Computational haplotype recovery and long-read validation identifies novel isoforms of industrially relevant enzymes from natural microbial communities

Samuel M. Nicholls1,2,3, Wayne Aubrey1, Arwyn Edwards3, Kurt de Grave2,4, Sharon Huws3,5, Leander Schietgat2, André Soares3, Christopher J. Creevey3,* and Amanda Clare1,*

Abstract

Population-level diversity of natural microbiomes represent a biotechnological resource for biomining, biorefining and synthetic biology but requires the recovery of the exact DNA sequence (or “haplotype”) of the genes and genomes of every individual present. Computational haplotype reconstruction is extremely difficult, complicated by environmental sequencing data (metagenomics). Current approaches cannot choose between alternative haplotype reconstructions and fail to provide biological evidence of correct predictions. To overcome this, we present Hansel and Gretel: a novel probabilistic framework that reconstructs the most likely haplotypes from complex microbiomes, is robust to sequencing error and uses all available evidence from aligned reads, without altering or discarding observed variation. We provide the first formalisation of this problem and propose “metahaplome” as a definition for the set of haplotypes for any genomic region of interest within a metagenomic dataset. Finally, we demonstrate using long-read sequencing, biological evidence of novel haplotypes of industrially important enzymes computationally predicted from a natural microbiome.

Keywords: ‘metagenome’, ‘haplotypes’, ‘long read sequencing’, ‘algorithm’
Running Title: Haplotype recovery from natural microbiomes
Contact: msn@aber.ac.uk (+441970 622 424)
It has become clear that population-level genetic variation drives competitiveness and niche specialization in microbial communities [1]. Novel combinations of variants in individuals (haplotypes) are filtered by natural selection so that those that confer an advantage are retained [2]. Recovering the haplotypes of enzyme isoforms for a given gene across all organisms in a microbiome (the “metahaplome”) would offer great biotechnological potential [3, 4] and allow unprecedented insights into microbial ecosystems [5].

Similar goals in humans are being achieved by the International HapMap Project which aims to describe the common patterns of human genetic variation that affect health, disease, responses to drugs and environmental factors [6]. However, microbial research has so far focused on higher-level characterizations of diversity, for example: the gene-set of all strains of a species (the pan-genome) [7], or quantification of individual SNPs found in microbial communities (variome) [8] or in viruses, the strains related by mutations in a highly mutagenic environment (the quasispecies) [9].

Reconstructing population-level variation in microbial communities is limited by our inability to culture in vitro many microbes from the environment. Researchers must instead rely on DNA isolated and sequenced directly from an environment (metagenomics) which generally results in highly fragmented and incomplete data containing sequencing errors. This complicates the already computationally difficult (NP-hard) [10] problem of haplotyping [11]. The generation of haplotypes from metagenomes is particularly difficult as existing de novo analysis pipelines for DNA sequence data generally assume a single individual of origin and, when applied to metagenomic datasets, remove low level variation and produce single consensus sequences [12]. Furthermore, naive sequence partitioning approaches such as contiguous clustering or clustering cannot sufficiently distinguish strains or require many samples [13]. Even specialized metagenomic assemblers [12, 14, 15] do not aim to solve the problem of recovering haplotypes.

To make the generation of approximate solutions both computationally tractable and accurate, focus has shifted towards the use of heuristics [16, 17, 18, 19, 20]. However, recent approaches for analogous problems typically produce a superset of many possible haplotypes and leave it to the user to choose the best candidates.

Additionally, the problem of recovering haplotypes from a metagenome has been left without a formal mathematical definition and methodologies are limited to diploid species or for those with well-defined genomes [21, 22]. Whilst researchers have identified the problem that consensus assembly poses for the downstream analysis of variants [23] and are moving towards alternative assembly approaches, such as graph-based assembly [24, 25], there are still no biologically validated methods for the recovery of individual haplotypes for regions of a metagenome.

We hypothesise that a probabilistic framework could identify the true haplotype diversity of industrially important enzymes in a microbiome and allow ranking and selection of the most likely haplotypes for further investigation. To test this, we have developed Hansel and Gretel, a Bayesian framework capable of recovering and ranking haplotypes using evidence of pairs of single nucleotide polymorphisms (SNPs) observed on sequenced reads. While specifically designed to extract haplotypes from metagenomic data of microbial communities, we show that the algorithm is general enough to be applied to analogous haplotyping problems.

We characterize the performance of our approach on simulated metagenomes, demonstrate its effectiveness on data from a highly complex natural microbial community and validate these results, using Sanger, Illumina and Nanopore sequencing. We demonstrate how, for the first time, the most likely haplotypes can be recovered with high fidelity from complex metagenomic samples, enabling the characterisation of the true population-level diversity of genes in microbiomes.

Results

The metahaplome

We provide the first formalisation of the problem of recovering haplotypes from a metagenome, and define the metahaplome as the set of haplotypes for any particular genomic region of interest within a metagenomic data set. The full mathematical definition is available in Supplementary Section 1.

Hansel and Gretel

We have developed Hansel, a data structure designed to efficiently store variation observed across sequenced reads, and Gretel, an algorithm that leverages Hansel for the recovery of haplotypes from a metagenome. Advantages include that our method:
• recovers haplotypes from metagenomic data
• does not need a priori knowledge of the number of haplotypes
• makes no assumptions about the distribution of alleles at any variant site
• does not need to distinguish between sequence error and variation
• uses all available evidence provided by the raw reads
• does not require any user-defined parameters
• does not require bootstrapping, model building or pre-processing
• can confidently rank its own results based on calculated likelihoods
• can be executed on an ordinary computer
• has been verified in vitro

The details of the data structure and algorithm are provided in the Online Methods. We provide open source implementations for the data structure API (Hansel) and the haplotype recovery algorithm (Gretel) at https://github.com/samstudio8/gretel.

We show in silico that recovery of haplotypes from metahaplomes is possible even with data sets consisting of short reads, and verify in vitro that our computational predictions identify novel isoforms of enzymes found in a natural microbial community. The following subsections describe the evaluation of Hansel and Gretel on the following data:
• synthetic metahaplomes,
• a metagenomic mock community from Quince et al. [26],
• enzymes from a real microbiome, validated using Oxford Nanopore long-read sequencing.
Figure 1: Boxplots summarising the proportion of variants on an input haplotype correctly recovered (y-axes) from groups of synthetic metahaplomes by Gretel. Single boxplots present recoveries from a set of five metahaplomes generated with some per-haplotype mutation rate (column facets), over 10 different synthetic read sets with varying read length (row facets) and per-haplotype read depth (colour fill). Each box-with-whiskers summarises the proportion of correctly recovered variants over the 250 best recovered haplotypes (yielded from 50 Gretel runs (5 metahaplome replicates × 10 read sets), each returning 5 best outputs). We demonstrate better haplotype recoveries can be achieved with longer reads and more dense coverage, as well as the limitations of recovery on data exhibiting fewer SNPs/hb. This figure may be used as a naive lookup table to assess potential recovery rates for one’s own data by estimating the level of variation, with the average read length and per-haplotype depth.

Synthetic metahaplomes

We evaluated the fidelity of the haplotype reconstructions from Hansel and Gretel using synthetic metahaplomes. Each synthetic metahaplome consisted of five 3000 bp haplotypes generated by simulated evolution using seq-gen [27], with a fixed mutation rate and a star phylogeny (see Methods). Five replicates of seven different mutation rates were generated for a total of 35 metahaplomes.

For each of the 35 metahaplomes simulated by seq-gen, we generated 180 sets of uniformly distributed pseudo-reads consisting of 10 replicates for each pairing of 3 read sizes and 6 per-haplotype depths. For the purpose of read alignment and variant calling, we aligned each read set against the 3000 nt starting sequence initially provided to seq-gen. Variants were called by assuming any heterogeneous genomic position over the aligned reads was a SNP.

A single run of Gretel will repeatedly recover haplotypes until the stopping criteria specified is met (see Methods). For each synthetic metahaplome replicate, we evaluated the fidelity of haplotypes reconstructed by Gretel though comparison with the input sequences used to generate the data. The reconstructed haplotype sequence with the greatest proportion of heterozygous positions in agreement with each of the original simulated sequences were determined. We present this recovery rate for the seven mutation rates in combination with the 3 read sizes and 6 per-haplotype read depths used (Figure 1).

We found that haplotype recovery improves with longer reads and greater coverage. We also observed potential lower bounds on our ability to recover haplotypes from a data set, as the facets with no successful recoveries show. Unsuccessful recoveries are a result of at least one pair of adjacent variants failing to be covered by any read, which is a requirement imposed on Gretel for recovery (see Methods). For shorter reads, low-level variation is more of a problem. 0.01 SNPs per haplotype base (hb) over 100 bp would yield just one SNP on average - insufficient evidence for Gretel.

Although one might expect high levels of variation to make the recovery of haplotypes more challenging, an abundance of variation actually provides more information for Gretel. We observe successful recoveries from data sets with high variation (0.1 SNPs/hb over five haplotypes of 3000 nt yields ≈ 1500 SNPs [Table 1]). With enough coverage (≥ 7x per-haplotype depth), recoveries at a high level of variation are more accurate than those in data sets with fewer SNPs.

For realistic levels of variation (0.01–0.02 SNPs/hb) [8], with per-haplotype read depth of ≥ 7x, we can recover haplotypes at a median accuracy of 80%. With higher per-haplotype depth (≥ 25x), Gretel is capable of recovering haplotypes with 100% accuracy (Figure 1).

Metahaplomes from a mock community

To show our method is capable at handling metagenomic data at scale, we used a mock community from Quince et al. (2017) [26]. A mock community is necessary as there
are currently no metagenomes that have been annotated with known haplotypes [24]. The community contains 5 *Escherichia coli* strains, and 15 other genomes commonly found in the human gut according to samples from the Human Microbiome Project [29]. The authors made available 16 million synthetic read pairs, generated from the 20 genomes to simulate a “typical HiSeq 2500 run” [26]. Additionally the original authors identified 982 single-copy core species genes (SCSGs) for *E. coli* and provided DNA sequences of all SCSGs for the five *E. coli* strains in the community.

We assembled the synthetic reads with MEGAHIT [30], and we could map 814 of the 982 SCSGs back to our assembly (see Methods). We executed *Gretel* over the 814 mapped SCSG sites with the aim to recover the haplotype for each of the five *E. coli* strains. The results are shown in Figure 2. *Gretel* is capable of achieving results with comparable accuracy to the current state-of-the-art for the related problem of strain deconvolution (DESMAN [20]). The binning step of the DESMAN pipeline led to a majority of the SCSGs being discarded, leaving only 372 (of 982) for their own analysis. Whereas DESMAN requires significant pre-processing, we show it is possible to achieve accurate haplotype recovery (over more sites) without the need to perform any pre-processing. We show that *Gretel* is capable of scaling to recover strain-specific haplotypes from a microbial community, for hundreds of highly variable *E. coli* genes with an average accuracy of 99.5%.

### Recovery from a real metahaplome

Finally, to validate our method empirically, we predicted haplotypes from a natural microbial community, using short-reads, and verified their existence by sequencing isolated amplicons with Nanopore long-read technology.

As part of a previous experiment on the colonisation of grass in the rumen [31], samples of rumen metatranscriptome were obtained from a 3 cows over a series of timepoints after feeding (see Methods). 118M read-pairs were generated with an Illumina HiSeq 2500 (100 bp). Reads were partitioned with *kmer* and assembled with *Velvet* to generate an assembly which served as a pseudo-reference. The previous study assembled the annotation using *MGKit* [32] with the *Uniprot* database.

To find isoforms of industrially relevant enzymes we filtered annotations to classes of hydrolase (Enzyme Classification (EC): 3.2, 3.4 and 5.3) known to be found in the rumen [1]. As a proof of concept, a subset of 259 regions were selected by criteria including length and distribution of variants over aligned reads (see Methods). *Gretel* was executed at each of the 259 sites to recover haplotypes. Forward and reverse primers were generated with *pds* [33] using the recovered haplotypes as template sequence. For laboratory analysis, 10 regions were hand-selected according to criteria including number of recovered haplotypes, gene length and to satisfy primer design constraints.

For each of the ten regions, a cDNA library was produced via gene-specific reverse transcription of the pooled RNA samples. Amplicons were isolated via high-fidelity PCR and extraction following gel electrophoresis. Five of the ten samples (Table 4) could be isolated from the original cDNA in sufficient amount for use with the protocol. Extracted DNA for the five successful PCR reactions was used as template for another round of high-fidelity PCR. DNA was isolated from excised gel bands, pooled and suspended with AMPure beads. Isolated DNA was verified via Sanger sequencing. Sequencing library preparation was performed with the Oxford Nanopore LSK108 ligation kit.

The library was loaded on an Oxford Nanopore MinION. Sequencing generated 634,859 reads that passed quality control (Albacore v2.0.2) in 1hr 28m. Nanopore sequences were aligned against the corresponding pseudo-references for the five targeted genes. The identity of each long-read (discarding indels) versus each *Gretel* predicted haplotype was calculated for the five genes. Supplementary Figure 4 shows the distribution of *phred* scores across reads. The mean score of 10.53 corresponds to an error rate of 8.85%. Despite this, we were able to identify individual molecules with extremely high identity to recovered haplotypes. The haplotype with the best likelihood for *Gretel* G123 (*Exoglucanase XynX*) region had an identity of 99.7%.

Figure 3 depicts, for G123, a comparison between several of the highest likelihood *Gretel* recovered haplotypes, and their associated highest identity sequenced DNA molecules. We show that *Gretel* has predicted novel isoforms of an exoglucanase enzyme, with potential biotechnical applications. Figures for the remaining genes can be found in Supplementary Section 10.2.

We show for the first time with *in vitro* evidence that a computational method is capable of recovering sequences of co-occurring variants that actually exist in nature, with high accuracy, from short read data.

---

**Figure 2** The boxplot summarises the percentage sequence identity (y-axis) of *Gretel* haplotypes recovered from each of the 814 gene sites, to five *E. coli* strains (column facets) known to exist in the mock community. *Gretel* was executed at 814 sites on an assembled mock metagenome, consisting of short-reads generated from five *E. coli* strain haplotypes, and 15 other genomes. The y-axis is truncated at 90%.
Figure 3: Comparison of our recovered haplotypes against Oxford Nanopore long-read data for Gretel G123 (Exoglucanase XynX). Outermost ring represents the metagenomic assembly (pseudo-reference). Grey banding represents coverage of original Illumina read data. Line plot depicts Sanger sequencing chromatogram for G123 PCR amplicon. Pairs of tracks toward the center align a DNA molecule sequenced by Oxford Nanopore MinION (outer, coloured) to a specific haplotype recovered by Gretel (inner). The haplotypes are masked (black) at sites homozygous over the displayed haplotypes to ease comparison of predicted variants. Heterozygous sites on the haplotypes are supported by Sanger sequencing peaks (e.g. 347, 1238), and co-occurring variants are supported by the Nanopore reads. In several positions (e.g. 347, 407) Gretel can be observed to correct the reference. Gretel can recover enzyme isoforms from a natural microbiome.

Discussion

Comparison to related work

In contrast to other methods, Gretel aims to make as few assumptions as possible. More importantly, our framework requires no configuration, has no parameters, requires no pre-processing of reads, does not discard observed information and is designed for metagenomic data sets where the number of haplotypes is unknown. Existing methods have one or more limitations which make them unsuitable for metagenomic analysis:

- they assume that the solution is a pair of haplotypes from diploid parents, and discard/alter observations until a pair of haplotypes can be determined [17, 34]
- they discard SNP sites that feature three or more alleles as errors [34]
- they can generate a unrealistically large number of unordered potential haplotypes [1, 35]
- they are too computationally expensive for high-depth short read data sets [36]
- they require a good quality reference genome [37]
- they are no longer maintained/are specific to certain data/cannot be installed [38]

It is important to note that no other tool that claims to recover haplotypes or strains from a microbial population has attempted to validate their work biologically.

More recent advances in the recovery of sequences from mixed populations are limited to ConStrains, SAVAGE, and DESMAN. ConStrains [28] aims to resolve strain-level differences within a set of metagenomic samples. It first uses MetaPhlAn to provide a species composition profile, and then chooses a corresponding set of core gene markers against which to align reads. The
frequencies of SNPs in this alignment are used to cluster haplotypes of an enzyme or gene of interest, but instead can track strains in samples by their profiles of variation over a particular set of marker genes.

**SAVAGE** [29] is designed specifically for the related problem of viral quasi-species recovery [10], with favourable comparison to the state-of-the-art for viral genomes. However, we note that the tool recovers many unordered haplotypes (>800 haplotypes for a lab mixture that contained just 5 strains of HIV). Additionally, as it uses an overlap assembly approach it is not particularly suited to the complexity of metagenomes. Overlap assembly approaches such as SAVAGE and Lens [4] create large numbers of potential haplotypes by naively branching at choices without long range information.

**Gretel** overcomes this by outputting each recovered haplotype with a likelihood, given the observed read data. Haplotypes can be ordered and filtered, and likelihoods are amenable to further statistical tests.

We evaluated **Gretel** on the same HIV data that SAVAGE reported in order to demonstrate that our method can also resolve highly variable viral quasi-species genomes. **Gretel** makes almost perfect recoveries from this sequenced laboratory mix of five strains. Our results are presented in Supplementary Section [17].

**DESMAN** [26] is a complex bioinformatics pipeline, that relies on read-binning, availability of good references, and a database of single-copy core species genes (SCSGs) with which to perform clustering. DESMAN then uses SNP frequencies to determine haplotypes. The use of frequencies observed across samples means that they only addressed single copy genes, as multiple copies would distort the frequencies. Furthermore, it makes use of binning software such as CONCOCT in order to filter reads before alignment to the SCSGs, and this binning process requires data from many samples (> 50 preferred). We were unable to run DESMAN on our synthetic data, which represents the scenario of analyzing genes that are not SCSGs, with diversity present in a single microbial sample. However, Figure 2 shows we were able to make excellent recoveries on the live *E. coli* haplotypes for 814 SCSGs (of the 982 provided) for their mock community, far more sites than DESMAN achieved.

Recovering the variation observed at the gene isoform level in a sample of a microbial community is a different problem to that of strain tracking, species binning or read clustering. **Gretel** provides the first practical solution to this important problem and at the same time performs as well or better than SAVAGE and DESMAN, on evaluation using their own benchmark data.

**Performance and tractability**

Our approach is influenced by the availability and quality of read alignments against the pseudo-reference, and the choice of pseudo-reference itself. It should be noted that the pseudo-reference is not used by Hansel or Gretel, it serves only as a common sequence against which to align raw reads. Sequences that happen to share identity with the pseudo-reference are recovered by **Gretel** from the evidence in the Hansel matrix, the reference confers no advantage over any other haplotype. Very high recovery rates on sequences that share identity with the pseudo-reference are a reflection of the strength of our approach, and not a trivial recovery.

Ultimately, the tractability of the problem is bound by the quality of the data available: both assemblers and aligners will exert influence over how many and how accurately haplotypes in a given metahaplome can be recovered. As stated by Lancia [11], it is entirely possible that, even without error, there are scenarios where data is insufficient to successfully recover haplotypes and the problem is rendered impossible.

Our framework has been designed for the recovery of haplotypes from a region of interest in a metagenome (such as variants of a gene involved in a catalytic reaction of interest, e.g. degradation of biomass), but given sufficient coverage of SNPs, our approach could work on regions significantly longer than that of a gene if desired and with data consisting of significantly longer reads.

Regarding time and resource requirements, **Gretel** is designed to work on all reads from a metagenome that align to some region of interest on the pseudo-reference. Typically these subsets are small (on the order of 10-100K reads) and so our framework can be run on an ordinary desktop in minutes, without significant demands on disk, memory or CPU. Run-times on data with very deep coverage, or many thousands of SNPs, such as the HIV 5-mix, run on the order of hours, but can still be executed on an ordinary desktop computer.

**Future work**

Although we demonstrate **Gretel**’s capability to recover haplotypes from a natural microbiome, there exists room for further work. We intend to revisit the following aspects of our approach:

- **Reweighting**
  The pairwise SNP observations that contributed to the most recent haplotype are reweighted in the Hansel matrix to permit new paths to be discovered. A balance must be satisfied to prevent haplotype skipping or duplication. We are experimenting with alternative reweighting schemes.

- **Naive insertion handling**
  Due to a size constraint on the Hansel matrix, further thought is needed to devise a practical methodology that permits proper consideration of insertions. However, unlike many other approaches **Gretel** does not discard reads containing insertions.

- **Greedy Search**
  We assume the “best” haplotype is the most likely haplotype, and that it can be recovered by selecting the edge with the highest probability at each SNP. However it is possible that **Gretel** could locate solutions whose overall likelihood may be better with an alternative search strategy.

- **Stopping Criterion**
  **Gretel** generates haplotypes until a dead end in the Hansel matrix is encountered, from which there is no evidence for any further transitions. Although we found that our approach can yield low-quality haplotypes before this time, they have lower likelihoods.

- **Unused Evidence**
  There remain sources of evidence not currently used by our algorithm — namely paired end reads and alignment base quality scores. Such data will certainly provide useful co-occurrence and confidence information for SNPs that span some known insert, however careful consideration on how to integrate this data to our approach is necessary.
Conclusion

In this work we offer the term *metahaplome* to represent the set of haplotypes for any particular region of interest within a metagenomic data set. The recovery of sequences from individuals within the metahaplome provides a rich resource of information, enabling detailed study of microbial communities. Synthetic exploitation of the variation observed can be used to improve industrial processes such as biorefining, biominning and synthetic biology [11][12].

To exploit this variation, we provided an implementation of *Hansel*: a data structure for the storage and manipulation of evidence of variation observed across reads in a sequenced metagenome. *Hansel* has value outside of this work, and can provide future algorithms a means to interact with the variation observed in a set of sequenced reads. We also provide *Gretel*, an algorithm for the recovery of haplotypes from a metahaplome.

For the first time, our work provides Nanopore and Sanger sequence evidence for the existence of computationally predicted haplotypes from a natural microbial community. We show with in *vitro* evidence that a computational method is capable of recovering isoforms of enzymes from a microbe, given only short-read sequencing. Long-read sequencing identified individual DNA molecules consistent with our predicted haplotypes. However, the error rate observed across our Nanopore sequencing run shows that it is not currently possible to recover haplotypes in a microbiome without error using long-read sequencing alone.

*Hansel* and *Gretel* have the potential to discover novel isoforms of enzymes responsible for catalytic reactions of biotechnological importance. This is not a trivial task; many existing in *silico* and in *vitro* techniques such as rational enzyme design have struggled to achieve this goal. Our work lays the foundation for the recovery of industrially relevant haplotypes from natural microbiomes.

Code Availability and Data Access

Our *Hansel* and *Gretel* framework is freely available, open source software available online at [https://github.com/samstudio8/hansel](https://github.com/samstudio8/hansel) and [https://github.com/samstudio8/gretel](https://github.com/samstudio8/gretel), respectively. The code used to generate metahaplotomes and synthetic reads for both the randomly generated and real-gene haplotypes, and the testing data used to evaluate our methods is also available online via [https://github.com/samstudio8/gretel-test](https://github.com/samstudio8/gretel-test).

Nanopore sequence data are available via ENA study PRJEB23483. RNA metatranscriptome data are available via ENA study PRJNA41919.

Acknowledgments

SMN is funded via the Aberystwyth University Doctoral Career Development Scholarship and the IBERS Doctoral Programme. WA is funded through the Coleg Cymraeg Cenedlaethol Academic Staffing Scheme. AE is funded via the Aberystwyth University Interdisciplinary Centre for Environmental Microbiology. SH acknowledges funding from: Biotechnology and Biological Sciences Research Council (BB/J0013/1 and BBS/E/W/10964A-01), TGAC Capacity and Capability Challenge Programme and the Coleg Cymraeg Cenedlaethol. AS is supported by the Sér Cynru National Research Network for Low Carbon Energy and Environment (NRN-LCEE). CJC was funded by the Biotechnology and Biological Sciences Research Council (BB/B023403) Institute Strategic Programme Grant, Rumen Systems Biology (BB/E/W/10964A01). *In vitro* work was supported by the Aberystwyth University Research Fund (URF12431).

SMN wishes to acknowledge Dr. Francesco Rubino for his prior work on the assembly and annotation of the rumen metatranscriptome data.

Disclosure Declaration

Oxford Nanopore Technologies Ltd (ONT) have covered costs for AS to attend and present at London Calling 2017 and AE to attend and present at Nanopore Community Meeting New York 2016 and 2017. ONT have provided free-of-charge materials for an outreach project by AE. We confirm ONT have had no role in the design, execution or interpretation of the present study. The remaining authors have no conflicts of interest to declare.

Author contributions

SMN, AC, CJC, WA with collaboration from KG and LS discussed and defined the theoretical problem. SMN, CJC, AC and WA chose data and designed *in silico* experiments. SMN wrote the code and documentation and executed experiments. *In vitro* experiments were designed by SMN and WA with collaboration from AE and AS. SH provided rumen metatranscriptome RNA and Illumina sequencing. SMN performed laboratory work under the supervision of WA. AE performed Oxford Nanopore library preparation with assistance of SMN and AS. SMN analyzed and interpreted the results with AC, CJC and WA. All authors contributed to the manuscript.

References


Methods

The metahaplome

We have provided a detailed mathematical definition of the metahaplome in Supplementary Section 1.

To enable recovery of a metahaplome from a metagenome with Gretel we require:

- \( g \), a known DNA region (for example a gene), to be identified by the user
- \( c[i : j] \), the region of contig \( c \) (from an assembly \( C \)) which has been identified as having similarity to \( g \)
- \( A(i, j) \), the alignments of the set of reads \( R \) against the contig region \( c[i : j] \)
- \( S(i, j) \), the genomic positions determined to be SNPs over the region \( c[i : j] \)

A metagenomic assembly (which we refer to as a ‘pseudo-reference’, \( C \)) can be generated by assembling sequenced reads, with an assembler such as Velvet [13]. One may identify a gene of interest \( g \), on a contig \( c \) by similarity search or gene prediction. We refer to gene \( g \) as the target. We want to recover the most likely haplotypes of \( g \) that exist in the metahaplome.

A subset of reads that align to the target region can be determined using a short read alignment tool such as bowtie2 [43]. Reads that fall outside the target of interest (i.e., reads that do not cover any of the genomic positions covered by the target) can be safely discarded: they do not provide relevant evidence to SNPs that appear on the region of interest.

Variation at single nucleotide positions across reads along the target, can then be called with a SNP calling algorithm such as samtools [15] or GATK [46]. To avoid loss of information arising from the diploid bias of the majority of SNP callers [34], our methodology aggressively considers any heterogeneous sites as a SNP.

The combination of aligned reads, and the locations of single nucleotide variation on those reads can be exploited by Hansel and Gretel to recover real haplotypes in the metagenome: the metahaplome.

Hansel: A novel data structure

We present Hansel, a probabilistically-weighted, graph-inspired, novel data structure. Hansel is designed to store the number of observed occurrences of a symbol \( \alpha \) appearing at some position in space or time \( i \), co-occurring with another symbol \( \beta \) at another position in space or time \( j \). For our approach, we use Hansel to store the number of times a SNP \( \alpha \) at the \( i \)’th variant of some contig \( c \), is observed to co-occur (appear on the same read) with a SNP \( \beta \) at the \( j \)’th variant of the same contig. Hansel is a four dimensional matrix whose individual elements \( H(\alpha, \beta, i, j) \) record the number of observations of a co-occurring pair of symbols (\( \alpha_i, \beta_j \)).

Different from the typical SNP matrix

Our representation differs from the typical SNP matrix model [14] that forms the basis of many of the surveyed approaches. Rather than a matrix of columns representing SNPs and rows representing reads, we discard the concept of a read entirely and aggregate the evidence seen across all reads by genomic position.

At first this structure may appear limited, but the data in \( H \) can easily be exploited to build other structures. Consider \( H(\alpha, \beta, i, 1, 2) \) for all symbol pairs \((\alpha, \beta)\). One may enumerate the available transitions from space or time point 1 to point 2. Extending this to consider \( H(\alpha, \beta, i, j, i + 1) \) for all \((\alpha, \beta)\) over \( i \), one can construct a simple graph \( G \) of possible transitions between all symbols. In our setting, \( G \) could represent a graph of transitions observed between SNPs on a genomic sequence, across all reads. Figure 4 shows how the Hansel structure records information about SNP pairs, and shows a simple graph constructed from this information.

![Figure 4: Three corresponding representations. (a) a set of aligned short read sequences, with called variants, (b) the actual Hansel structure where each possible pair of symbols (00, 01, 10, 11) has a matrix storing counts of occurrences of that ordered symbol pair between two genomic positions across all of the aligned reads, (c) a simple graph that can be constructed by considering the evidence provided by adjacent variants. Note this representation ignores evidence from non-adjacent pairs, which is overcome by the dynamic edge weighting of the Hansel data structure’s interface.](image)

Intuitively, one may traverse a path through \( G \) by selecting edges with the highest weight in order to recover a series of symbols that represent an ordered sequence of SNPs that constitute a haplotype in the metahaplome. The weight of an edge between two nodes may be defined as the number of reads that provide direct evidence for that pair of SNP values occurring together.

Different from a graph

Although the analogy to a graph helps us to consider paths through the structure, the available data cannot be fully represented with a graph such as that seen in Figure 4 alone. A graph representation defines a constraint that only considers pairs of adjacent positions \((i, i + 1)\) over \( i \). Edges can only be drawn between adjacent SNPs and their weightings cannot consider the evidence available in \( H \) between non-adjacent
SNP symbols. Without considering information about non-adjacent SNPs, one can traverse \( G \) to create paths (sequences of SNPs) that do not exist in the observed data set, as shown in Figure 5. To prevent construction of such invalid paths and recover genuine paths more accurately, one should consider evidence observed between non-adjacent symbols when determining which edge to traverse next.

![Diagram](image)

Figure 5: Considering only adjacent SNPs, one may create paths for which there was no actual observed evidence. Here, the reads \{0011, 0001, 0100\} do not support either of the results \{0000, 0101\}, but both are valid paths through a graph that permits edges between pairs of adjacent SNPs.

Using information from non-adjacent SNPs, and the path so far

The Hansel structure is designed to store pairwise co-occurrences of all SNPs (not just those that are adjacent), across all reads. We may take advantage of the additional information available in \( H \) and build upon the graph \( G \). Incorporating evidence of non-adjacent SNPs in the formula for edge weights allows decisions during traversal to consider previously visited nodes, as well as merely the current node path, \( i \).

That is, given a node \( i \), the decision to move to a symbol at \( i + 1 \) can be informed not only by observations in the reads covering positions \( (i, i + 1) \), but also \( (i−1, i + 1), (i−2, i + 1) \), and so on. Such a scheme allows for the efficient storage of some of the most pertinent information from the reads, and allows edge weights to dynamically change in response to the path as it has been constructed thus far. Outward edges between \( (i, i + 1) \) that would lead to the construction of a path that does not exist in the data can now be influenced by observations in the reads beyond that of the current node and the next. Our method mitigates the risk of constructing paths which do not truly exist.

The consideration and storage of pairwise SNPs fits well with the Naive Bayes model employed to simplify the potentially expensive calculation of conditional probabilities (Supplementary Section 4).

Although we describe Hansel as “graph-inspired”, allowing edge weights to depend on the current path through \( G \) itself leads to several differences between the Hansel structure and a weighted directed acyclic graph. Whilst these differences are not necessarily disadvantageous, they do change what we can infer about the structure.

A dynamic structure

The structure of the graph is effectively unknown in advance. That is, not only are the weights of the edges not known ahead of traversal (as they depend on that traversal), but the entire layout of nodes and edges is also unknown until the graph is explored (although, arguably this would be true of very large simple graphs too). Indeed, this means it is also unknown whether or not the graph can even be successfully traversed.

Also of note is the fact that the graph is dynamically weighted. The current path represents a memory that affects the availability and weights of outgoing edges at each node. Edge weights are calculated probabilistically during traversal. They depend on the observation of SNP pairs between some number of the already selected nodes in the path, and any potential next node. Supplementary Section 5 provides the equation and intuition for the probabilistic calculation of edge weights.

In exchange for these minor caveats, we have a data structure that permits graph-like traversal that is intrinsic to our problem definition, whilst utilising informative pairwise SNP information collected from observations on raw metagenomic reads. Hansel fuses the advantages of a graph’s simple representation (and its inherent traversability) with the ability to efficiently store pertinent information by considering only pairs of SNPs across all reads.

Gretel: An algorithm for recovering haplotypes from metagenomes

We introduce Gretel, an algorithm designed to interface with the Hansel data structure to recover the most likely haplotypes from a metahaplome. To obtain likely haplotypes, Gretel traverses the probabilistic graph structure provided by Hansel, selecting the most likely SNPs at each possible node (i.e. traversing edges with the greatest probability), given some subset of the most recently selected nodes in the path so far. At each node, an \( L \)-th order Markov chain model is employed to predict which of the possible variants for the next SNP is most likely, given the last \( L \) variants in the current path. Execution of Gretel can be broken into the following steps:

1. Parse the read alignments and retain only the bases that cover SNP sites, discarding any conserved base positions as they provide no haplotype information.
2. Populate the Hansel structure with all pairwise observations from each of the reads.
3. Exploit the Hansel graph API to incrementally recover a path until a variant has been selected at each SNP position:
   - Query for the available transitions from the current position in the graph to the next SNP
   - Calculate the probabilities of each of the potential next variants appearing in the path given the last \( L \) variants
   - Append the most likely variant to the path and traverse the edge
4. Report this path as a haplotype and then remove the information for this path from the data by reweighting observations that contributed to this path. This will allow for new paths to be retrieved next.
5. Repeat (3-4) until the graph can no longer be traversed or an optional additional stopping criterion has been reached.
Greedy path construction

Haplotypes are reconstructed as a path through the Hansel structure, one SNP at a time, linearly, from the beginning of the sequence. At each SNP position, the Hansel structure is queried for the variants that were observed on the raw reads at the next position. Hansel also calculates the conditional probabilities of each of those variants appearing as the next SNP in the sequence, using a Markov chain of order L that makes its predictions given the current state of the observations in the Hansel matrix and the last L selected SNPs. Gretel’s approach is greedy: we only consider the probabilities of the next variant. Our razor is to assume that the best haplotypes are those that can be constructed by selecting the most likely edges at every opportunity.

Reweighting to find multiple haplotypes

Whilst our framework is probabilistic, it is not stochastic. Given the same Hansel structure and operating parameters, Gretel will behave deterministically and return the same set of haplotypes every time. However, we are interested in recovering the metahaplome of multiple, real haplotypes from the set of reads, not just one haplotype. Hansel exposes a function in its interface for the reweighting of observations. Once a path through the graph is completed (a variant has been chosen for all SNP sites), the observations in the Hansel matrix are reweighted by Gretel.

Currently, Gretel reduces the weight of each pairwise observation that forms a component of a completed path - in an attempt to reduce evidence for that haplotype existing in the metahaplome at all, allowing evidence for other haplotypes to now direct the probabilistic search strategy.

Gretel’s outputs

Finally, Gretel outputs recovered sequences as FASTA, requiring no special parsing of results to be able to conduct further analyses. In addition to the sequences themselves, Gretel outputs a ‘crumbs’ file, which contains metadata for each of the recovered sequences: log probability of that sequence existing given the reads, how much of the evidence in Hansel the sequence was supported by, and how much of the evidence was reweighted as a result of that path being chosen.

Currently, Gretel will continuously recover paths out of the remaining evidence until it encounters a node from which there is no evidence that can inform the next decision.

In silico testing methodology

We describe our approach for initial evaluation of our work, using simulated data. We evaluate the performance of our framework against metahaplotomes consisting of synthetic reads derived from randomly generated haplotypes.

Read generation and variant calling

Reads are generated in silico with our Python tool: (shredder) Our synthetic reads are designed to be simplistic: errorless and of uniform length and coverage. The synthetic read sets form a basis for testing the Hansel and Gretel packages during development, as well as providing a platform on which to investigate the influence that parameters such as read length, number of haplotypes, and mutation rate have on recovery.

For a given FASTA file, our tool generates reads of a uniform user-defined length and coverage, for each of the sequences in the file. The tool calculates the number of reads to generate to achieve the approximate coverage, given the length of the sequence, and the selected read length. A BED file can be used to mask particular areas of one or more of the input FASTA sequences.

Uniform coverage is approximated by randomly generating the start positions of all of the reads across the input sequence (and also allowing for up to half of a read to fall off either end of the sequence).

As our tool is aware of the start position of every read that it generates, it is possible to also produce an alignment of those reads in SAM format. This allows us to align reads without introducing biases and assumptions from external tools.

Pileups of our generated reads typically feature many tri- or tetra-allelic sites (especially as mutation rate increases). To avoid diploid tool bias, our evaluation repository also contains a simple snpper tool that generates a VCF for a given BAM. snpper outputs a VCF record for any heterogeneous site. Our haplotype recovery approach is robust to noise arising from sequencing error (see Results). As such we can aggressively call variants by assuming any heterogeneous site is a SNP.

All tools, documentation, and data for evaluation are open source and freely available via our data and testing repository: https://github.com/samstudio8/gretel-test

Evaluating recovery accuracy

To evaluate the accuracy of a run of Gretel, each known input haplotype is compared pairwise to each of the recovered output haplotypes. Each input haplotype is matched to a corresponding “best” recovered haplotype. Best is defined as the output haplotype that yields the smallest Hamming distance from a given input haplotype. For each synthetic metahaplome, we perform a multiple sequence alignment with MUSCLE to determine the definitive SNP positions. When calculating Hamming distance, we consider only these corresponding positions. That is, we exclude the comparison of homogenous sites from the evaluation metric, to ensure we only consider our accuracy on positions that require recovery. For our results we report the proportion of SNPs that were correctly recovered by Gretel, expressed as a percentage.

Comparing sites enumerated by the multiple sequence alignment of the original haplotypes, as opposed to the VCF of each individual read set ensures Gretel is penalised when a SNP has not been called from the read set.

Regardless of quality, all input haplotypes are assigned a best output haplotype. An output haplotype may be the best haplotype for more than one input. If more than one output haplotype has the same Hamming distance, the first that was found is chosen. If Gretel could not complete at least one haplotype (i.e. a pair of adjacent SNP positions were not covered by at least one read), all input haplotypes are awarded 0%.
Synthetic (seq-gen) metahaplomes

With the desire to first test our approach on data sets with well-defined and controllable read properties, but still posing a recovery problem, seq-gen [27] was used to generate sets of DNA sequences that would serve as haplotypes of a synthetic metahaplome. seq-gen simulates the evolution of a nucleotide sequence along a given phylogeny. For testing Gretel, we provided a star shaped guide tree with uniform branch lengths, such that all haplotypes would be equally dissimilar to each other. These uniform branch lengths correspond to the rate of per haplotype base (hb) nucleotide heterogeneity. Thus, each taxa in the tree has a DNA sequence based upon the evolution of the given starting sequence, following simulated evolution at the given rate.

<table>
<thead>
<tr>
<th>Mutation Rate (SNPs/hb)</th>
<th>Average number of variants called</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>17.25</td>
</tr>
<tr>
<td>0.005</td>
<td>71.20</td>
</tr>
<tr>
<td>0.010</td>
<td>141.54</td>
</tr>
<tr>
<td>0.015</td>
<td>209.25</td>
</tr>
<tr>
<td>0.020</td>
<td>277.28</td>
</tr>
<tr>
<td>0.050</td>
<td>640.63</td>
</tr>
<tr>
<td>0.100</td>
<td>1159.62</td>
</tr>
</tbody>
</table>

Table 1: Mean number of variants called over the 900 generated synthetic read sets, for each per-haplotype base (hb) mutation rate. The generated sequences for each metahaplome were 3000 nt long.

The same starting sequence was shared by all of our generated trees. We used a randomly generated sequence of 3000 nt with 50% GC content. We fixed the number of taxa in the trees at five, but varied the mutation rate across seven levels (Table 1). 35 trees were generated, to amortise any effect on haplotype recovery introduced by the alignments of the reads themselves. We generated 6300 read sets (3 read sizes, 6 per-haplotype base levels, 7 mutation rates, 10 read replicates, 5 tree replicates).

Metahaplomes from a mock community

In lieu of a true, annotated metagenome, we sourced a benchmark microbial community from Quince et al. (2017)[28]. The community consists of 5 Escherichia coli strains, and 15 other genomes commonly found in the human gut according to the Human Microbiome Project (HMP). The community is obtained in the supplement to the author’s original manuscript. In their work, 1.504 x 10^9 reads were generated from the 20 genomes, distributed across 64 paired-end samples (11.75 million read pairs per sample). Reads were configured to simulate a “typical HiSeq 2500 run”.

As part of their preprint, the authors made available a subset of the generated mock community. The subset contains 16 samples, with 1 million reads each, for a total of 32 million reads. Reads were assembled with MEGAHIT [30], using default parameters, as per the author’s recommendations (github.com/chrisquince/DESMAN/blob/master/complete_example/README.md commit 9045fe2). Following the example, we discarded assembled contigs shorter than 1 kbp, to yield an assembly described by Table 2.

Table 2: Statistics for our MEGAHIT assembled from read data provided by Quince et al.

<table>
<thead>
<tr>
<th>Contigs</th>
<th>Total bp</th>
<th>Min</th>
<th>Max</th>
<th>N50</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Assembly</td>
<td>≥ 1kbp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17,866</td>
<td>67,189,963</td>
<td>200</td>
<td>689,365</td>
<td>53,290</td>
<td>4605 s</td>
</tr>
<tr>
<td>6,357</td>
<td>61,651,258</td>
<td>1,000</td>
<td>689,365</td>
<td>63,517</td>
<td>-</td>
</tr>
</tbody>
</table>

The original Quince et al. paper also identified 982 single-copy core species genes (SCSGs) for E. coli. Additionally the work provided DNA sequences for all 982 genes, for each of the five different E. coli strains found in the mock community. SCSGs were mapped to the pseudo-reference with blastn, with alignments requiring a threshold of at least 75% of the average length of the five haplotypes for each SCSG. We found that for 814 of the 982 genes, all five strains could be aligned against the pseudo-reference.

Reads across the 16 samples were concatenated to create one paired-end sample containing 16 million reads. The reads were then also mapped to the pseudo-reference with bowtie2 (--sensitive-local).

Gretel was then executed on the aligned reads, once for each of the 814 identified SCSG regions with the aim of recovering the five strain haplotypes from the synthetic short-reads. SNPs were called for each of the 814 identified SCSG regions with the aim of recovering the five strain haplotypes from the synthetic short-reads. SNPs were called for each of the 814 identified SCSG regions with the aim of recovering the five strain haplotypes from the synthetic short-reads.
Recovery from a real metahaplome

A previous experiment [31] isolated RNA from 32 rumen samples from 3 cows over 6 timepoints (0, 1, 2, 4, 6 and 8 hours) after feeding. In preparation for metatranscriptomic sequencing, the polyA fraction was removed (MicroPoly(A) Purist, Ambion). 18S rRNA was also removed (both Riboflamin Plant Kit and Eukaryote Kit, Invitrogen). 16S rRNA was removed (Ribozero RNA removal kit (bacteria), Epicentre) all according to the manufacturer’s protocols. The resulting enriched microbial mRNA was prepared for sequencing using TruSeq Stranded mRNA Library Prep kit (Illumina). Subsequently, the library was sequenced using an Illumina HiSeq 2500 (100bp paired end sequencing). 118 million paired-end reads were generated and are deposited under the ENA study PRJNA419191.

As part of the previous work, reads were partitioned with khmer, assembled with Velvet and proteins were predicted and annotated with Enzyme Commission (EC) numbers using MKkit with the Uniprot database.

Recovery of haplotypes with Gretel

To recover industrially relevant enzyme isoforms from the metatranscriptome, we focused our attention on hydrolases known to be found in the rumen [1]. The existing GFF was filtered to create a subset of all entries with Enzyme Commission (EC) numbers 3.2, 3.4 and 5.3. 3,419 regions from the GFF were identified and cross-referenced to the new read alignment. Regions were filtered with the following criteria:

• minimum coverage ≥ mean minimum coverage (19.7x)
• length ≥ new mean region length (615.7)
• standard deviation of coverage ≤ average standard deviation of coverage over remaining regions (76.79x)

Filtering returned 259 possible candidates. Each sample’s original short-reads were re-aligned to the existing assembly with bowtie2 (--local) before merging all samples with samtools merge to create one canonical alignment of all reads (248,692,426 alignments). Gretel was individually executed over the 259 regions, using the aligned reads to recover haplotypes.

Each set of recovered haplotypes was sorted by descending likelihood. For each haplotype, a corresponding “flattened” consensus was calculated by flipping any base that disagreed with the base call of any haplotype with a better likelihood, to an ‘N’.

PCR Amplicons

Stock RNA from the 32 samples was pooled in proportion with the density of read coverage to the 10 regions from each sample’s corresponding Illumina data. Gene-specific reverse transcription for the ten chosen genes (Table 1) was performed with a Qiagen QuantiTect® Reverse Transcription Kit. Thus, each selected region had an individual corresponding cDNA library.

Gene-specific PCR (30 cycles, 65°C annealing temperature, 20s) was performed for each of the 10 genes with New England Biolabs Phusion® High-Fidelity DNA Polymerase, using the corresponding cDNA (1:10 dilution) and primer pair. Bands were excised following gel electrophoresis and DNA extracted with a Qiagen QIAquick® Gel Extraction Kit. PCR, gel electrophoresis and extraction were repeated to manufacture a sufficient number of amplicons for the Oxford Nanopore ligation protocol.

Five of the 10 sequences (G11, G31, G90, G123 and G251) could be produced at the expected length and adequate amount for Nanopore sequencing. However, G11 was contaminated with rRNA carryover from the reverse transcription and no haplotypes could be determined. Isolated DNA was verified via Sanger sequencing at the Translational Genomics Facility, Aberystwyth.

Nanopore Sequencing

Amplicons were pooled in an attempt to equalize the molarity of the five inputs in the required 1500 ng. The pooled DNA volume was 433 μl and required concentrating. DNA was recovered by following an AMPure Bead Cleanup protocol (60% bead concentration) and resuspended in 46 μl nuclease-free water. We followed the Oxford Nanopore SQK-LSK108 laboratory protocol to prepare a library for sequencing.

The DNA was loaded onto a FLO-MIN106 flowcell. The platform test returned 1,402 viable single cell pores. Sequencing was performed with MinKNOW (v1.7.14) running an unmodified NC_48Hr_sequencing.FLO-MIN106.LSK108 protocol. The run was manually terminated after 1h 28m 35s and yielded 672,388 reads. Base calling was completed with Albacore (v2.02). 634,859 reads passed quality control. Supplementary Figure 1 plots mean phred score and read length, against frequency.

Table 3: Number of Nanopore reads mapped back to the pseudo-reference for the five sequenced genes.

<table>
<thead>
<tr>
<th></th>
<th>G11</th>
<th>G31</th>
<th>G90</th>
<th>G123</th>
<th>G251</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapped</td>
<td>0</td>
<td>70,359</td>
<td>150,860</td>
<td>96,411</td>
<td>9,419</td>
</tr>
</tbody>
</table>

Haplotype Verification

Nanopore reads were aligned to the original five regions of the pseudo-reference (Table 3) with bowtie2 (--sensitive-local). Our Python script (hamming_reads.py) was used to parse the CIGAR strings of the alignment, and calculate the Hamming distance of all reads against recovered haplotypes for each gene. Due to the abundance of homopolymer runs and slippage in the sequenced Nanopore reads, we chose to ignore indels when calculating Hamming distance. Our Circos plots align several single molecule DNA sequences against Gretel’s recovered haplotypes.
<table>
<thead>
<tr>
<th>G#</th>
<th>Protein Name</th>
<th>Forward Primer</th>
<th>Reverse Primer $(5' \to 3')$</th>
<th>PCR Outcome</th>
<th>PCR Outcome Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>G11</td>
<td>P45796 Rabinoxylan arabinofuranohydrolase</td>
<td>ATT TAG CTG CAT C</td>
<td>GCC TGC TGT TAA CAA CAA CAA</td>
<td>Successful</td>
<td></td>
</tr>
<tr>
<td>C31</td>
<td>ALT3 XR</td>
<td>1251</td>
<td>CCC TGT TCC TCT ATA CTT CGC ATG ATG C</td>
<td>rRNA Contamination</td>
<td></td>
</tr>
<tr>
<td>G90</td>
<td>059219 Xyloglucan-specific endo-beta-1,4-glucanase BoGH9A</td>
<td>1464</td>
<td>ATT TCT CAA CGC CAC CGG TCT TAC C</td>
<td>Successful</td>
<td></td>
</tr>
<tr>
<td>G107</td>
<td>D5E715</td>
<td>1407</td>
<td>GGT ATC TCT CTG CAC TAC TAC ACC TGC AC</td>
<td>No PCR Product</td>
<td></td>
</tr>
<tr>
<td>G123</td>
<td>P35835 YncX</td>
<td>1429</td>
<td>TGG TCG AAG ATT CCT TAT TCG GCT ATG TCG</td>
<td>No PCR Product</td>
<td></td>
</tr>
<tr>
<td>G142</td>
<td>Q029470 PII-type proteinase</td>
<td>1425</td>
<td>GCA ACA GCT GTG CTG CCC TGA AGT G</td>
<td>No PCR Product</td>
<td></td>
</tr>
<tr>
<td>G152</td>
<td>P15293</td>
<td>1472</td>
<td>GCA GCG CAG CAC TTG AGG ACG TCA GAC C</td>
<td>No PCR Product</td>
<td></td>
</tr>
<tr>
<td>G162</td>
<td>Q9EB3</td>
<td>1477</td>
<td>AGA AGC CTA CCT TGG AGG AGC TG</td>
<td>No PCR Product</td>
<td></td>
</tr>
<tr>
<td>G165</td>
<td>A4L17L6</td>
<td>1497</td>
<td>CAG CCA TCT GAA CCT CAA GGT GGA AAC</td>
<td>No PCR Product</td>
<td></td>
</tr>
<tr>
<td>G251</td>
<td>Q72818</td>
<td>1107</td>
<td>GCA AGA ACC TCA CCA AGC TCA CTC TC</td>
<td>No PCR Product</td>
<td></td>
</tr>
</tbody>
</table>