BIOINFORMATICS

Vol. 00 no. 00 2011
Pages 1–8

Xenome — A Tool for Classifying Reads from Xenograft Samples

Thomas Conway[1]*, Jeremy Wazny,[1] Andrew Bromage,[1] Martin Tymms,[2] Dhanya Sooraj,[2] Elizabeth D. Williams[2] and Bryan Beresford-Smith[1]

[1]NICTA Victoria Research Laboratory, Department of Computer Science and Software Engineering, The University of Melbourne, Parkville, Australia
[2]Monash Institute of Medical Research, Monash University, Clayton, Australia

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: Shotgun sequence read data derived from xenograft material contains a mixture of reads arising from the host and reads arising from the graft. Classifying the read mixture to separate the two allows for more precise analysis to be performed.

Results: We present a technique, with an associated tool *Xenome*, which performs fast, accurate and specific classification of xenograft derived sequence read data. We have evaluated it on RNA-Seq data from human, mouse and human-in-mouse xenograft data sets.

Availability: *Xenome* is available for non-commercial use from http://www.nicta.com.au/bioinformatics

Contact: tom.conway@nicta.com.au

1 INTRODUCTION

Xenograft models are an important tool for many areas of biomedical research, including oncology, immunology and HIV pathology. A typical scenario, drawn from oncology research, is that of a human prostate cancer grown in an immunocompromised mouse model. Doing so allows researchers to investigate aspects of the cancer that are not necessarily preserved in cell lines, and it allows investigations into the interactions between the cancer and the surrounding stromal tissue. The mouse may be biopsied or harvested and samples of cancer and/or stroma collected at various time points during an experiment.

Difficulties arise, when sequencing the genome or transcriptome of the samples because host (mouse) material (i.e. DNA/RNA) will inevitably comingle with the graft (human) material. If a sufficiently careful section is taken, it has been generally assumed that the level of host contamination is low enough that it may be ignored. This may be a dangerous assumption, however, since the level of gene expression is non-uniform. If the overall level of host contamination in a graft sample is measured to be 10% overall, it may still be the case for a given gene that the host homologue accounts for most or all of the expression.

Contamination may be minimized by physical or biochemical techniques such as conservative sectioning, cell sorting, or laser capture micro-dissection, but these techniques can be a significant source of technical bias, or in some cases may require infeasibly large amounts of starting material. Further, in the case of transcriptomic investigation, classifying host and graft *in vitro* may fail to adequately capture the interactions between them.

An alternative strategy is to sequence an acknowledged mixture of host and graft, then use *in silico* methods to classify the individual sequence reads. This is the approach discussed here. We demonstrate a simple technique, based on an analysis of sequence reads using *Tophat*, and a more precise technique based on a k-mer decomposition of the host and graft reference sequences, *Xenome*. In both cases the primary goal of the analysis is to classify reads into four classes: reads attributable to the *host*, reads attributable to the *graft*, reads which could be attributed to *both*, and reads which are attributable to *neither*.

To the best of our knowledge, there are no results in the literature examining the classification of high throughput sequencing short reads from xenograft models. The studies we know of are concerned with microarray expression profiles or alternative methods for estimating the amount of host material or cell types in the samples. For example, Lin *et al.* (2010) investigate the use of species specific variation in gene length and a multiplex PCR reaction to ascertain the relative amount of mouse and human DNA. Wang *et al.* (2010) use microarray gene profiling data and *in silico* techniques to estimate the quantity of various tissue components. In Samuels *et al.* (2010) there is an analysis of a mouse xenograft model using microarray data. They conclude that if there is more than 90% human DNA then the expression profiles are not unduly skewed. They also describe an experimental method for removing homologous genes based on cross-hybridization analysis of the probes. Ding *et al.* (2010) use short read sequencing to study a cancer genome and identify mutations/deletions. They estimate tumour cellularity using pathological assessment, and state that their xenograft is 90% tumour cells. They also map NOD/SCID (mouse) genomic data to human and mouse genomes, reporting 3.17% and 95.85% mapping rates respectively and so apply no correction for the murine cells. We note that in the context of non-uniform RNA-Seq data ignoring the contribution of the murine expression can lead to biases.

For the remainder of the paper, we will assume, unless otherwise stated, that sequence reads arise from RNA-Seq. However, the techniques we present are applicable to genomic DNA sequences

*to whom correspondence should be addressed

© Oxford University Press 2011.

(including ChIP-Seq and MeDIP-Seq) and also to other mixtures of DNA species.

2 METHODS

Under the assumption that a graft sample has only a low level of host material contamination, the simplest analysis is to use a regular mapping-based RNA-Seq analysis tool, such as *Tophat* and assume that either the observed expression is dominated by the graft, which has the greatest number of input cells, or that the homology between the host species and graft species is such that reads arising from host material will tend to map poorly, and the resultant inferred level of gene expression will be negligible.

In some cases, these assumptions may be true, but in the case of human cancer xenografts in mice, for example, the second assumption is false for many transcripts, and a more precise technique is desirable.

Therefore, we have developed two techniques — one based on the existing RNA-Seq resequencing tool *Tophat* (Trapnell *et al.*, 2009), and one based on k-mer decompositions of the host and graft references. For genomic DNA, another resequencing tool could just as well be used.

2.1 A *Tophat* Based Method

A more precise analysis may be performed by using *Tophat* (Trapnell *et al.*, 2009) to analyze the reads. Firstly, *Tophat* is used to process the read set with the graft genome as reference. Secondly, *Tophat* is used to process the read set with the host genome as reference. Lastly, the accepted alignments from the *Tophat* mappings are post-processed to partition the reads into four classes: *host*, *graft*, *both*, or *neither*. *Tophat* provides mapping quality scores in its output, but they only reflect whether or not the read mapped to multiple locations. If the quality scores reflected a measure of certainty that the read maps to the given location, a more sophisticated approach would be to extend the classification to assign reads that map with high certainty to one genome and low certainty to the other to the appropriate specific class rather than *both*. We do not pursue this further here.

An implementation of this method may be achieved easily with *Tophat*, *SAMtools* (Li *et al.*, 2009) and some simple scripts. As will be apparent in the results presented in Section 3, although very few reads are misclassified (i.e. classified as *host* instead of *graft*, or vice versa), a significant proportion of the reads, even in a pure graft or pure host sample, fall in to the *both* class. If these ambiguous reads were uniformly distributed in their origin across the genome, this would have only a small impact, but as we will elucidate in Section 4, the ambiguous reads are non-uniformly distributed. As a result, a significant number of genes cannot have their expression unambiguously pinned to the host or the graft, though at least compared to a single analysis, the set based analysis makes clear which reads may be clearly associated with the host or the graft, and does not assume that all gene expression in the sample is explained by the graft.

2.2 A k-mer Based Method

Our method proceeds in two phases: constructing a reference data structure, then classifying reads with respect to that reference data structure. The reference structure is built from the sets of k-mers in a pair of reference sequences, which we will refer to as the *host* and the *graft*.

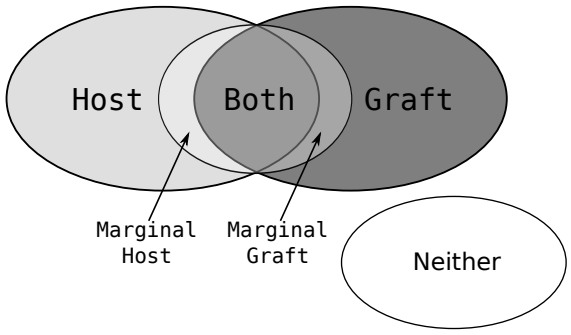

Fig. 1. A Venn diagram showing the different classes that a given k-mer may belong to. The marginal host (and marginal graft) partitions are for those host (and graft) k-mers that are Hamming distance 1 from a k-mer in the graft (and host) reference.

2.2.1 Definitions

DEFINITION 1. *Consider a k-mer x. We denote its reverse complement by $\bar{x}$.*

A canonical k-mer $\hat{x}$ is defined by a choice function $\mathcal{C}$ giving a deterministic choice between x and $\bar{x}$:

$$\forall x : \hat{x} = \mathcal{C}(x) = \mathcal{C}(\bar{x})$$

In principle, we can choose any such function: *min* or *max* being obvious candidates, and the results of our method are identical for all choices. In Section 4 we will present our specific choice which has important performance ramifications.

This definition can be extended to a set of k-mers S in the obvious way:

$$\hat{S} = \{\hat{x} : x \in S\}$$

DEFINITION 2. *Consider a set of canonical k-mers $\hat{S}$. We say that a k-mer x is has marginal membership of $\hat{S}$ if $\hat{x}$ does not exist in $\hat{S}$, but has a Hamming distance 1 neighbor y such that $\hat{y}$ is a member of $\hat{S}$.*

To aid our computation of marginal inclusion, we define the function $\mathcal{M}$:

$$\mathcal{M}(x, \hat{S}) = \{\hat{y} : y \in \mathit{Ham}_1(x)\} \cap \hat{S}$$

where $\mathit{Ham}_1(x)$ is the set of Hamming distance 1 neighbors of x. Note that

$$\{\hat{y} : y \in \mathit{Ham}_1(x)\} = \{\hat{y} : y \in \mathit{Ham}_1(\bar{x})\}$$

2.2.2 Reference Construction For both host and graft reference sequences, we construct the set of canonical k-mers ($\hat{H}$ and $\hat{G}$ respectively). From these we compute a complete set of canonical reference k-mers $\hat{S} = \hat{H} \cup \hat{G}$. Note that $\forall x \in \hat{S}, x = \hat{x}$.

The sets of canonical k-mers tend to be large: with $k = 25$, there are 2.4 billion in the human genome, 2.1 billion in the mouse genome, and 4.5 billion in the union — only 12 *million* are shared.

A naive representation (2 bits per base, packed), would use 50 bits per 25-mer, or about 26 GB. As discussed in our previous work (Conway and Bromage (2011)), information theory gives a lower bound for the memory usage. For a domain of 4^k possible k-mers, the minimum number of bits required to represent a set of n k-mers is

$$\log_2 \binom{4^k}{n}$$

In the case of the union $\hat{S}$ above, this is about 10 GB, or less than half what is required for the naive representation. As also discussed in our previous work, *succinct* data structures have been developed to give concrete representations that approach this theoretical lower bound. The one we use, due to Okanohara and Sadakane (2006), works very well, requiring about 13 GB.

For each reference k-mer, we determine whether it occurs in the host, the graft, both, or if it occurs in one and has a marginal occurrence in the other. These classes are denoted **h**, **g**, **b** or **m**, respectively (see Figure 1). More formally, we compute the function $\mathcal{K}$ for each $\hat{x} \in \hat{S}$:

$$\mathcal{K}(\hat{x}) = \begin{cases} \mathbf{b} & \text{if } \hat{x} \in \hat{G} \wedge \hat{x} \in \hat{H} \\ \mathbf{g} & \text{if } \hat{x} \in \hat{G} \wedge \hat{x} \notin \hat{H} \wedge (\hat{H} \cap \mathcal{M}(\hat{x}, \hat{S}) = \emptyset) \\ \mathbf{h} & \text{if } \hat{x} \notin \hat{G} \wedge \hat{x} \in \hat{H} \wedge (\hat{G} \cap \mathcal{M}(\hat{x}, \hat{S}) = \emptyset) \\ \mathbf{m} & \text{if } \Big(\hat{x} \in \hat{G} \wedge \hat{x} \notin \hat{H} \wedge (\hat{H} \cap \mathcal{M}(\hat{x}, \hat{S}) \neq \emptyset)\Big) \\ & \quad \vee \Big(\hat{x} \notin \hat{G} \wedge \hat{x} \in \hat{H} \wedge (\hat{G} \cap \mathcal{M}(\hat{x}, \hat{S}) \neq \emptyset)\Big) \end{cases}$$

$\mathcal{K}$ may be extended to project not just k-mers to classes, but also a set of k-mers $\hat{Q}$ to a set of classes in the obvious way:

$$\mathcal{K}(\hat{Q}) = \{\mathcal{K}(\hat{x}) : \hat{x} \in \hat{Q}\}$$

These classes are precomputed and stored as a sequence of 2-bit values corresponding to the k-mers in the succinctly store reference set.

The class **m** denotes that the k-mer exists in $\hat{H}$ and has *marginal* membership of $\hat{G}$ or vice versa. That is, they are *marginally* distinctive, in the sense that a single polymorphism or sequencing error may cause a k-mer in that class to change from being marginally host to being marginally graft. Since the marginal set is symmetric (every marginal host k-mer has a corresponding marginal graft k-mer, and vice versa), and represents k-mers that are not very discriminating, we combine both marginal sets into a single marginal set with the class **m**.

2.2.3 Classification Classification proceeds by taking each read r and constructing the set of canonical k-mers that may be derived from the read $\hat{Q}_r$. We then map the set of k-mers to a set of classes by the function $\mathcal{K}$ described above. k-mers that do not occur in the reference, $\hat{S}$, are ignored. The k-mer classes that occur for a given read determine the classification of the read as a whole according to the following function:

$$C(\hat{Q}_r) = \begin{cases} \textit{graft} & \text{if } \mathbf{g} \in \mathcal{K}(\hat{Q}_r) \wedge \mathbf{h} \notin \mathcal{K}(\hat{Q}_r) \\ \textit{host} & \text{if } \mathbf{g} \notin \mathcal{K}(\hat{Q}_r) \wedge \mathbf{h} \in \mathcal{K}(\hat{Q}_r) \\ \textit{ambiguous} & \text{if } \mathbf{g} \in \mathcal{K}(\hat{Q}_r) \wedge \mathbf{h} \in \mathcal{K}(\hat{Q}_r) \\ \textit{both} & \text{if } \mathcal{K}(\hat{Q}_r) \cap \{\mathbf{g}, \mathbf{h}\} = \emptyset \wedge \mathcal{K}(\hat{Q}_r) \neq \emptyset \\ \textit{neither} & \text{if } \mathcal{K}(\hat{Q}_r) = \emptyset \end{cases}$$

The read classes *graft* and *host* denote cases where there is at least one k-mer which unambiguously comes from that k-mer class, and there are no contradictory k-mers (i.e. unambiguously from the opposing reference). The *ambiguous* class corresponds to the case where there are k-mers which appear to be contradictory — unambiguously host, and unambiguously graft. The *both* class represents cases where there are only k-mers which are either unambiguously common to both the host and graft or k-mers which may belong to either if they contain a single polymorphism or sequencing error. The last class, *neither*, represents those cases where there were no matching k-mers.

2.2.4 Implementation Concerns In Section 2.2 we introduced an abstract canonicalization function $\mathcal{C}$. The most commonly used concrete function is *min* (or *max*) which selects the lexicographically/numerically smaller of the k-mers x and $\bar{x}$:

$$\mathcal{C}_{min}(x) = min(x, \bar{x})$$

We use the following definition, assuming some reasonable hash function f:

$$\mathcal{C}_{hash}(x) = \begin{cases} x & \text{if } f(x) < f(\bar{x}) \\ min(x, \bar{x}) & \text{if } f(x) = f(\bar{x}) \\ \bar{x} & \text{if } f(x) > f(\bar{x}) \end{cases}$$

Assuming f is a reasonable hash function, this definition effectively makes a random, but deterministic choice between x and $\bar{x}$.

We use this definition, rather than the more common lexicographic one, because lexicographic canonicalization leads to the set of canonical k-mers being non-uniformly distributed across the set of all possible k-mers.

There are two ways in which a more uniform distribution of k-mers improves performance of *Xenome*. The first is that the succinct bitmap representation that we use (due to Okanohara and Sadakane (2006)) performs better on a uniform distribution of bits (that is, a uniform distribution of k-mers). The second is that it improves the performance of our intermediate hash table, from which the succinct bitmap data structure is built.

The way the hash table is used is that as the input sequences are read, they are decomposed into k-mers which are stored in the hash table, which is of a fixed (controlled by a command line parameter) size. When the hash table fills (that is, unresolvable collisions arise), it is sorted and written out to disk. When all source k-mers have been read, the sorted runs are merged and the main succinct bitmap is constructed.

The specific representation used is a succinct cuckoo hash table, broadly similar to Arbitman *et al.* (2010). The succinct representation relies on the fact that the location of the slot where a key x (in this case, a k-mer) is stored contains some of the information present in the key.

Consider an idealized hash table with load factor 1 (i.e. the number of values stored in the table equals the number of slots in the hash table), with no collisions. If the width of the hash table is 2^J, then J bits of the key are implied by the slot chosen. For keys of N bits, therefore, the entries of the hash table need store only $N - J$ bits. The simplest way this may be realized would be to just use J bits of the key as the slot number, and store the remainder in that slot, but of course, this is likely to have an unacceptable number of collisions.

Instead, we use an invertible hash function (based on a single-stage Feistel network, see (Luby and Rackoff, 1988)) to turn a key

x into a slot number s and a stored component v in such a way that given s and v we can recover x (assuming a hash function f):

$$\begin{aligned}\mathcal{F}_f(x) &= \left\langle \left(x \mod 2^J\right) \oplus f(\tfrac{x}{2^J}), \tfrac{x}{2^J} \right\rangle \\ \mathcal{F}_f^{-1}(\langle s, v\rangle) &= s \oplus f(v) + v2^J\end{aligned}$$

The size of the key data stored in the hash table is, then, $2^J(N - J)$ bits. If $2^J \ll 2^N$, this is within $(1 + o(1)) \log_2 \binom{2^N}{2^J}$ bits, and hence succinct.

(Of course, idealized hash tables are not possible if the set of keys is not known in advance. Hence we use cuckoo hashing (Pagh and Rodler, 2004), in which collisions are resolved (as far as possible) using multiple hash functions.)

To make the sorting of keys more efficient, our hash function preserves the $N - J$ most significant bits of the keys, which are then stored in the hash table (along with a few bits to determine which hash function was used). This means that we can perform an initial bucketing without inverting the hash functions; the buckets can then be processed independently, using multiple threads if required.

Now consider a set of random k-mers. For hash-based canonicalization, given a good hash function, there is an equal probability of observing any base in the most significant position in $\hat{x}$. For lexicographic canonicalization, there is a probability of $\frac{4}{10}$ that it is A, $\frac{3}{10}$ that it is C, $\frac{2}{10}$ that it is G, and only $\frac{1}{10}$ that it is T. For a set of N random k-mers, the expected entropy at the most significant base is 2.0 bits and 1.85 bits respectively.

This matters, because hash functions are frequently vulnerable to poor behavior in the presence of highly correlated sets of keys. In real, biologically derived sets of k-mers, the set of k-mers is non-random, so there is likely to be less entropy to begin with, and therefore loss of entropy due to canonicalization is exacerbated. This is especially problematic in the case of the above family of invertible hash functions, since it is precisely the most-significant bits of the key which are passed to the underlying hash function f.

The upshot of this is that $\mathcal{C}_{min}$ leads to highly correlated k-mers, which in turn lead to a high probability of unresolvable collisions even when the hash table is nearly empty, resulting in a large number of short runs. Because the set of canonical k-mers that result from $\mathcal{C}_{hash}$ are less correlated, the hash table is unlikely to encounter unresolvable collisions until it is almost full (80%–85% in practice).

2.3 *Xenome* — Usage and Workflow

To use the `xenome` tool, first it must be invoked to construct the reference data structures. A typical invocation will give the FASTA filenames for the host and graft genomes, a filename prefix for the index files, and optionally, the amount of working RAM to use (in GB), and the number of threads to use:

```
$ xenome index -M 24 -T 8 -P idx \
    -H mouse.fa -G human.fa
```

This will run for some time — on the human/mouse references, around 4–6 hours on an 8-core server. This need only be done once for a given pair of references, and a given k-mer size. The k-mer size defaults to 25, which seems to work well.

With the reference data structures built, read data may be segregated. To make the output files easier to identify, command line flags can be used to name the host and graft files. If paired data is being classified, the `--pairs` flag should be given — pairs are classified by computing $\mathcal{K}(\hat{Q})$ over all the k-mers in the pair.

```
$ xenome classify -T 8 -P idx --pairs \
    --graft-name human --host-name mouse \
    --output-filename-prefix XYZ
    -i XYZ_1.fastq -i XYZ_2.fastq
```

This yields the following set of files:

```
XYZ_ambiguous_1.fastq  XYZ_ambiguous_2.fastq
XYZ_both_1.fastq       XYZ_both_2.fastq
XYZ_human_1.fastq      XYZ_human_2.fastq
XYZ_mouse_1.fastq      XYZ_mouse_2.fastq
XYZ_neither_1.fastq    XYZ_neither_2.fastq
```

If the total size of the index files is larger than RAM, `xenome` will perform poorly, but the flag `-M` can be supplied with the maximum desired working set size, and the classification will be done in multiple passes each using less memory.

On a single server with 8 AMD Opteron cores running at 2 GHz and with 32 GB of RAM *Xenome* processes approximately $15,000$ read pairs per second.

Having classified the reads, typical usage would be to then run *Tophat* and *Cufflinks* to perform intron-aware gapped alignments and compute gene expression. Instead of running *Tophat* on all of the reads, having separated the reads according to their origin, we can run it with the human genome just against the human fraction of the reads, and against the mouse genome on the mouse fraction of the reads. It may well be desirable to combine the *both* and *ambiguous* fractions with the human fraction to run against the human genome, and also with the mouse fraction to run against the mouse genome. If this is done, attention should be paid to homologous genes, since it is possible that the human and mouse homologues may be represented by the same reads. In some cases, the correct attribution of the gene expression may be apparent from the nature of the experiment. For example, a sample from a human prostate cancer grown in mouse may show ambiguous expression in MYH genes which would be reasonably attributed to the stromal mouse tissue.

3 RESULTS

Our analysis examines two questions: whether or not it is technically feasible to separate host and graft reads *in silico*; and whether or not the fast technique we have proposed (*Xenome*) yields a worthwhile improvement over the mapping (*Tophat*) based technique.

In the first experiment, we take a sample of human cDNA sequence data (SRR342886), and a sample of mouse cDNA sequence data (SRR037689) and analyze them both with *Tophat* and *Xenome*, and compare the results. This allows us to evaluate the degree to which sequences are misclassified (assigned to *human* rather than *mouse* or vice versa), and the specificity of the classification — the proportion of sequences which are not classified as *both*. The use of pure human or mouse cDNA gives an experiment where the correct assignment of reads is known.

The second experiment runs the same analysis on sequence data from a human prostate cancer xenograft growing in a mouse host (BM18, see McCulloch *et al.* (2005)). In this case, however, we not only classify the reads, but use the *Tophat* mappings to compute approximate levels of gene expression (measured in fragments per

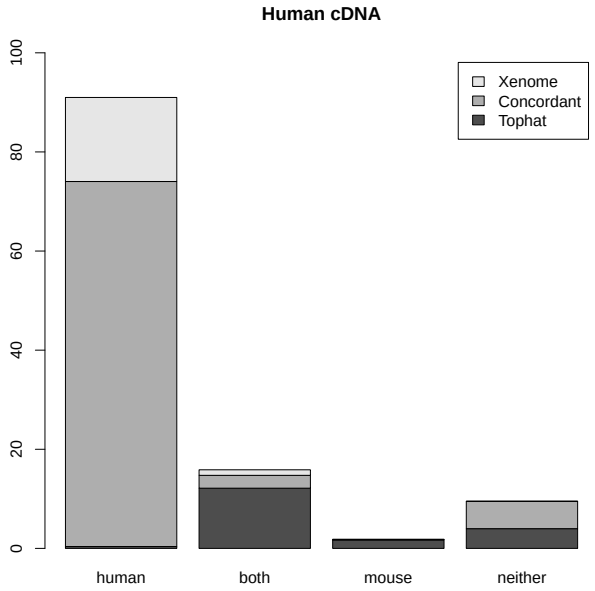

Fig. 2. Summary of the results with Human cDNA. Each of the classes of reads is divided into those reads assigned to the class only by *Xenome* (*Xenome*), only by the *Tophat* analysis (*Tophat*) or by both *Xenome* and the *Tophat* analysis (Concordant).

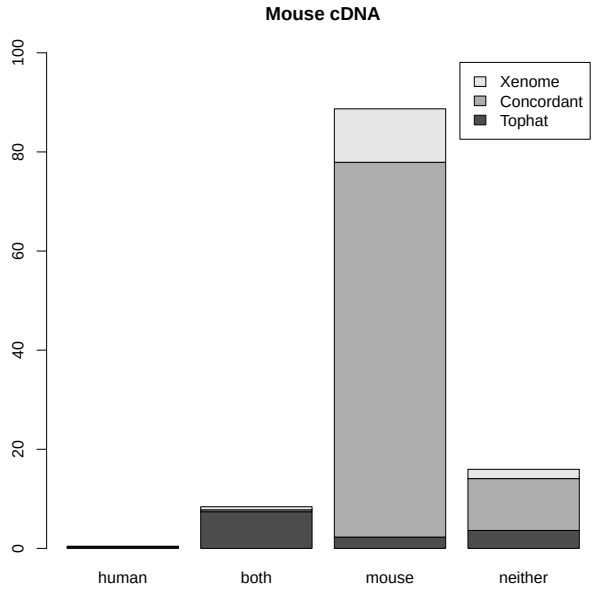

Fig. 3. Summary of the results with Murine cDNA.

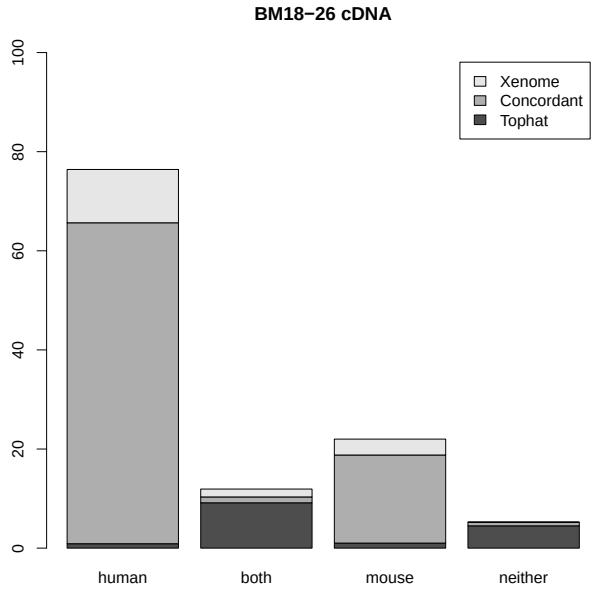

Fig. 4. Summary of the results with BM18 xenograft cDNA.

thousand bases of transcript per million mapped reads, or *FPKM* (Trapnell *et al.* (2010)) and use human species specific quantitative RT-PCR (qRT-PCR) on selected genes to validate the results. In this instance, we have no gold standard by which we can judge the results, but the qRT-PCR will give some degree of validation, and known aspects of the biology of the cancer can give some qualitative corroboration.

Tophat uses a global analysis, combining the results of all the read mappings to locate exons, junctions and so on. By contrast, *Xenome* performs pre-computation on the two reference genomes, then classifies each read independently. Therefore, for each of the three sets of reads, we ran *Tophat* with the human reference genome and again with the mouse reference genome, then, as described in Section 2.1, the mappings were post-processed to determine which reads belonged to each of the four classes. Each of the four partitions of the sets of reads was then partitioned with *Xenome* to allow us to easily determine which reads were classified as the same by both procedures, and which were classified differently.

Figures 2, 3, and 4 summarize the results. For the three samples, the proportion of reads receiving the same classification were 82%, 87% and 84% respectively. As can be seen from the human and mouse only figures, both techniques are accurate, in as much as they misclassify only a small proportion of the reads (the worst case being the *Tophat*-based analysis of the human cDNA which misclassified 1.2% of the reads — all the other analyses misclassified 0.2% – 0.3%). The main difference between the *Tophat*-based and *Xenome* analyses is that the latter yields better specificity — the fraction of reads classed as *both* is significantly smaller in the *Xenome* analysis. From this we can conclude that the *in silico* classification of sequences is feasible and accurate.

The second experiment, using RNA-Seq data from a prostate cancer xenograft into a mouse demonstrates that the classification works on a real mixture. As described above, we performed the

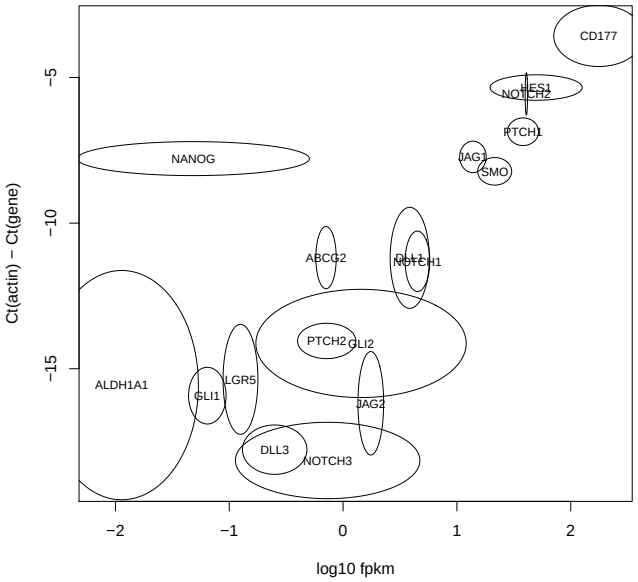

Fig. 5. Validation of the *in silico* classification of xenograft RNA-Seq data with qRT-PCR. The horizontal axis shows $\log_{10} FPKM$ for the *Xenome*-derived gene expression for the 18 test genes. The vertical axis shows the Ct value for each gene relative to the Ct of actin. There were two RNA-Seq samples processed (biological replicates), and 4 replicates of the qRT-PCR. For each gene, an ellipse is shown centered on the mean $\log_{10} FPKM$ in the x-axis, and on the mean relative Ct in the y-axis. The horizontal and vertical radii show the variance in the samples.

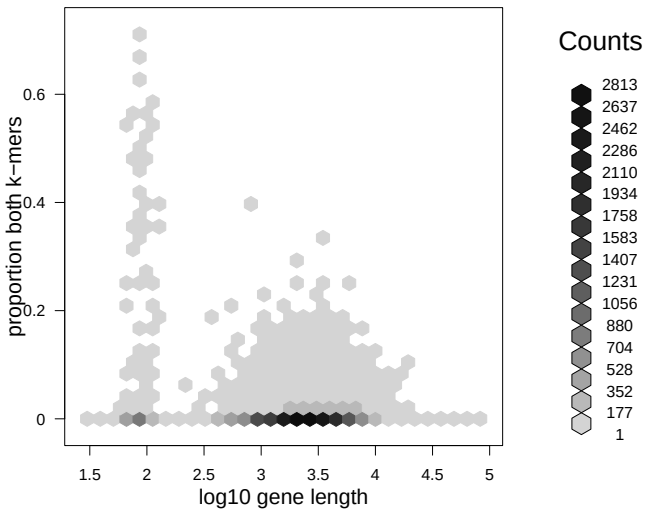

Fig. 6. An *in silico* analysis showing the degree of ambiguity in HG19 refGene, according to the k-mer based analysis used by *Xenome*. In this analysis, $k = 25$.

same process to partition the reads into classes, then we used the refGene genome coordinates as calculated by *Tophat* to assign reads to genes, from which we computed an expression level using the fragments per kilobase of transcript per million mapped reads formula (Trapnell *et al.* (2010)):

$$FPKM = \frac{f \times 10^9}{zN}$$

where f is the number of fragments (reads, for single ended data, or pairs for paired data), z is the combined length of the exons of the gene, and N is the total number of mapped fragments. Although this quantification is peripheral to the technique we are presenting, we have computed expression levels for the purposes of comparing with some qRT-PCR data for the same biological data. The qRT-PCR data were available for 18 genes: ABCG2, ALDH1A1, CD177, DLL1, DLL3, GLI1, GLI2, HES1, JAG1, JAG2, LGR5, NANOG, NOTCH1, NOTCH2, NOTCH3, PTCH1, PTCH2, and SMO. Figure 5 shows the $\log_{10} FPKM$ versus the difference of the Ct for each target gene and the Ct for actin (which was used as a housekeeping gene). With the exception of NANOG, the two methods correlate reasonably well (the Pearson correlation coefficient is 0.80). We have investigated the NANOG data, and cannot explain the low *FPKM*. Whether this is a sequencing issue or a biological variation in the mice is unknown, but the low level of expression does not appear to be related to the behavior of *Xenome* or *Tophat*.

4 DISCUSSION

We have presented a simple read classification method based on *Tophat*, and our refined classification approach, *Xenome*. *Xenome* can be used to efficiently and effectively partition the read set for subsequent processing by tools such as *Tophat*.

What is not apparent from the results above is the relative behaviour at the level of a single gene. It should be expected that the distribution of ambiguously mapped reads (classed as *both*) should be non-uniform, since some genes in the two genomes are more highly conserved than others.

The first result we present in Figure 6 on this point is an *in silico* analysis showing the proportion of each human gene (ignoring introns) covered by k-mers that are not classed as *human*. It is clear that the vast majority of genes contain few or no k-mers that are not classed as *human*.

The fraction of k-mers which are ambiguous gives a worst-case view of how *Xenome* might be expected to perform. In order for a read to be classified as *both*, all of its k-mers must be of the *both* class, or there must be at least one k-mer from each of the two genomes (which happens less than 2% of the time in the samples we have tried). Conversely, a single *host* or *graft* k-mer is sufficient to classify the read into the respective class. Therefore for a read to be classified as *both*, the reference must contain a sufficiently long run of consecutive k-mers of class *both* and/or SNPs and sequencing errors must eliminate all the distinctively *host* or *graft* k-mers.

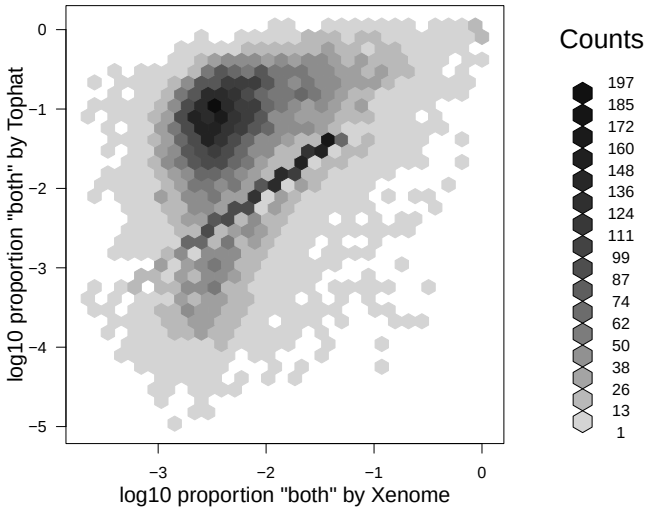

Fig. 7. A plot showing the distribution of human genes with respect to the proportion of xenograft reads which are classed as *both* by the *Tophat* based analysis and the *Xenome* analysis. The reads considered are only those mapped by *Tophat* since *Xenome* does not yield mappings, so cannot be used to assign reads to genes. Only genes for which at least 20 reads mapped were considered. The horizontal axis corresponds to the number of reads classified as *both* or *ambiguous* by *Xenome* as a proportion of all the reads that might possibly be human (i.e. *both*, *ambiguous*, or *human*). The vertical axis corresponds to the number of reads classified as *both* by the *Tophat* based analysis, once again, as a proportion of all the reads that might possibly be human.

The second result we report on this point, presented in Figure 7 is the relative proportion of reads which are classified as *both* on a per-gene basis. What is evident in this figure is that although there are many genes for which the proportion of *both* reads is tightly correlated between *Tophat* and *Xenome*, there are a large number of genes for which the *Tophat* based analysis has significantly more *both* reads. There are 15,591 genes for which there were at least 20 mapped reads in the BM18 xenograft sample. Of these, there were 65 for which *Xenome* assigned *both* or *ambiguous* to at least half the reads mapping to the gene; there were 498 for which the *Tophat* based analysis assigned *both* to at least half the reads mapped to the gene. For the most highly conserved genes, there is not much that can be done with this data directly — further signal processing or other data would be required to determine the relative expression in the host and graft.

While we have developed *Xenome* with RNA-Seq on human/mouse xenografts in mind, we anticipate it will be an effective tool for other similar mixtures. For example, capturing the differential methylation around genes between host and graft using MeDIP-Seq may shed light on the interraction between the two.

The need for our technique is substantially motivated by the fact that for a xenograft to be viable there must be very strong homology between the host and graft organisms. This leads to a situation where there is a high probability that a read may map to either genome, and it is this problem that *Xenome* specifically addresses. A side benefit is that the classification is done independently for each read, and results in groups of reads in each class; each group may then be processed independently with further tools (such as *Tophat*, *Cufflinks* (Roberts *et al.*, 2011), or others). Given that many kinds of analysis require global processing of the input data, being able to process a coherent subset of the data can lead to a time/space gain. This benefit extends beyond the sphere of xenografts. For example there are situations where a parasite or pathogen cannot be cultured independently (for example Chlamydia, and some fungi), so samples will generally contain a mixture of host and pathogen. In some examples, although the pathogen can be cultured independently, there are phenotypic differences between organisms growing in culture and those growing in a host. In both cases, there are benefits to being able to classify the two groups of reads, even though straight mapping based approaches will be less sensitive to cross-talk than xenograft data.

ACKNOWLEDGEMENT

National ICT Australia (NICTA) is funded by the Australian Government's Department of Communications; Information Technology and the Arts; Australian Research Council through Backing Australia's Ability; ICT Centre of Excellence programs.

This project was supported in part by funding from the Prostate Cancer Foundation of Australia (EDW) and the Victorian Government's Operational Infrastructure Support Program. EDW is supported by an Australian NHMRC Career Development Award (#519539).

REFERENCES

Arbitman, Y., Naor, M., and Segev, G. (2010). Backyard cuckoo hashing: Constant worst-case operations with a succinct representation. *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 787–796.

Conway, T. C. and Bromage, A. J. (2011). Succinct data structures for assembling large genomes. *Bioinformatics*, **27**(4), 479–86.

Ding, L., Ellis, M., Li, S., Larson, D., Chen, K., Wallis, J., Harris, C., McLellan, M., Fulton, R., Fulton, L., *et al.* (2010). Genome remodelling in a basal-like breast cancer metastasis and xenograft. *Nature*, **464**(7291), 999–1005.

Li, H., Handsaker, B., Wysoker, A., and Fennell..., T. (2009). The sequence alignment/map format and samtools. ....

Lin, M., Tseng, L., Kamiyama, H., and Kamiyama..., M. (2010). Quantifying the relative amount of mouse and human dna in cancer xenografts using species-specific variation in gene length. ....

Luby, M. and Rackoff, C. (1988). How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, **17**(2), 373–386.

McCulloch, D., Opeskin, K., Thompson, E., and Williams, E. (2005). BM18: A novel androgen-dependent human prostate cancer xenograft model derived from a bone metastasis. *The Prostate*, **65**(1), 35–43.

Okanohara, D. and Sadakane, K. (2006). Practical entropy-compressed rank/select dictionary. *CoRR*, **abs/cs/0610001**.

Pagh, R. and Rodler, F. F. (2004). Cuckoo hashing. *J. Algorithms*, **51**(2), 122–144.

Roberts, A., Pimentel, H., Trapnell, C., and Pachter, L. (2011). Identification of novel transcripts in annotated genomes using RNA-seq. *Bioinformatics*, **27**(17), 2325–2329.

Samuels, A., Peeva, V., Papa, R., and Firth..., M. (2010). Validation of a mouse xenograft model system for gene expression analysis of human acute lymphoblastic leukaemia. *BMC* ....

Trapnell, C., Pachter, L., and Salzberg, S. L. (2009). TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, **25**(9), 1105–1111.

Trapnell, C., Williams, B., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M., Salzberg, S., Wold, B., and Pachter, L. (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology*, **28**(5), 511–515.

Wang, Y., Xia, X., Jia, Z., Sawyers, A., and Yao. . . , H. (2010). In silico estimates of tissue components in surgical samples based on expression profiling data. *Cancer research*.