Examining De Novo Transcriptome Assemblies via a Quality Assessment Pipeline

Noushin Ghaffari, Osama A. Arshad, Hyundoo Jeong, John Thiltges, Michael F. Criscitiello, Byung-Jun Yoon, Aniruddha Datta, Charles D. Johnson

Supplemental Document

This document presents the tools, settings, and extra material for the *de novo* transcriptome assembly, quality check and further annotations of Pacific whiteleg shrimp. Majority of the assemblies, evaluations and annotations ran at Pittsburgh Supercomputing Center (PSC). This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation. Furthermore, The ADA system at Texas A&M Supercomputing Facility, and supercomputers at The Brazos Cluster at Texas A&M University (TAMU) hosted our runs. We appreciate the support from aforementioned supercomputing centers.

Tools and settings

In our experiments, we used multiple software tools. In this section, we provide their settings, parameters and needed commands. In most cases we used the default settings, therefore, the used commands were straightforward and provided by the manual of the tool. For the cases that we used customized commands, we provided them in this section.

- Trinity
    - Run #1: software release r2013-02-25
    - Run #2: software release r2014-04-13
    - Run #3: software release r2014-07-17
    - All Trinity runs used the default settings of the software. We used the sample script provided by PSC. We made minor modification to the script to customize it for our

data. The awarded grant by NSF XSEDE covered our core hours at the PSC Blacklight system.

- SOAPdenovo-Trans release 1.03, default settings. Configurations:
    - max_rd_len=100, avg_ins=277, reverse_seq=0, asm_flags=3
- Trans-ABySS version 1.5.1, configurations:
    - K-mer 32 (default): --threads 16 --mpi 32
    - K-mer 48: --threads 16 --mpi 32
- The Oases version 0.2.08 and Velvet version 1.2.09, default settings. Configuration:
    - -ins_length 277
- Core Eukaryotic Genes Mapping Approach (CEGMA) pipeline version 2.5
- DEnovo TranscriptOme rNa-seq Assembly with or without the Truth Evaluation (DETONATE), the RSEM-EVAL version 1.6 part of the software suite is used.
- GMAP version 2014-08-04
- Tophat2 was used for mapping reads to the contigs.
    - We used Transrate (ver0.3.1), which automatically calls Bowtie2 (ver 2.2.3).
- CLC Genomics Workbench (CLC GWB), version 6.0.1.
    - The BLAST tool from the CLC GWB was used to search for sequences and homology similarities between our contigs and *Daphnia pulex* reference genome, running BlastX and BlastN.
    - The similarity parameters that used for BlastX and BlastN are:
        - BLASTX
            - Expectation value = 10.0
            - Word size = 3
            - Mask lower case = NO
            - Filter low complexity = YES
            - Maximum number of hits = 1
            - Protein matrix and gap costs = BLOSUM62_Existence_11_Extension_1
            - Number of threads = 1
            - Query genetic code = 1
        - BLASTN
            - Match/Mismatch and GapCosts= Match 1, Mismatch 3, Existence 5, Extension 2
            - Expectation value = 10
            - Word size = 11
            - Filter lower complexity =YES

            - Maximum number of hits = 1
            - Number of threads = 1
    - We exported the results into Microsoft Excel and applied two filters on the Blast E-values: 1E-4 and 1E-10
- Blast to UniProt/Swiss-Prot Databases BLAST+ version 2.2.29, the UniProt/Swiss-Prot databases released July 2014 as the reference
- Contig intersection: BLAST version 2.2.29+ to match similar contigs between assemblies. A BLAST database was made from each assembly, and BlastN compared each database/assembly pair. Hits with >= 90% identity were considered matches, and were reported.
- The JVENN program version v.1.5 to find the shared proteins among all six assemblies
    - http://bioinfo.genotoul.fr/jvenn/

The protein homology search

This section provides that commands and setting that were used for making Blast search against UniPro/Swiss-Prot databases.

Making BLAST database from UniProt/Swiss-Prot:

```
#!/bin/bash
module load blast
cd `dirname $0`
# https://github.com/johnmay/metingear/wiki/Tutorial-3:-Handling-Protein-Data
mkdir db
zcat uniprot/uniprot_sprot.fasta.gz | makeblastdb \
  -dbtype prot -out db/uniprot_sprot -title uniprot_sprot.fasta \
  -parse_seqids -hash_index
```

Running BLAST (using GNU Parallel to run multiple instances of BLAST in parallel, each with 100k chunks of the input):

```
#!/bin/bash
module load parallel
module load blast/2.2.29
set -ex
cd `dirname $0`
```

```
BLAST_ARGS="-num_threads 1 -max_target_seqs 1 -db db/uniprot_sprot -outfmt 6"
BLAST_CMD="blastx"
mkdir out
# https://www.biostars.org/p/63816/
# EXAMPLE: Blast on multiple machines
for i in ../data/*.fa ; do
   out=`basename $i .fa`
   date
   cat $i | parallel --block 100k --recstart '>' --pipe $BLAST_CMD $BLAST_ARGS -query - >
out/${out}.blast
   date
done
```

Summarize the number of contigs matching each UniProt sequence:

```
#!/bin/bash
cd `dirname $0`
for i in out/*.blast ; do
   cut -f1-2 $i | uniq | cut -f2 | sort | uniq -c | sort -g > ${i}.count
done
```

After the BlastX results for each assembler generated, we used the following steps to filter the insignificant hits, and found the common hits:

1. Filtering based on the E-value: hits with the value < 1E-4 were kept
    a. The R package CHNOSZ was used for eliminating the insignificant hits:
        i. Input_Blast <- read.blast(input_file_name_Path, evalue = 1e-4, similarity = 30)
2. The remaining hits were sorted and the number of times that each protein hit appeared were counted to calculate the protein-hit frequencies
    a. cut -d, -f3 assembly.blast_filtrd.csv| sort | uniq -c | sort -g > assembly.blast_filtrd _Sort_Unq.csv
3. The protein appearance frequencies were categorized as:
    - X >= 200
    - 100 =< X < 200
    - 50 =< X < 100
    - 10 =< X < 50

- X < 10

were X is the total number of hits.

4. In the last step, we intersected the SOAPdenovo-Trans hits that appeared more than 15 times, with all the assemblies. The goal was to find the most frequent hits that are shared among BlastX of the assembly outputs, using the following pseudo R commands:
   - input_file_name_Path <- read.csv(input_file_name_Path, header=F)
   - colnames(input_file_name_Path) = c("hits_frequncy","protein_ID")
   - input_file_name_Path_gte15 <- subset(input_file_name_Path, input_file_name_Path$hits_frequncy >= 15)
   - matched_SOAPdenov_input <- as.matrix(match(input_file_name_Path_SOAPdenovo_gte15$protein_ID,input_file_name_Path_gte15[,2]))

Resources

The different assemblers used during our experiments, required numerous core hours and considerable memory. In this section, we provide the resources used for generating our results.

SUPPLEMENTAL TABLE 1. COMPUTATIONAL AND MEMORY RESOURCES FOR TRINITY, TRANS-ABYSS, AND SOAPDENOVO-TRANS

Tool	Computation/ recourses	Memory usages (maximum)	Run time	Provider - Comments
Trinity, run #1 (r2013-02-25)	96 CPUs	221,878 Mb	~113Hrs	PSC Blacklight
Trinity, run #2 (r2014-04-13)	96 CPUs	942,691 Mb	~120Hrs	PSC Blacklight
Trinity, run #3 (r2014-07-17)	96 CPUs	574,198 Mb	~75Hrs	PSC Blacklight
Trans-ABySS, k-mer 32	43 Max process	69,068 Mb	~10Hrs	ADA, TAMU Supercomputing
Trans-ABySS, k-mer 48	43 Max process	66,685 Mb	~ 10 Hrs	ADA, TAMU Supercomputing
SOAPdenovo-Trans	4 Max process	132,078 Mb	~3Hrs	ADA, TAMU Supercomputing

Finding common *Daphnia pulex* annotated contigs

We also compared the protein homologies with *D. pulex,* as a reference*,* for transcriptome assembly results. The results from each BlastX search were intersected with the other corresponding sets to find the shared proteins.

As the following Tables show there is a very high degree of concordance in the homologous proteins found in *D. pulex* from all the assemblers, with more than 7,200 proteins shared among all six assemblies (E-value < 1E-10). The Venn diagrams are drawn from the unique and identical protein significant hits using the JVENN program. The JVENN program can find the overlaps among all six outputs of our study.

SUPPLEMENTAL TABLE 2. COMPARISON OF DAPHNIA PULEX PROTEIN HOMOLOGS FOUND BY BLASTX SEARCH FROM EACH ASSEMBLY FOR DIFFERENT E-VALUE THRESHOLD

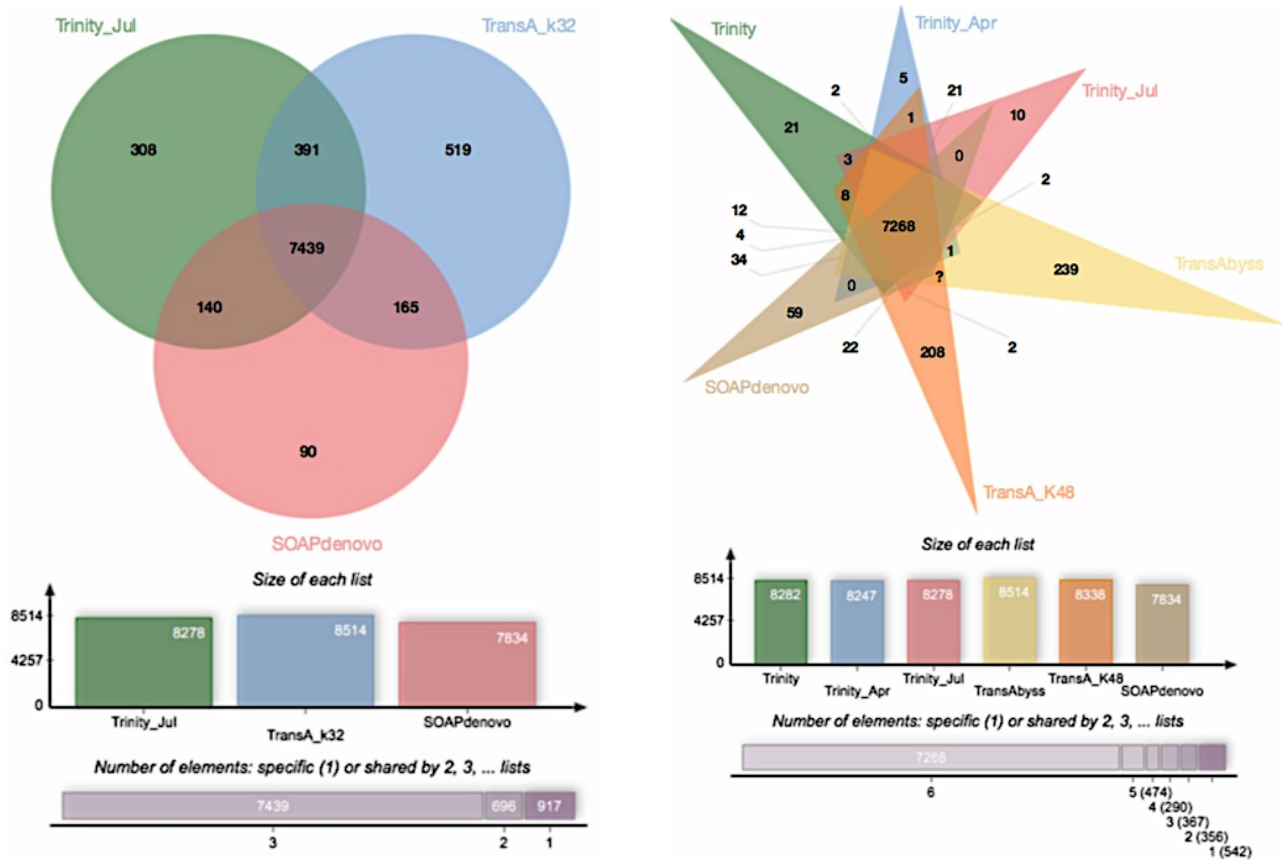
A) E-VALUE THRESHOLD: 1E-4

	SOAPdenovo-Trans	Trans-ABySS_Run1_kmer32	Trans-ABySS_Run2_kmer48	Trinity_Run1_rFeb13	Trinity_Run2_rApr14	Trinity_Run3_rJul14
SOAPdenovo-Trans	**18,136**	7,604	7,506	7,614	7,575	7,579
Trans-ABySS_Run1_kmer32		**33,725**	7,830	7,844	7,809	7,830
Trans-ABySS_Run2_kmer48			**35,074**	7,717	7,682	7,701
Trinity_Run1_rFeb13				**30,534**	8,139	8,151
Trinity_Run2_rApr14					**24,766**	8,205
Trinity_Run3_rJul14						**25,161**

B) E-VALUE THRESHOLD: 1E-10

	SOAPdenovo-Trans	Trans-ABySS_Run1_kmer32	Trans-ABySS_Run2_kmer48	Trinity_Run1_rFeb13	Trinity_Run2_rApr14	Trinity_Run3_rJul14
SOAPdenovo-Trans	**7,242**	7,072	6,972	7,801	7,034	7,032
Trans-ABySS_Run1_kmer32		**7,760**	7,375	7,287	7,228	7,246
Trans-ABySS_Run2_kmer48			**7,605**	7,167	7,113	7,126
Trinity_Run1_rFeb13				**7,609**	7,479	7,486
Trinity_Run2_rApr14					**7,542**	7,501
Trinity_Run3_rJul14						**7,570**

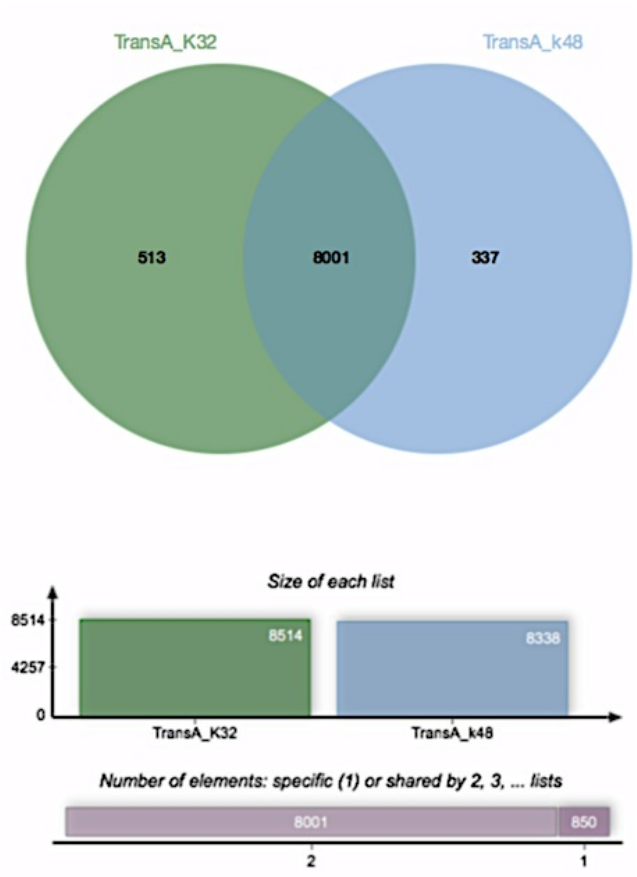
Supplemental Figures 1 and 2 show the Venn diagrams for interesting various assembler's protein hits. The Venn diagrams are drawn from the unique and identical protein significant hits using the JVENN program. The diagrams show the results for E-value filtration 1E-10. The majority of protein hits are shared among all assembled transcripts. In order to examine the effect of different runs of the same program, we draw Venn diagram for two Trans-ABySS runs, and the three Trinity runs. As supplemental Figure 2 shows, runs overlap very well.



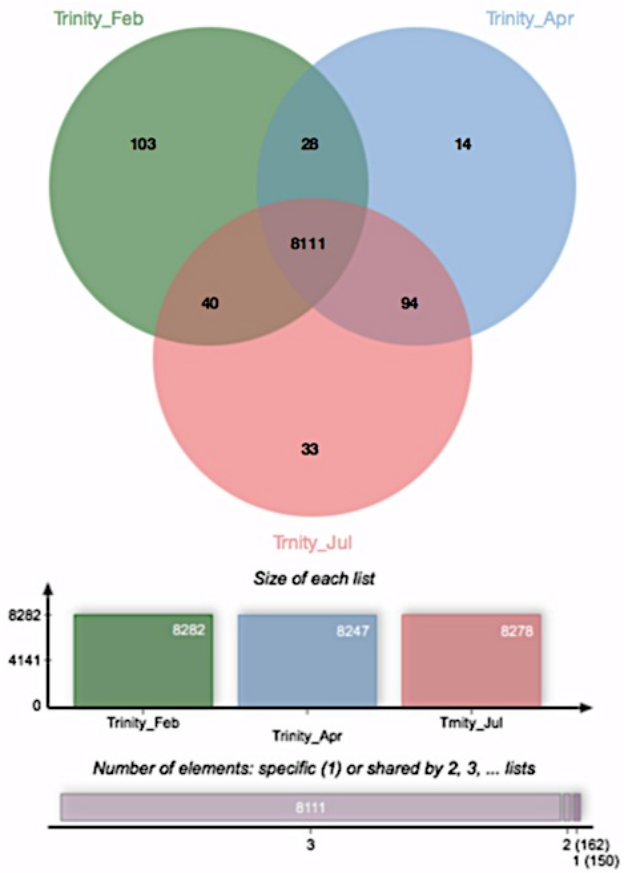
A) PROTEIN HITS SHARED AMONG TRINITY_RUN3_RJUL14, SOAPDENOVO-TRANS, AND TRANS-ABYSS_RUN1_KMER32

B) PROTEIN HITS SHARED AMONG ALL SIX ASSEMBLIES

Supplemental Fig. 1. Comparison of Daphnia pulex protein homologs found by BLASTX search from three (A) and six (B) assemblies.



A) PROTEIN HITS SHARED BETWEEN TRANS-ABYSS_RUN1_KMER32 AND TRANS-ABYSS_RUN2_KMER48



B) PROTEIN HITS SHARED AMONG ALL THREE TRINITY RUNS

Supplemental Fig. 2. Comparison of Daphnia pulex protein homologs found by BLASTX search from two (A) and three (B) assemblies.

Contig Intersection

This section provides the details of intersecting contigs, and our results. In order to find the shared portion of the transcripts with any other assembly's contigs, we followed these steps:

- For each assembly
    - Make BLAST database (source database)
- For each assembly
    - With each BLAST database
        - Run blastn (-max_target_seqs 1 -outfmt 6)

        - Source Database versus Target Database
- For each output file
    - Ignore hits less than 90% identity (column 3)
    - Count the number of unique contigs

This process results in finding the percentage of contigs that each assembly shares with others.

Following Supplemental Table presents the sharing percentages.

SUPPLEMENTAL TABLE 3. BLAST COMPARISONS TO FIND % OF SHARED TRANSCRIPTS AMONG ASSEMBLIES

Blast Queries, Source Database vs. Target Database	Contigs Found in Target BLAST Database (>=90% identity)	Total Source Contigs	% Shared
SOAPdenovo-Trans vs. SOAPdenovo-Trans	62350	62514	99.74%
SOAPdenovo-Trans vs. Trans-ABySS_Run1_kmer32	55444		88.69%
SOAPdenovo-Trans vs. Trans-ABySS_Run2_kmer48	48633		77.80%
SOAPdenovo-Trans vs. Trinity_Run2_rApr14	60542		96.85%
SOAPdenovo-Trans vs. Trinity_Run1_rFeb13	60828		97.30%
SOAPdenovo-Trans vs. Trinity_Run3_rJul14	60629		96.98%
Trans-ABySS_Run1_kmer32 vs. SOAPdenovo-Trans	111853	119772	93.39%
Trans-ABySS_Run1_kmer32 vs. Trans-ABySS_Run1_kmer32	119146		99.48%
Trans-ABySS_Run1_kmer32 vs. Trans-ABySS_Run2_kmer48	109246		91.21%
Trans-ABySS_Run1_kmer32 vs. Trinity_Run2_rApr14	116103		96.94%
Trans-ABySS_Run1_kmer32 vs. Trinity_Run1_rFeb13	116603		97.35%
Trans-ABySS_Run1_kmer32 vs. Trinity_Run3_rJul14	116042		96.89%
TransABySS_Run2_kmer48 vs. SOAPdenovo-Trans	103515	110556	93.63%
TransABySS_Run2_kmer48 vs. Trans-ABySS_Run1_kmer32	108085		97.76%
TransABySS_Run2_kmer48 vs. Trans-ABySS_Run2_kmer48	109353		98.91%
TransABySS_Run2_kmer48 vs. Trinity_Run2_rApr14	106502		96.33%
TransABySS_Run2_kmer48 vs. Trinity_Run1_rFeb13	106925		96.72%
TransABySS_Run2_kmer48 vs. Trinity_Run3_rJul14	106418		96.26%
Trinity_Run2_rApr14 vs. SOAPdenovo-Trans	75831	102093	74.28%
Trinity_Run2_rApr14 vs. Trans-ABySS_Run1_kmer32	74528		73.00%
Trinity_Run2_rApr14 vs. Trans-ABySS_Run2_kmer48	65189		63.85%
Trinity_Run2_rApr14 vs. Trinity_Run2_rApr14	101603		99.52%

Trinity_Run2_rApr14 vs. Trinity_Run1_rFeb13	99911		97.86%
Trinity_Run2_rApr14 vs. Trinity_Run3_rJul14	101349		99.27%
Trinity_Run1_rFeb13 vs. SOAPdenovo-Trans	84231	110474	76.25%
Trinity_Run1_rFeb13 vs. Trans-ABySS_Run1_kmer32	83115		75.23%
Trinity_Run1_rFeb13 vs. Trans-ABySS_Run2_kmer48	73804		66.81%
Trinity_Run1_rFeb13 vs. Trinity_Run2_rApr14	108000		97.76%
Trinity_Run1_rFeb13 vs. Trinity_Run1_rFeb13	110015		99.58%
Trinity_Run1_rFeb13 vs. Trinity_Run3_rJul14	108077		97.83%
Trinity_Run3_rJul14 vs. SOAPdenovo-Trans	77390	103773	74.58%
Trinity_Run3_rJul14 vs. Trans-ABySS_Run1_kmer32	76217		73.45%
Trinity_Run3_rJul14 vs. Trans-ABySS_Run2_kmer48	66860		64.43%
Trinity_Run3_rJul14 vs. Trinity_Run2_rApr14	102807		99.07%
Trinity_Run3_rJul14 vs. Trinity_Run1_rFeb13	101512		97.82%
Trinity_Run3_rJul14 vs. Trinity_Run3_rJul14	103288		99.53%

Furthermore, we BLAST searched the contig sections for each assembly against pairs of other assemblers. Following Supplemental Table presents the percentage of shared contigs.

SUPPLEMENTAL TABLE 4. BLAST COMPARISONS TO FIND % OF SHARED TRANSCRIPTS AMONG EACH ASSEMBLY AND PAIRS OF OTHER ASSEMBLIES

Blast Queries, Source Database vs. Pairs of Target Databases	Contigs Found in Target BLAST Databases (>=90% identity)	Total Source Contigs	% Shared
SOAPdenovo-Trans vs. both Trans-ABySS_Run1_kmer32 and TransABySS_Run2_kmer48	48159	62514	77.04%
SOAPdenovo-Trans vs. both Trans-ABySS_Run1_kmer32 and Trinity_Run2_rApr14	54541		87.25%
SOAPdenovo-Trans vs. both Trans-ABySS_Run1_kmer32 and Trinity_Run1_rFeb13	54820		87.69%
SOAPdenovo-Trans vs. both Trans-ABySS_Run1_kmer32 and Trinity_Run3_rJul14	54620		87.37%
SOAPdenovo-Trans vs. both TransABySS_Run2_kmer48 and Trinity_Run2_rApr14	47917		76.65%

SOAPdenovo-Trans vs. both TransABySS_Run2_kmer48 and Trinity_Run1_rFeb13	48152		77.03%
SOAPdenovo-Trans vs. both TransABySS_Run2_kmer48 and Trinity_Run3_rJul14	47980		76.75%
SOAPdenovo-Trans vs. both Trinity_Run2_rApr14 and Trinity_Run1_rFeb13	60274		96.42%
SOAPdenovo-Trans vs. both Trinity_Run2_rApr14 and Trinity_Run3_rJul14	60341		96.52%
SOAPdenovo-Trans vs. both Trinity_Run1_rFeb13 and Trinity_Run3_rJul14	60355		96.55%
Trans-ABySS_Run1_kmer32 vs. both SOAPdenovo-Trans and TransABySS_Run2_kmer48	104532	119772	87.28%
Trans-ABySS_Run1_kmer32 vs. both SOAPdenovo-Trans and Trinity_Run2_rApr14	110484		92.25%
Trans-ABySS_Run1_kmer32 vs. both SOAPdenovo-Trans and Trinity_Run1_rFeb13	110929		92.62%
Trans-ABySS_Run1_kmer32 vs. both SOAPdenovo-Trans and Trinity_Run3_rJul14	110371		92.15%
Trans-ABySS_Run1_kmer32 vs. both TransABySS_Run2_kmer48 and Trinity_Run2_rApr14	106976		89.32%
Trans-ABySS_Run1_kmer32 vs. both TransABySS_Run2_kmer48 and Trinity_Run1_rFeb13	107444		89.71%
Trans-ABySS_Run1_kmer32 vs. both TransABySS_Run2_kmer48 and Trinity_Run3_rJul14	106902		89.25%
Trans-ABySS_Run1_kmer32 vs. both Trinity_Run2_rApr14 and Trinity_Run1_rFeb13	115437		96.38%
Trans-ABySS_Run1_kmer32 vs. both Trinity_Run2_rApr14 and Trinity_Run3_rJul14	115371		96.33%
Trans-ABySS_Run1_kmer32 vs. both Trinity_Run1_rFeb13 and Trinity_Run3_rJul14	115342		96.30%
TransABySS_Run2_kmer48 vs. both SOAPdenovo-Trans and Trans-ABySS_Run1_kmer32	102989	110556	93.16%
TransABySS_Run2_kmer48 vs. both SOAPdenovo-Trans and Trinity_Run2_rApr14	102396		92.62%
TransABySS_Run2_kmer48 vs. both SOAPdenovo-Trans and Trinity_Run1_rFeb13	102802		92.99%
TransABySS_Run2_kmer48 vs. both SOAPdenovo-Trans and Trinity_Run3_rJul14	102287		92.52%
TransABySS_Run2_kmer48 vs. both Trans-ABySS_Run1_kmer32 and Trinity_Run2_rApr14	105791		95.69%
TransABySS_Run2_kmer48 vs. both Trans-ABySS_Run1_kmer32 and Trinity_Run1_rFeb13	106214		96.07%
TransABySS_Run2_kmer48 vs. both Trans-ABySS_Run1_kmer32 and Trinity_Run3_rJul14	105701		95.61%

TransABySS_Run2_kmer48 vs. both Trinity_Run2_rApr14 and Trinity_Run1_rFeb13	105880		95.77%
TransABySS_Run2_kmer48 vs. both Trinity_Run2_rApr14 and Trinity_Run3_rJul14	105802		95.70%
TransABySS_Run2_kmer48 vs. both Trinity_Run1_rFeb13 and Trinity_Run3_rJul14	105788		95.69%
Trinity_Run2_rApr14 vs. both SOAPdenovo-Trans and Trans-ABySS_Run1_kmer32	69153	102093	67.74%
Trinity_Run2_rApr14 vs. both SOAPdenovo-Trans and TransABySS_Run2_kmer48	62032		60.76%
Trinity_Run2_rApr14 vs. both SOAPdenovo-Trans and Trinity_Run1_rFeb13	75353		73.81%
Trinity_Run2_rApr14 vs. both SOAPdenovo-Trans and Trinity_Run3_rJul14	75709		74.16%
Trinity_Run2_rApr14 vs. both Trans-ABySS_Run1_kmer32 and TransABySS_Run2_kmer48	64366		63.05%
Trinity_Run2_rApr14 vs. both Trans-ABySS_Run1_kmer32 and Trinity_Run1_rFeb13	73859		72.34%
Trinity_Run2_rApr14 vs. both Trans-ABySS_Run1_kmer32 and Trinity_Run3_rJul14	74382		72.86%
Trinity_Run2_rApr14 vs. both TransABySS_Run2_kmer48 and Trinity_Run1_rFeb13	64615		63.29%
Trinity_Run2_rApr14 vs. both TransABySS_Run2_kmer48 and Trinity_Run3_rJul14	65049		63.72%
Trinity_Run2_rApr14 vs. both Trinity_Run1_rFeb13 and Trinity_Run3_rJul14	99709		97.66%
Trinity_Run1_rFeb13 vs. both SOAPdenovo-Trans and Trans-ABySS_Run1_kmer32	77616	110474	70.26%
Trinity_Run1_rFeb13 vs. both SOAPdenovo-Trans and TransABySS_Run2_kmer48	70505		63.82%
Trinity_Run1_rFeb13 vs. both SOAPdenovo-Trans and Trinity_Run2_rApr14	83424		75.51%
Trinity_Run1_rFeb13 vs. both SOAPdenovo-Trans and Trinity_Run3_rJul14	83502		75.59%
Trinity_Run1_rFeb13 vs. both Trans-ABySS_Run1_kmer32 and TransABySS_Run2_kmer48	72989		66.07%
Trinity_Run1_rFeb13 vs. both Trans-ABySS_Run1_kmer32 and Trinity_Run2_rApr14	82090		74.31%
Trinity_Run1_rFeb13 vs. both Trans-ABySS_Run1_kmer32 and Trinity_Run3_rJul14	82205		74.41%
Trinity_Run1_rFeb13 vs. both TransABySS_Run2_kmer48 and Trinity_Run2_rApr14	72903		65.99%
Trinity_Run1_rFeb13 vs. both TransABySS_Run2_kmer48 and Trinity_Run3_rJul14	73019		66.10%

Trinity_Run1_rFeb13 vs. both Trinity_Run2_rApr14 and Trinity_Run3_rJul14	107638		97.43%
Trinity_Run3_rJul14 vs. both SOAPdenovo-Trans and Trans-ABySS_Run1_kmer32	70716	103773	68.14%
Trinity_Run3_rJul14 vs. both SOAPdenovo-Trans and TransABySS_Run2_kmer48	63581		61.27%
Trinity_Run3_rJul14 vs. both SOAPdenovo-Trans and Trinity_Run2_rApr14	77075		74.27%
Trinity_Run3_rJul14 vs. both SOAPdenovo-Trans and Trinity_Run1_rFeb13	76887		74.09%
Trinity_Run3_rJul14 vs. both Trans-ABySS_Run1_kmer32 and TransABySS_Run2_kmer48	66032		63.63%
Trinity_Run3_rJul14 vs. both Trans-ABySS_Run1_kmer32 and Trinity_Run2_rApr14	75830		73.07%
Trinity_Run3_rJul14 vs. both Trans-ABySS_Run1_kmer32 and Trinity_Run1_rFeb13	75494		72.75%
Trinity_Run3_rJul14 vs. both TransABySS_Run2_kmer48 and Trinity_Run2_rApr14	66482		64.06%
Trinity_Run3_rJul14 vs. both TransABySS_Run2_kmer48 and Trinity_Run1_rFeb13	66232		63.82%
Trinity_Run3_rJul14 vs. both Trinity_Run2_rApr14 and Trinity_Run1_rFeb13	101133		97.46%