Efficient sequence clustering for RNA-seq data without a reference genome

Florian Battke[1†*], Stephan Körner[1†], Steffen Hüttner[2], Kay Nieselt[1]
[1] *Center for Bioinformatics, University of Tübingen, Germany*
[2] *Hölle & Hüttner AG, Derendinger Str. 40, 72072 Tübingen, Germany*
[†] *equally contributing authors*

battke@informatik.uni-tuebingen.de

Abstract: New deep-sequencing technologies are applied to transcript sequencing (RNA-seq) for transcriptomic studies. However, current approaches are based on the availability of a reference genome sequence for read mapping. We present Passage, a method for efficient read clustering in the absence of a reference genome that allows sequencing-based comparative transcriptomic studies for currently unsequenced organisms. If the reference genome is available, our method can be used to reduce the computational effort involved in read mapping. Comparisons to microarray data show a correlation of 0.69, proving the validity of our approach.

1 Background

Changes in transcription are the most important mechanism of differentiation and regulation. Until recently, the transcriptional activity of a cell was measured by PCR in the case of few genes, or microarrays were used to investigate the whole transcriptome of an organism or tissue. Both methods require previous knowledge about the organism's transcripts, either in the form of ESTs or a complete reference genome sequence for primer resp. probe design. SAGE (serial analysis of gene expression) [VZVK95] is a method to study transcriptional activity based on sequencing of short transcript fragments. The advent of new *deep sequencing* technologies (also called next-generation or second generation sequencing methods) now allows to study the transcriptome in unprecedented detail by directly sequencing the pool of expressed transcripts. Using RNA-seq [WGS09] and a known reference genome, transcriptional activity can be measured with single-base precision.

Sequencing the pool of expressed transcripts creates millions of short (36-500 bases) sequences, called *reads*. These need to be mapped against the reference genome sequence allowing for mismatches due to sequencing errors or SNPs, which creates a huge computational challenge. Many

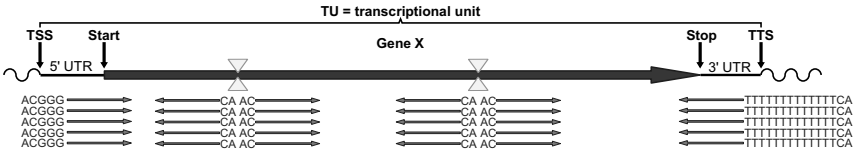

Figure 1: Schematic view of a transcriptional unit. RsaI restriction sites are indicated by yellow triangles. Transcript fragments (red) are sequenced starting from restriction sites in either direction, downstream from the transcription start site (TSS) as well as upstream from the transcription termination site (TTS), resulting in different sequence prefixes (GGG, CA, poly-T).

tools exists for that task, such as SOAP2 [LYL+09], MAQ [LRD08], VMatch [Kur03], RazerS [WER+09], and Bowtie [LTPS09]. Some programs are able to map reads covering splice junctions (TopHat [TPS09], QPALMA [DBOSR08]), others can map reads against several genomes at once, such as GenomeMapper [SHO+09]. Secondly, though more and more reference genomes are made available, the vast majority of organisms remain unsequenced and thus beyond the scope of RNA-seq studies.

Here we present Passage [Hü09], extending the idea of SAGE to create a new efficient method for transcriptome studies in the absence of a reference genome sequence. It makes use of a newly established experimental protocol resulting in reads originating only from well-defined genomic positions. Passage clusters these reads very efficiently to compute expression levels. Comparative studies can also be performed easily based on our method.

2 Material and Methods

Sample Preparation Purified mRNA is incubated with anchored Oligo (dT_{13}) and modified Smart (dG_3) oligonucleotides. These primers contain RsaI restriction sites. Reverse transcription is performed to obtain cDNA which is then amplified using long-distance PCR. After purification steps, the cDNA is cut into *transcript fragments* using the restriction enzyme RsaI. This step replaces the fragmentation step (e.g. by sonication) that is usually performed in RNA-seq protocols. Sequencing adapters are ligated to the fragments, and the fragments are analyzed by deep-sequencing. The universal primer site can be used for different sequencing techniques such as GS FLX™ (Roche Diagnostics/454) and the Genome Analyzer™ (Illumina). Barcode sequences can be included in the adapters to allow parallel sequencing of several samples. The resulting reads start with the

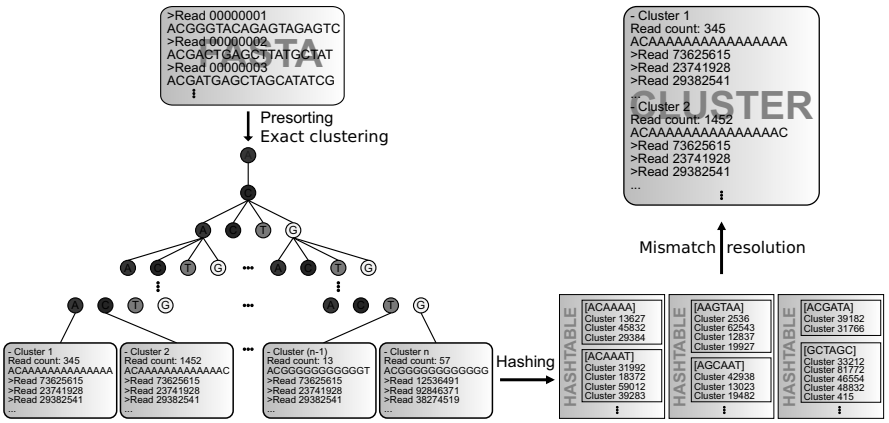

Figure 2: PASSAGE workflow: Reads for one sample (after optional presorting) are used to build a trie, resulting in a clustering of perfectly matching sequences. The reads' sequences are split into three parts and a hash map is built for each such part. These hash maps are used in the mismatch resolution step to cluster all reads with at most two mismatches, resulting in the final clustering output.

optional barcode, followed by a *prefix* and the genomic sequence. Three types of fragments can be distinguished by their prefixes: 5' UTR fragments start with `ACGGG`, 3' UTR fragments with $\texttt{ACT}_{13}$, and internal fragments with the restriction sequence `AC` (see figure 1). This protocol is adapted from [LRR+10] describing a 3'-fingerprint analyzed on a 2-D gel electrophoresis system to next-generation sequencing transcriptomics.

Cluster algorithm Our read clustering algorithm, PASSAGE, employs a three-step process. Starting from a fasta file containing read sequences, this involves *presorting*, *exact clustering* and *mismatch resolution*. The result is a file containing the number of reads contained in each cluster, the cluster's consensus sequence, the IDs of the reads, and a normalized expression estimate (reads per million reads). Result files from multiple samples can be joined into a tabular file containing one column per sample and one row per cluster, which can be analyzed with any standard microarray analysis software.

Presorting Reads are sequenced either from the 3', the 5' or the interior region of a transcriptional unit. This is reflected in different read *prefixes* (see figure 1). Differentiating the reads by these prefixes is not only useful for reducing computational costs, but rather to allocate the reads to the different parts of a transcriptional unit. With the addition of *barcode* sequences, different samples can be analysed in the same se-

quencing run, adding another common prefix for all reads of that sample. During presorting, reads are sorted according to their barcode (to distinguish different samples) and their prefix.

Exact clustering Based on the presorting result, each prefix is processed as follows. A trie of read sequences is generated such that reads are assigned to a common leaf if their sequences are identical. Since we know that reads either overlap by 100% or not at all, this effectively clusters all reads deriving from the same transcript fragment. Reads are placed into the tree by matching their bases one by one to the corresponding tree path until either a leaf is reached (and the read sequence is completely matched) or a new branch has to be created to accomodate the read's sequence. The result of the tree building step is a list of clusters, each cluster containing identical reads.

Mismatch resolution No current sequencing technology is error-free, thus we can not expect all reads from the same locus to be identical. In order to resolve this, we include a mismatch resolution step in our clustering algorithm. If k mismatches should be allowed, the minimal perfect match length results from equidistant distribution of these mismatches over the clusters' sequences. Thus we partition the clusters' sequences into $k + 1$ parts and create $k + 1$ hash maps. Clusters are placed into these hash maps according to the parts of their sequences. Thus, two clusters differing by at most k mismatches will be found in the same hash bucket in at least one of the hash maps. To ensure similar load factors in the presence of long common sequence prefixes, the first sequence part is slightly longer.

Clusters of identical reads are processed according to their size, starting with the largest cluster (in terms of the number of reads contained). From each hash map, candidate clusters are selected for merging. Ungapped alignments are computed and clusters are merged if their distance is at most k mismatches. Merged clusters are removed from all hash maps and the process is repeated until all clusters have been processed. Analysis showed that usually there is one very large and several smaller clusters for a given locus, and that the reads in the large cluster accurately represent the true genomic sequence. Thus we use the largest cluster's sequence as the consensus sequence for the joined cluster.

3 Results

We illustrate our method using the two closely related yeast species *Candida albicans* and *Candida dubliniensis*. Both are facultative pathogens,

			100pg	1μg
Dataset 1, 40bp	*C. albicans*	Y, 30°	4.1	† 4.6
		· YF, 37°	3.7	5.1
	C. dubliniensis	· HF, 37°	4.9	4.8
		YF, 37°	4.2	4.7
Dataset 2, 76bp	*C. dubliniensis*	· HF, 37°	7.6 / 7.6	∗3.8 / 5.6
		YF, 37°	4.4 / 5.5	∗6.1 / 8.6

Table 1: Conditions and number of reads (millions) sequenced for the two datasets. Y, yeast extract peptone dextrose, is a complete medium for yeast growth. F, fetal calf serum (10%). H, H_2O. The amount of total RNA used for sequencing was either 100pg or 1μg. The second dataset contains replicate sequencings. Hyphae-inducing conditions are marked with (·). Data used for comparison with other tools is indicated with (†), those used for validation using microarrays are marked with (∗).

C. albicans is of higher clinical importance as the most common agent causing candidosis. Both species have a genome size of about 14Mb organized in eight chromosomes and roughly the same number of genes (6185 in *C. albicans*, 5983 in *C. dubliniensis*). Cultures were grown under different conditions to study the induction of yeast or hyphae cell proliferation. RNA-seq data was generated from different amounts of total RNA and different read lengths (see table 1). In total, we analyzed 16 RNA-seq runs, using PASSAGE with a maximum of two mismatches.

To assess the robustness of the protocol, we compared the two sequencings for each sample in dataset 1 by mapping the reads to all annotated genes (using Bowtie with up to two mismatches). More than 80% of the annotated genes found in the 100pg sample were also found in the 1μg sample, with the total number of transcripts being about twice as high in the 1μg samples (mean 4344 vs. 2247).

Data volume reduction Both clustering steps significantly reduce data volume (see figure 3). The efficiency of data reduction depends on the quality of the sequencing process. Fewer mismatches allow more reads to be clustered to their correct cluster and thus increase the reduction factor. In our studies using 16 different datasets (8 with 40-mer reads, 8 with 76-mer reads), exact clustering reduced data volume by about 84% (factor 6.1). Mismatch resolution results in a further reduction by about 58% (factor 2.4), resulting in a total reduction of about 93%. The reduction during perfect clustering can be seen as the result of summarizing the

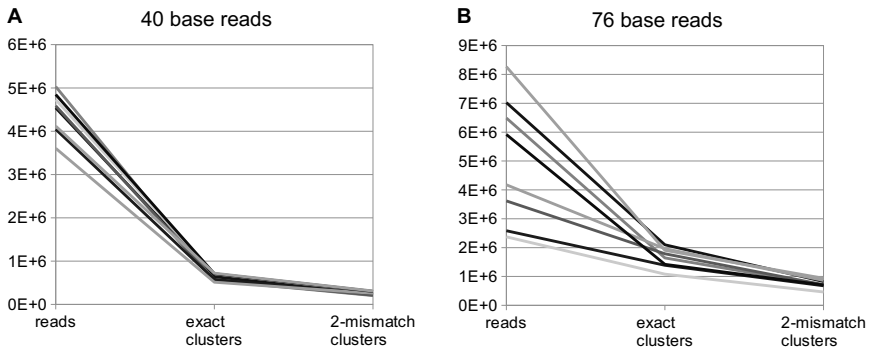

Figure 3: Reduction in data volume (number of reads resp. clusters) achieved by exact clustering and mismatch resolution. 16 datasets were analysed, eight of them with reads of length 40, eight with length 76. Longer reads (B) result in less reduction than shorter reads (A) due to higher error rates.

transcription strength (which varies between conditions) to the number of uniquely sequenced transcripts (which is expected to be more similar for all conditions). During the mismatch resolution step, a reduction is achieved by correcting for the error rate inherent in the sequencing technology, which should also be similar for all experiments.

Runtime analysis Presorting is important to reduce runtime and memory consumption and can be accomplished in $O(n)$ where n is the number of input reads. Exact clustering can also be done in $O(n)$. The time complexity of the mismatch resolution step depends on the average size of the hash buckets and the initial size of the cluster list: If c exact clusters are hashed randomly into buckets, let the average bucket size be ℓ. Merging the clusters can then be done in $O(\frac{c}{\ell} \cdot \ell^2)$. In the worst case, all clusters are hashed into one bucket, yielding $\ell = c$ and $O(c^2)$ runtime. The optimal case would be $\ell = 1$, yielding $O(c)$ runtime.

For real data, we see very small values for ℓ: We found $\ell = 2.5$ for reads of length 76 and $\ell = 4$ for reads of length 40. Thus, for the average case ℓ can be considered constant which results in a runtime of $O(c)$ (see figure 4A). Even in cases where a large number of clusters are collected in one hash bucket, we observe runtime linear with respect to the sum of sizes of the largest buckets in each hash map (see figure 4B). As c is bounded by the number of reads, n, overall average runtime is $O(n)$.

Comparison to other tools PASSAGE was written to cluster short reads and to use the size of the clusters as a measure of transcription.

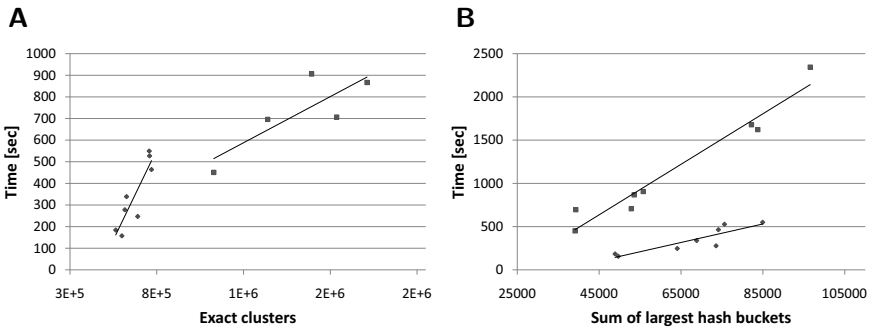

Figure 4: Runtime measurements for the mismatch resolution phase for reads of length 40 bases (diamonds) and 76 bases (squares), respectively, with linear regression curves. On average, runtime is linear in the number of exact clusters used as input (A). In the case of very uneven distribution of clusters to hash buckets, average runtime remains linear with respect to the sum of sizes of the largest hash bucket in each of the three hash maps (B). In both cases, we observe different slopes depending on the length of the reads.

It makes use of the fact that reads either overlap completely or not at all. We believe that no other tool currently offers the same functionality. However, in order to test our algorithm against other tools, we selected the EST assemblers Cap3 [HM99] and Mira3 [CWS99] as well as the short read de novo assemblers Velvet [ZB08] and Locas [KOS+10]. As input we chose a FastA file with approx. 4.6 million Solexa reads of length 40 (184 million bases). Tests were performed on a computer with a 2.5Ghz processor and 8 GB of memory.

We used Bowtie to compute a direct read mapping against the genome of *C. albicans* (assembly 21, obtained from `www.candidagenome.org`) to get the number of "real" clusters. We allowed at most two mismatches (after removing the 3' and 5' prefixes from the respective reads). 3.97 million reads (86%) were mappable and were consequently used for the comparison. Bowtie mapped the reads to 49235 unique mapping positions, thus all methods that produce a significantly lower number of clusters (resp. contigs) combine expression measurements that should be kept separate.

Table 2 shows the runtime and number of assembled clusters for each program used here. It is important to note that these were written for generic assembly tasks while Passage is optimized for our biological protocol. We tried to set parameters such that the results would be most closely comparable to those obtained by Passage. While loading the reads, Cap3's memory consumption grew rapidly beyond the physical limit of our ma-

Program	CPU time	memory	clusters/contigs	*mcl*
Passage	1 min	1200 MB	44,817	40
Cap3[a]	–	>16000 MB	–	–
Mira3	12h 47 min	5600 MB	74,017	40.07
Locas	4h 38 min	5500 MB	2,650	40.08
Velvet	2 min	1300 MB	217	44.55
Bowtie[b]	6 min	21 MB	49,235[c]	–

Table 2: Runtimes and resulting number of clusters/contigs for all tested programs. *mcl*, mean consensus length. [a]Cap3 did not complete due to memory restrictions; [b]Bowtie requires a genomic sequence; [c]unique mapping sites.

chine. The program terminated after filling all available memory. Mira3, Velvet and Locas worked within the limits of our setup. Velvet runs very fast, producing only a very small number of contigs. These contigs are also too long on average, suggesting that it did too good a job of assembling mismatching reads and thus expression estimates derived from Velvet's output are combinations of the real expression values for different transcripts. Locas has a much higher runtime but produces more clusters with a better mean length, yet still far too few to produce correct expression estimates. Mira3's clusters are also close to the optimal length, but the program produces almost twice the number of clusters than Passage and its more generic approach to assembly is reflected in an extremely high runtime. These clusters have extremely vague consensus sequences with often more than 50% ambiguous bases ($r, y, s, w, k, m, b, v, d, h, n, *$) which again suggests that different clusters have been merged that should have stayed separate.

Furthermore, most assemblers sacrifice specificity (in the detection of overlaps) for speed, while Passage is guaranteed to correctly cluster all reads with $\leq k$ mismatches to the assumed genomic sequence. Passage finds about 4500 clusters less than Bowtie because we do mismatch resolution without a reference genome, sometimes leading to the fusion of two very small clusters from distinct genomic positions with almost identical sequence.

Validation with microarrays We chose two experiments to validate the expression values computed using our approach with microarray data (see table 1). These samples were analyzed using a custom microarray with 50-mer probes for all *C. dubliniensis* ORFs (febit, Heidelberg). Two

samples were analyzed using Passage and the resulting clusters were mapped to the *C. dubliniensis* genome using Bowtie. Of 5983 genes, 5144 (86%) genes could be analyzed. We only considered clusters with at least one read in each experiment, resulting in a list of 4377 (73.2%) genes. Fold-changes were computed independently for the microarray and the Passage data. First the fold-change between the two samples was computed for each cluster. The fold-changes of all clusters mapping to a common gene were averaged to obtain a fold-change value for each gene analyzed. The correlation between the fold-changes obtained from the microarray hybridizations and the Passage results was 0.69.

4 Discussion

We present a method for transcriptomic studies based on short RNA sequencing. It is especially useful in the absence of a reference genome. Reference sequences are only available for a tiny fraction of organisms, and while more and more genomes are sequenced, this still remains an issue for many research projects. Using a specialized protocol for the creation of the transcript pool, we greatly reduce the number of different read sequences and facilitate comparison between different samples. It effectively limits sequence overlaps to either complete or no overlap at all. Using this feature, our algorithm can rapidly cluster the reads and estimate expression for the corresponding transcripts in time linear to the number of read sequences.

A comparison to other tools shows that Passage is very fast and produces a sensible number of clusters, which allows to compute reliable expression estimates. We validated the expression levels computed using Passage with microarray data. Our method allows the application of well established software for comparative transcriptomics such as R [R D09] and Mayday [BSN10] to any (currently unsequenced) organisms and metatranscriptomic samples. If a genome sequence is available, Passage clusters can be mapped against that reference to elucidate genomic locations as well as assign the short cluster to longer (predicted) transcripts. Here, our algorithm also reduces the computational effort necessary for mapping, due to the great reduction in the number of sequences that need to be mapped, thus meeting one of the great challenges of NGS technologies.

Acknowledgements
We are grateful to Kai Sohn at the Fraunhofer IGB for developing the protocol and conducting the biological experiments. Part of this project was funded by a grant of the Baden-Württemberg Ministry of Science.

References

[BSN10] F. Battke, S. Symons, and K. Nieselt. Mayday - Integrative analytics for expression data. *BMC Bioinformatics*, 11(1):121, 2010.

[CWS99] B. Chevreux, T. Wetter, and S. Suhai. Genome Sequence Assembly Using Trace Signals and Additional Sequence Information. *Computer Science and Biology: Proceedings of the German Conference on Bioinformatics (GCB)*, 99:45–56, 1999.

[DBOSR08] F. De Bona, S. Ossowski, K. Schneeberger, and G. Rätsch. Optimal spliced alignments of short sequence reads. *BMC Bioinformatics*, 9(Suppl 10):O7, 2008.

[HM99] X. Huang and A. Madan. CAP3: A DNA sequence assembly program. *Genome Res*, 9(9):868–877, Sep 1999.

[Hü09] Hüttner, S. *Passage – Genexpressionsanalysen.* patent pending, AZ 102009058298.3, Dt. Patentamt München, 2009.

[KOS+10] J.D. Klein, S. Ossowski, K. Schneeberger, D. Weigel, and D.H. Huson. LOCAS – A low coverage assembly tool for resequencing projects. *(manuscript in preparation)*, 2010.

[Kur03] S. Kurtz. The Vmatch large scale sequence analysis software. *Ref Type: Computer Program*, pages 4–12, 2003.

[LRD08] H. Li, J. Ruan, and R. Durbin. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res*, 18(11):1851, 2008.

[LRR+10] E. Lindemann, B. Rohde, S. Rupp, J. Regenbogen, and K. Sohn. A multidimensional electrophoretic system of separation for the analysis of gene expression (MESSAGE). *Electrophoresis*, 31(8):1330–1343, 2010.

[LTPS09] B. Langmead, C. Trapnell, M. Pop, and S. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome biology*, 10(3):R25, 2009.

[LYL+09] R. Li, C. Yu, Y. Li, T.W. Lam, S.M. Yiu, K. Kristiansen, and J. Wang. SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics*, 25(15):1966, 2009.

[R D09] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009.

[SHO+09] K. Schneeberger, J. Hagmann, S. Ossowski, N. Warthmann, S. Gesing, O. Kohlbacher, and D. Weigel. Simultaneous alignment of short reads against multiple genomes. *Genome Biol*, 10:R98, 2009.

[TPS09] C. Trapnell, L. Pachter, and S.L. Salzberg. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, 25(9):1105, 2009.

[VZVK95] V.E. Velculescu, L. Zhang, B. Vogelstein, and K.W. Kinzler. Serial analysis of gene expression. *Science*, 270(5235):484, 1995.

[WER+09] D. Weese, A.K. Emde, T. Rausch, A. Döring, and K. Reinert. RazerS—fast read mapping with sensitivity control. *Genome Res*, 19(9):1646, 2009.

[WGS09] Z Wang, M Gerstein, and M Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet*, 10(1):57–63, Jan 2009.

[ZB08] D.R. Zerbino and E. Birney. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res*, 18(5):821, 2008.