVs and indels and phasing them into haplotypes.

[1] Note that the contigs of Nielsen et al. are also clustered into unordered sets belonging to the same species.