**ZANFONA User Manual**

**Zanfona is a command-line genome assembly finishing program.**

ZANFONA is designed to work with single-organism Eukaryotic preassemblies constructed from paired short-end Illumina reads. Short reads normally yield fragmented assemblies for organisms with larger genomes. Aligning to a reference genome of a related species often results in a chromosome structure and gene arrangement that mirrors the reference, even when those structures do not exist in the species under study. ZANFONA improves a fragmented assembly by comparison with reference genomes from related species without mirroring large-scale structure.

The reference genome used does not have to be chromosome-level, and can also be a fragmented pre-assembly. For studies sequencing a number of related organisms, the assemblies for each species can serve as a reference for the others simultaneously. ZANFONA works with an iterative approach, with each subsequent run on a reference set improving the target assembly.

Starting with raw, paired-end Illumina data:

1. **Trim low quality sequence** and adapter/primers with Trimmomatic or other similar programs. Trimmomatic is available here:

<https://github.com/timflutre/trimmomatic>

1. **Generate a pre-assembly** using any of the many available programs. The authors use Spades but any similar program will be suitable.

Spades is available here:

<https://github.com/ablab/spades>

1. **Download Zanfona:**

[**https://github.com/zanfona734/zanfona**](https://github.com/zanfona734/zanfona)

calculate\_joins.pl

calculate\_gap\_lengths.pl

join\_contigs.pl

user\_manual.docx (this file)

1. **Set up reference genome set**:

Reference genomes can be fragmented pre-assemblies from related species, or public genomes downloaded from Genbank:

<https://www.ncbi.nlm.nih.gov/assembly/>

Note: Multiple reference genomes can be combined into one reference set, and this same set can be used for multiple Alembic iterations. However, the larger the number of genomes used in the same reference set will increase the time spent on the Blast step. The authors use six references at once. Users may increase or decrease that number according to their needs and resources.

Concatenate reference genomes into a single fasta file. When using Genbank genome assemblies, it is recommended to leave the headers as they are downloaded. The information in the header will be recorded in the log file to allow tracking of evidence for each join if desired. For references not yet in Genbank, users may want to add the organism name to the fasta headers for this reason.

1. **Chop reference genome:**

The reference genome is chopped into fragments of 1000 nt. This number was optimized for vertebrates but has been tested for invertebrates and plants. Note: the step requires BBmap. Depending on how BBMap was installed you may need to add a path to allow the program to run properly.

*shred.sh in=reference.fasta out=reference\_chopped.fasta length=1000 minlength=1000 overlap=0 &*

1. **Make BlastDB from target genome**

Construct a Blast database from the single genome being assembled (target genome). This must be done one genome at a time, do not concatenate the targets.

Name your target re-assembly “target.fasta”, or change the name in the command and in subsequent steps.

*makeblastdb -in target.fasta -dbtype nucl -out target*

The Blast suite, including makeblastdb, can be found here:

<https://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/>

1. **Blast reference to target:**

blastn -query reference\_chopped.fasta -db target -out out.blastn -num\_descriptions 2 -num\_alignments 2 -outfmt 7 &

The resulting output will be in this format:

# BLASTN 2.11.0+

# Query: NC\_052102.1 Zerene\_cesonia\_1.1, whole genome shotgun sequence\_16000-16999

# Database: target

# Fields: query acc.ver, subject acc.ver, % identity, alignment length, mismatches, gap opens, q. start, q. end, s. start, s. end, eva

lue, bit score

# 2 hits found

NC\_052102.1 312796 83.133 83 11 3 187 269 176 97 3.49e-11 73.1

NC\_052102.1 40846 85.294 68 9 1 191 258 904 970 4.52e-10 69.4

1. **Determine order and orientation of preassembly contigs**

Make calculate\_joins.pl executable:

chmod a+x calculate\_joins.pl

Format Blast output: Note: make sure the Blast step is actually completed before starting this. Blast will output data as it is running and this process may take hours depending on the size of the reference set.

cat out.blastn | ./calculate\_joins.pl > output.csv

The output will look like this:

query,contig,min-start,max-start,min-stop,max-stop,orientation

NW\_024045153.1,3153,4513,4513,4783,4783,forward

NW\_024045153.1,304605,3,3,112,112,forward

NW\_024045160.1,303732,14,14,79,79,forward

NW\_024045160.1,411789,161,161,188,188,forward

NW\_024045180.1,37676,871,871,740,740,reverse

NW\_024045180.1,5559,194,194,325,325,forward

…

Note there are exactly two lines per query, column 1. Column 2 is the pair of contigs to be joined. Columns 3-6 show the regions of the query and target and are used for calculating order, overlap and gap lengths in the second step. Column 7 is the orientation the contig will be in the joined pair.

1. **Calculate join overlaps and gap lengths**

Make calculate\_gap\_lengths.pl executable:

*chmod a+x calculate\_gap\_lengths.pl*

Run:

*cat output.csv | ./calculate\_gap\_lengths.pl > zanfona\_joins.csv*

The output looks like this:

NW\_024045153.1,304605,forward,3153,forward,-4401

NW\_024045160.1,303732,forward,411789,forward,82

NW\_024045180.1,5559,forward,37676,reverse,-546

NW\_024045257.1,7117,reverse,1246,reverse,-5526

NW\_024045329.1,5559,forward,37676,reverse,-483

NW\_024045336.1,141959,reverse,113775,reverse,212

NW\_024045340.1,44912,forward,11471,reverse,-3735

Note: this is the working table used for making joins. It is presented as one line per join.

Column 1: query name.

Columns 2-5: coordinates and orientation.

Column 6: length of the gap (positive values) or the overlap (negative values).

1. Assemble joins

This step makes the joins using ./join\_contigs.pl.

cat zanfona\_joins.csv | ./join\_contigs.pl &

The final output file, target2.fasta, will be the completed genome with joins made, with calculated overlap and join lengths.

Note: this may take a considerable amount of time to run. For some users it may be desirable to do this in stages (ex: 1000 joins per run) to avoid exceeding allowed run times per job.

1. Repeat

Repeat steps 4-6 with a different reference genome until the resulting joined genome has the desired N50 value, or there are no additional joins that can be made. Remember to rename target2.fasta to target.fasta before starting the next round.

Statistics (such as N50) can be monitored by using BBMap’s stats.pl program.

stats.sh target.fasta > stats.out