The transcriptome of A.crassus: Evaluating a method of combining assemblies

Emanuel Heitlinger

1 Overview

The pre-processed *Anguillicola crassus* data-set consisting of 100491819 bases in 353055 reads (58617 generated using "FLX-chemistry", 294438 using "Titanium-chemistry") was assembled following an approach proposed by [1]: Two assemblies were generated, one using `newbler v2.6` [2], the other using `mira v3.2.1` [3]. The resulting assemblies (referred to as first-order assemblies) were merged with `Cap3` [4] into a combined assembly (referred to as second-order assembly).

2 The `newbler` first-order assembly

During transcriptome-assembly (with options `-cdna -urt`) `newbler` can split individual reads spanning the breakpoints of alternate isoforms, to assemble e.g. the first portion of the reads in one contig, the second portion in two different contigs. Later multiple so called isotigs would be constructed and reported, one for each putative transcript-variant. While this approach could be helpful for the detection of alternate isoforms, it also produces short contigs (especially at error-prone edges of high-coverage transcripts) when the building of isotigs fails. The read-status report and the assembly output in `ace`-format the program provides are including short contigs only used during the assembly-process, but not reported in the contigs-file used in transcriptome-assembly projects (`454Isotigs.fna`). Therefore to get all reads not included in contigs (i.e. a consistent definition of "singleton") it was necessary to add all reads appearing only in contigs not reported in the fasta-file to the reported singletons. The number of singletons increased in this step form the 26211 reported to 109052. We later also address the usefulness of `newbler's` report vs. the expanded singleton-category, but for the meantime we define singletons as all reads not present in a given assembly.

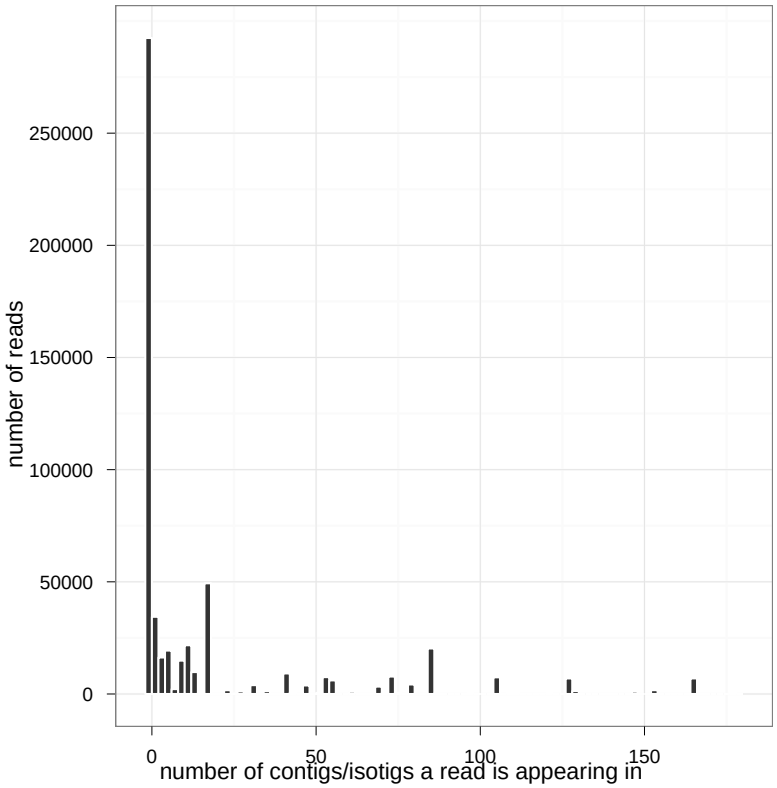

Figure 1: Number of contigs/isotigs newbler splitted one read into

As mentioned above, the splitting of reads in the `newbler` assembly can give useful information on possible isoforms. Figure 1 gives a histogram of the number of contigs/isotigs `newbler` splited a single read into. The number of read-splits (one read in more than 100 contigs) seems artificially inflated. If information would correspond to real isoforms it should be about an order of magnitude lower. The maximal number of read-splits in a given contig and it's usefulness will be discussed later in greater detail.

3 The `mira`-assembly and the second-order assembly

The `mira` assembly (with options `-job=denovo,est,accurate,454`) provided a second estimate of the transcriptome. In this assembly individual reads are not split. The number of reads not used in the `mira`-assembly was 65368.

To combine the two assemblies `cap3` was used with default parameters and including the quality information from first-order assemblies. The reminder of this text deals with the exploratory analysis of how information from both estimates of the transcriptome are integrated into the final second-order assembly.

	Newbler	Mira	Second-order(MN)
Max length	6300	6352	6377
Number of contigs	15934	22596	14064
Number of Bases	8085922	12010349	8139143
N50	579	579	662
Number of congtigs in N50	4301	6749	3899
non ATGC bases	375	29962	5245
Mean length	508	532	579

Table 1: Basic statistics for the first-order assemblies and the second-order assemly (for which only the most relieable category of contigs is shown efsec:data-categ-second)

Table 1 gives basic summary-statistics of the different assemblies. `mira` clearly produced the biggest assembly, both in terms of number of contigs and bases), the second-order assembly is slightly smaller size than the `newbler` assembly. The second-order assembly had on average longer contigs than both first-order assemblies and a higher weighted median contig size (N50).

4 Data-categories in the second-order assembly

Three main categories of assembled sequence data can be distinguished in the second-order assembly, each one with different reliability and purpose in downstream applications:

The first category of data obtained are the singletons of the final second-order assembly. It comprises raw sequencing reads that neither of the first-order assemblers used. It is therefore the intersecion of the `newbler`-singletons (as defined in 2) and the `mira`-singletons. 47669 reads fell in this category. A second category of sequence contains the first-order contigs, that could not be assembled in the second-order assembly (the singletons in the `cap3`-assembly; M_1 and N_1 in table 2). Furthermore second-order contigs in which first-order contigs from only one assembler are combined (M_n and N_n in table 2) also have to be included in this category. Sequences in this category should be considered only moderately reliable as they are supported by only one assembly algorithm.

Finally the category of contigs considered most reliable contains all second-order contigs with contribution from both first-order assemblies (MN in table 2).

	M_1	M_n	MN	N_n	N_1
Snd.o.con		164	13887	13	
Fst.o.con	2347	897	mira=19352/newbler=14410	40	1484
reads	42172	21153	one=269868/both=193308	1538	13100

Table 2: **Number of reads, first-order contigs (Fst.o.con) and second-order contigs (Snd.o.con) for different categories of contigs (M_1 and N_1 = first-order contigs not assembled in second-order assembly, from mira and newbler respectively; M_n and N_n = assembled in second-order contigs only with contigs from the same first-order assembly; MN = assembled in second-order contigs with first order contigs from both first order assemblies**

For the last, most reliable (MN) category, reads contained in the assembly can be categorized depending on whether they entered the assembly via both or only via one first-order assembly.

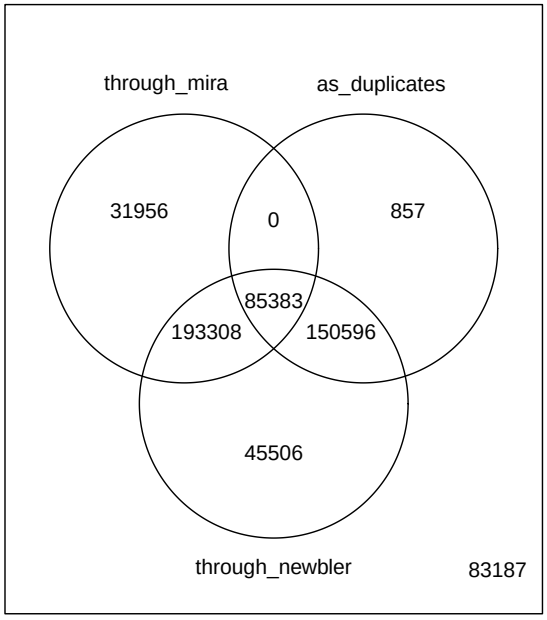

Figure 2: The way of reads into the most reliable (MN) assembly-category

Figure 2 gives a more detailed view of the fate of the reads `newbler` splited during first-order assembly. Interestingly most reads `newbler` splited ended in the high-quality category of the second order assembly.

5 Contribution of first-order assemblies to second-order contigs

Looking at the contribution of contigs from each of the assemblies to one second-order contig in figure 3 a it becomes clear, that the `mira`-assembly had a high number of redundant contigs. These were assembled into the same contig by `newbler` and finally also in one second-order contig by `Cap3`.

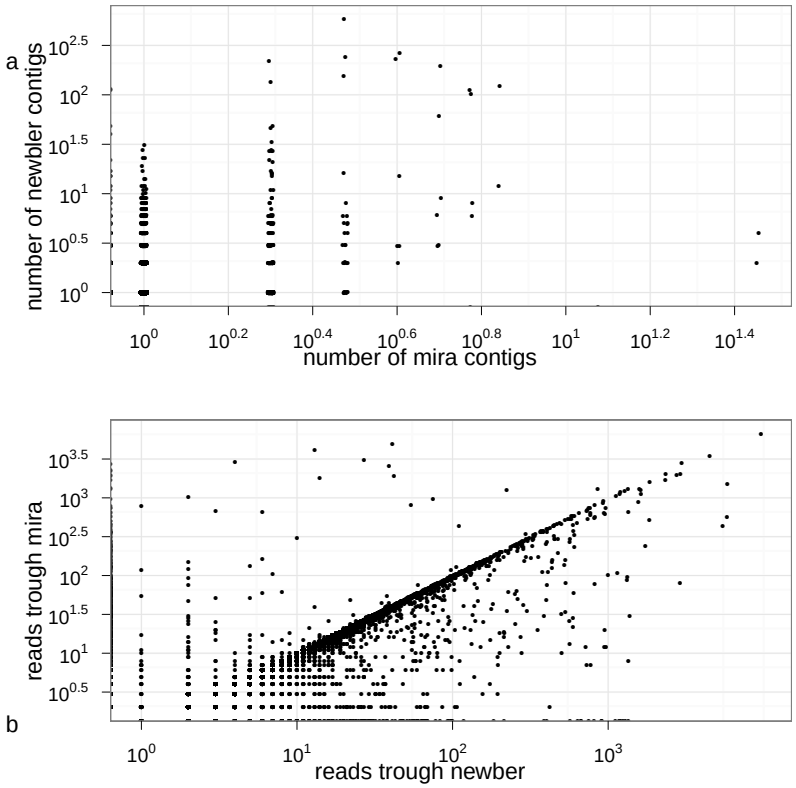

Figure 3: Number of first-order contigs from both first-order assemblies for each second order contig (a) number of reads through `newbler` and `mira` for each second-order contig (b)

A different picture emerges from the contribution of reads through each of the first-order assemblies (figure 3 b). Here for most second-order contigs many more reads are contributed through `newbler`-contigs. This is because `newbler` has more reads summed over all contigs caused by the duplication due to splitting of reads.

6 Evaluation of the assemblies

To further compare assemblies (`mira, newber` first-order assemblies including or excluding their singletons) and the second-order assembly (including different contigs-categories and singletons) we evaluated the number of bases or proteins their contigs and singletons (partially) cover in the related model-nematodes, *Caenorhabditis elegans* and *Brugia malayi*. To this purpose we used `blast` (`blastx` e-value cut-off 1e-5) and a custom perl-script provided by S. Kumar.

In addition, the size of the assembly can give an indication of redundancy or artificially assembled data: If it increases without improving the reference-coverage the dataset is likely to contain more redundant or artificial information, a more parsimonious assembly should be preferred.

Figure 4 gives the database-coverage in bases for the two reference species ploted against the size of the assembly-dataset in bases.

From the assemblies excluding singletons (in the lower left corner with lower size and database-coverage) the highly reliable contig-category of the second-order assembly produced the highest per-base coverage in both reference-species, with the `newbler` assembly on a second place and `mira` producing the lowest reference-coverage. When adding the contigs considered lower quality supported by only one assembler to the second-order assembly the reference-coverage increased moderately.

Including singletons the `mira` and `newber` assemblies were of increased size. A comparison of the `newbler's` reported singletons with all singletons addet to the `newbler`-assembly shows, that the reported singletons increased reference-coverage to the same amount than all singletons, while the non-reported singletons only increased the size of the assembly. It can be concluded, that the latter contain hardly any additional information but only error-prone or variant reads.

The second-order assembly including the intersection of first-order singletons performed similar to the `newbler` assembly for the number of bases coverd, but was larger in size. Adding the less reliable set of one-assembler supported second-order-contigs the assembly coverd the most bases in both references. When not the singleton of the second-order assembly (as defined in 2) but the intersection of `newbler's` "reported singletons" and `mira's` singletons were considered a very parsimonious assembly with high reference-coverage (termed fullest assembly; and labeled FU in the plots above) was obtained.

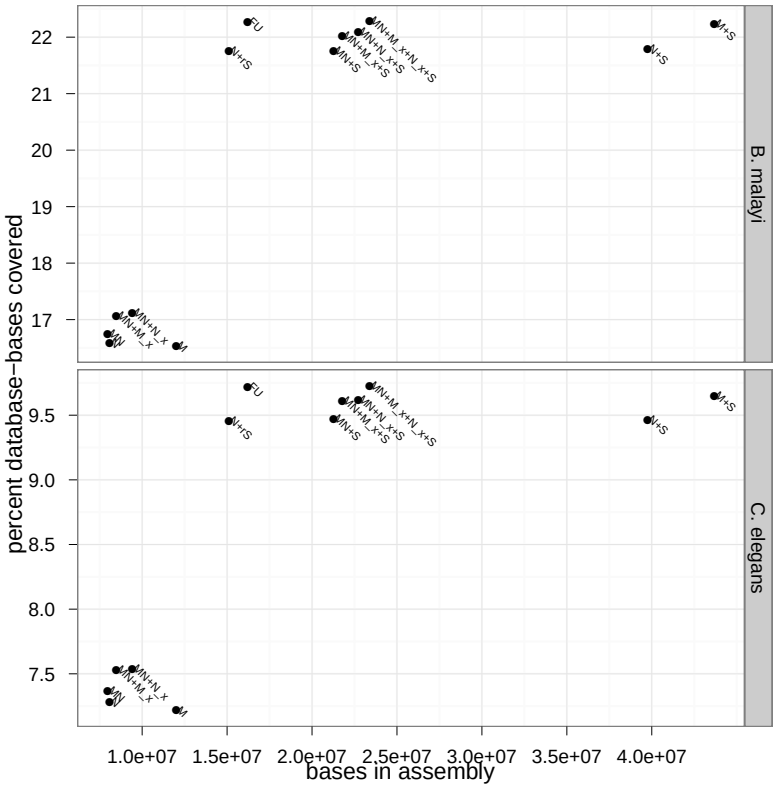

Figure 4: Base-content and reference-transcriptome coverage (in bases) for different assemblies and assembly-combinations (M = `mira`; N = `newbler`; $M + S$ = `mira` + singletons; $N + S$ = `newbler` plus singletons; $N + Sr$ = `newbler` plus singletons reported in readstatus.txt; MN = second-order contigs supported by both first-order; $MN + N\_x$ = second-order MN plus contigs only supported by `newbler`; $MN + M\_x$ = same for `mira`-first-order-contigs; $MN + M\_x + S$ and $MN + N\_x + S$ same with singletons; FU = second-order contigs supported by both or one assembler plus the intersection of `newbler` reported singletons and `mira`-singletons = the basis for the "fullest assembly" used in later analyses)

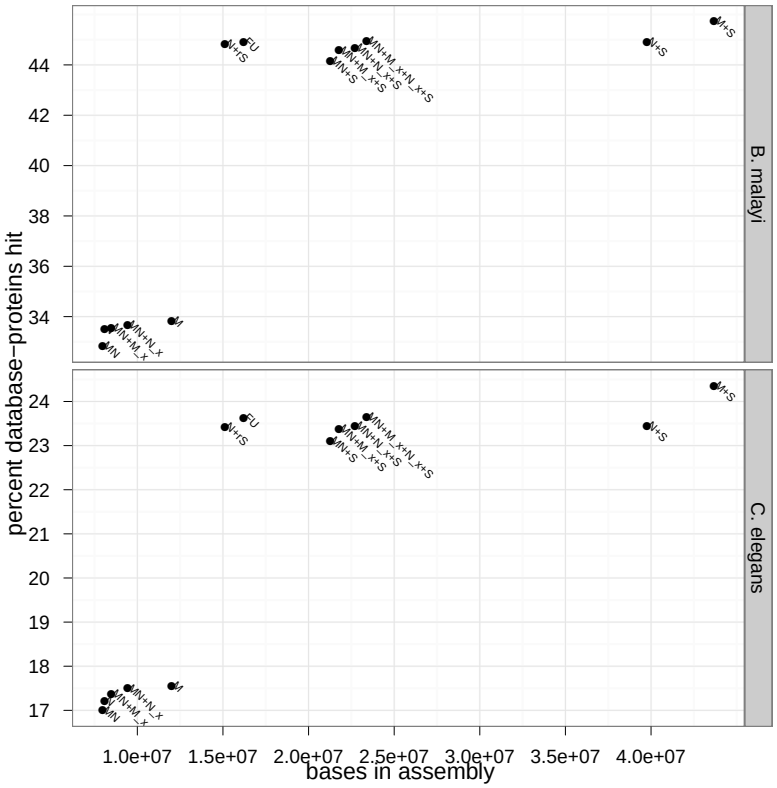

Figure 5: Base-content and reference-transcriptome coverage in percent of proteins hit for different assemblies and assembly-combinations (for category-abrevations see Figure 4)

Considering the reference-database with any kind of coverage the second-order assembly performed less preferable. Excluding singletons it was covering similar numbers of database-proteins than the `newber`-assembly and and was outperformed by the `mira`-assembly, although the latter showed again to be least parsimonious. The same general picture emerged from this analysis when singletons were considered additionally. `newbler` and second-order assemblies coverd similar amounts of reference-data.

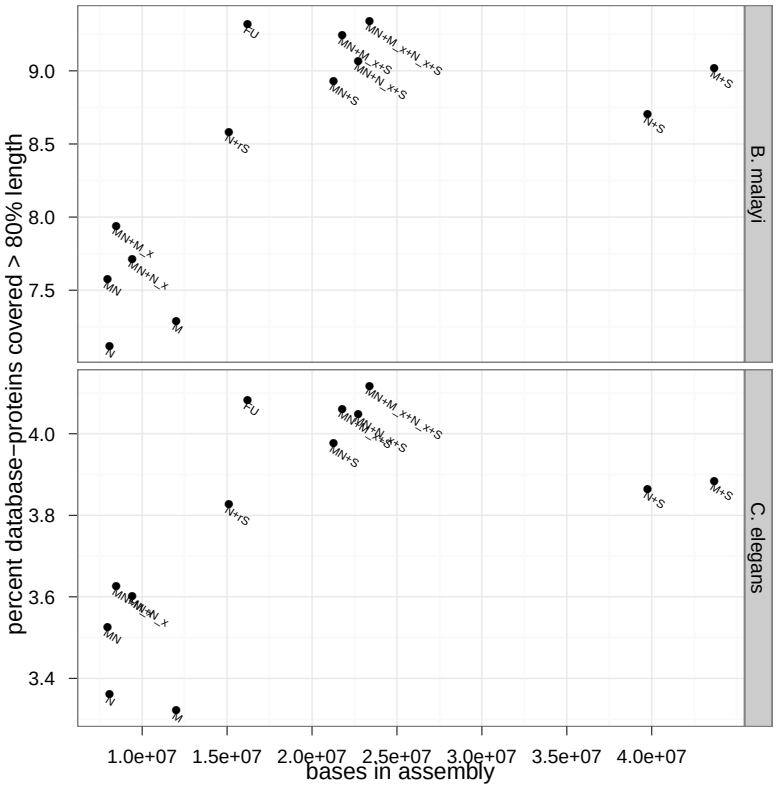

Figure 6: Base-content and reference-transcriptome coverage in percent of proteins coverd to at least 80% of their length for different assemblies and assembly-combinations (for category-abrevations see Figure 4)

When database-proteins covered for at least to 80% of their length are considered the second-order assembly showed it's superiority: Both ex- and including singletons the second-order assembly outperformed the first-order assemblies. Moderate gains in reference coverage were made again for the addition of dubious single-assembler supported second-order contigs. We give most weight in our analysis to these results as in average longer correct contigs will allow finding the highest number of putative full-lenth genes.

Given this evaluation we defined a "minimal adequate" assembly as the subset of contigs of the second-order assembly supported by both assemblers (labeled NM above).

Given the performance of the singletons `newbler` reported we defined a "fullest-assembly" as all second-order contigs (including those supported by only one assembler) plus the intersection of reported `newbler`-singletons and `mira` singletons.

7 Measurments on second-order assembly

Based on the following reads through the complicated assembly process, we calculated the following for each contig in the second-order assembly, to report it to for use in later analysis.

- number of `mira` and `newbler` first-order contigs
- number of reads through `mira` and reads through `newbler`
- number of reads being split by `newbler` in first-order assembly
- number of read-split events in the first-order assembly (equals the sum of reads multiplied by number of contigs a read has been split into)
- maximal number of first-order contigs a read in the contig has been split into during `newbler`-assembly
- the number of reads same-read-paires from the `newbler` and `mira` first order-assembly merged in a second order contig
- cluster-id of the contig: All contigs "connected" by sharing reads (similar to the graph clustering reported in [5]).
- number of other second order contigs containing the same read (size of the cluster)

7.1 Contig coverage

```
null device
          1

null device
          1
```

As well defined coverage-information is not readyly avaiable from the output of this combined assembly aproach (although we followed individual reads through the process) we inferred coverage by mapping the reads used for assembly against the fullest assembly using ssaha2 [6] with parameters (-kmer 13 -skip 3 -seeds 6 -score 100 -cmatch 10 -ckmer 6 -output sam -best 1). We converted the `sam`-ouput via a sorted `bam`-file to `pileup`-format using `samtools` [7].

For a second evalutation we excluded best-hits mapping to multiple contigs before converting the `sam`-file.

- mean per base coverage
- mean unique per base coverage

7.2 Example use of the contig-measurements

Based on these measurements the emergence of a given contig from the assembly process can be reconstructed. Table 3 gives an excerpt of the contig-measurements reported in additional-file `contig-data.csv`. The example contigs are all from large contig-clusters (cluster.size), where interpretation of the assembly history is complicated, but not impossible:

	Contig1047	Contig10719	Contig104	Contig13672
reads_through_Newbler	16	1351	0	14
reads_through_Mira	26	651	135	0
Newbler_contigs	1	5	0	2
Mira_contigs	1	9	4	0
category	MN	MN	M_n	N_n
num.new.split	8	1314	0	0
sum.new.split	16	2628	0	0
max.new.split	2	2	0	0
num.SndO.pair	13	644	0	0
cluster.id	CL62	CL6	CL176	CL235
cluster.size	24	18	5	5
coverage	4.200342	267.495458	41.003369	2.920755
uniq_coverage	4.248960	7.425507	2.568000	1.196078

Table 3: example table for assembly-measurements on contigs (as given in extttcontig-data.csv)

Contig1047 is in the well trusted MN category of contigs. It consists of only one contig from each first-order assembly (newbler_contigs and mira_contigs), each containing a set of reads of moderate size: 16 from `newbler` (reads_through_newbler) 26 from `mira` (reads_through_mira). 8 of the 16 reads `newbler` used in its one assembled contig were also assembled to a differnt `newbler`-contig (num.new.split). That each of the 8 reads was only appearing in one other `newbler`-contig is visible from the fact, that the number of split events is 16 (sum.new.split) and the maximal number of splits for one read is 2 (max.new.split). 13 (num.SndO.pair) same-read-pairs from the tow different first-order assemblies were merged in this second-order contig,

leaving 3 (16-13) reads in `newbler`-contigs and 13 (26-13) reads in `mira` contigs, which all could potentially have ended up in other contigs. The contig is in a cluster (CL62), which contains in total 24 contigs (cluster.size). It has to be addmitted that the whole graph-structure linking this 24 contigs can't be reconstructed from this contig summary data. On the other hand the summary data makes clear, from what source the links for cluster-affiliation have resulted: In this case from 3 and 13 unlinked read-paires from both first-order assemblies and 8 split-reads from `newbler`-fistr order contigs.

A comprehensive interpretation of the other example-contigs depicted is left to the reader. It should just be remarked, that in case of one-assembler supported contigs, all reads in that contig could potentially be represented in other contigs, making average cluster-size in these contigs bigger than in MN category.

One of the most interesting measurement calculated for each contig is the cluster-membership and cluster-size. Such clusters can represent close paralogs, duplicated genes, isoforms from alternative splicing or allelic variants.

These measurements can be used in later analysis to e.g. reevaluate the likelihood of misassembly in a given set of contigs. An evaluation of other cluster members, when biologically interesting properties are inferred for a contig in a cluster is e.g. advised and will be demonstrated in later in the manuscript.

8 Finalising the fullest assembly set

In order to minimize the amount of sequence with artificially inferred isoform-breakpoints we used the unique-mapping-information described above to detect contigs and singletons not supported by any raw data (reads). Table 4 gives a summary of these unsupported data by contig-category. For all downstream-analysis we removed the all well trusted MN-category-contigs no coverage and the contigs (and singletons) from other categories having no unique coverage.

	singletons	M_1	M_n	MN	N_1	N_n
coverage == 0	546	34	2	36	158	0
unique coverage == 0	584	48	2	42	210	3

Table 4: number of contigs with a coverage and unique-coverage of zero, inferred from mapping of raw reads, listed by contig-category

Thereby we reduced our dataset to 40187 tentative unique genes (TUGs), redifining the “fullest assembly” dataset.

References

[1] Kumar S, Blaxter ML: **Comparing de novo assemblers for 454 transcriptome data**. *BMC Genomics* 2010, **11**:571, [http://dx.doi.org/10.1186/1471-2164-11-571].

[2] Margulies M, Egholm M, Altman WE, Attiya S, Bader JS, Bemben LA, Berka J, Braverman MS, Chen YJ, Chen Z, Dewell SB, Du L, Fierro JM, Gomes XV, Godwin BC, He W, Helgesen S, Ho CH, Ho CH, Irzyk GP, Jando SC, Alenquer ML, Jarvie TP, Jirage KB, Kim JB, Knight JR, Lanza JR, Leamon JH, Lefkowitz SM, Lei M, Li J, Lohman KL, Lu H, Makhijani VB, McDade KE, McKenna MP, Myers EW, Nickerson E, Nobile JR, Plant R, Puc BP, Ronan MT, Roth GT, Sarkis GJ, Simons JF, Simpson JW, Srinivasan M, Tartaro KR, Tomasz A, Vogt KA, Volkmer GA, Wang SH, Wang Y, Weiner MP, Yu P, Begley RF, Rothberg JM: **Genome sequencing in microfabricated high-density picolitre reactors**. *Nature* 2005, **437**:376–380, [http://dx.doi.org/10.1038/nature03959].

[3] Chevreux B, Pfisterer T, Drescher B, Driesel AJ, Muller WE, Wetter T, Suhai S: **Using the miraEST assembler for reliable and automated mRNA transcript assembly and SNP detection in sequenced ESTs**. *Genome Res.* 2004, **14**:1147–1159, [http://www.ncbi.nlm.nih.gov/pmc/articles/PMC419793].

[4] Huang X, Madan A: **CAP3: A DNA sequence assembly program**. *Genome Res.* 1999, **9**:868–877, [http://www.ncbi.nlm.nih.gov/pmc/articles/PMC310812].

[5] Schwartz TS, Tae H, Yang Y, Mockaitis K, Van Hemert JL, Proulx SR, Choi JH, Bronikowski AM: **A garter snake transcriptome: pyrosequencing, de novo assembly, and sex-specific differences**. *BMC Genomics* 2010, **11**:694.

[6] Ning Z, Cox AJ, Mullikin JC: **SSAHA: a fast search method for large DNA databases**. *Genome Res.* 2001, **11**:1725–1729.

[7] Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis GR, Durbin R: **The Sequence Alignment/Map format and SAMtools.** *Bioinformatics* 2009, **25**(16):2078–2079.