**Code Availability For:**

Long-read based genome assembly of *Drosophila gunungcola* reveals less chemosensory genes in flower-breeding species

Ateesha Negi1, Ben-Yang Liao2, andShu-Dan Yeh1\*

1 Department of Life Sciences, National Central University, Taiwan (ROC),   
2 Division of Biostatistics and Bioinformatics, National Health Research Institutes, Taiwan (ROC).

\* Correspondence: [shudanyeh@ncu.edu.tw](mailto:shudanyeh@ncu.edu.tw)

**Detailed computational pipeline for analyzing and annotating the *Drosophila gunungcola* genome.**

**Table of contents:**

1. *De novo* Assembly
2. Genome Assembly Assessment
3. Haplotypes and contaminated contigs detection
4. Genome Annotation
5. Mitogenome Assembly and Annotation
6. Phylogenetic analysis:
7. Finding ORs and IRs

**System used:**

Server: High Performance Cluster by T3 Taiwan: [t3-c4.nchc.org.tw](http://t3-c4.nchc.org.tw/) and NCHC iservice:

<https://iservice.nchc.org.tw>

MAC OS: macOS Big Sur v11.6.2; 2 GHz Quad-Core Intel Core i5 ; 16 GB 3733 MHz LPDDR4X; Intel Iris Plus Graphics 1536 MB.

Linux System: X86\_64 (Ubuntu 20.04); 2.9 GHz Quad-Core Intel Core i7-10700; 16MB Cache; LA1200.

**A). *De novo* Assembly:**

1) SequelTools v1.3.0 (Hufnagel et al., 2020) (<https://github.com/ISUgenomics/SequelTools> ) pipeline was used to access the quality of the raw reads and filtering the reads.

Usage: default pipeline with recommended inputs

# ran on server through PBS  
# performing raw read QC   
# input type = subreads.txt and scraps.txt containing BAM file of raw reads and scrap reads

SequelTools.sh -t Q

-u subreads.txt

-c scraps.txt

-n 20

# performing raw read filtering

SequelTools.sh -t F

-u subreads.txt

-c scraps.txt

-C -P -N -Z 1000

-n 20

2) Canu v1.8 (Koren et al., 2017)

(<https://github.com/marbl/canu>) was used for *de novo* assembling of filtered reads.

Usage: recommended parameters

# ran on server through PBS  
# input type = filtered read

canu useGrid=0

maxThreads=20

maxMemory=90g

genomeSize=170m

correctedErrorRate=0.085

corMhapSensitivity=normal

-pacbio-corrected = /filtered\_read.fasta

-minReadLength = 7,000

-stopOnReadQuality = 0

-minOverlapLength = 1,000

3) Falcon v1.1.5 (Chin et al., 2016) ( <https://github.com/PacificBiosciences/FALCON>) was used for *de novo* assembling of filtered reads.

Usage: recommended parameters

# ran on server through PBS  
# input type = filtered read fasta  
  
length\_cutoff = 9000 (Length cutoff used for seed reads used for initial mapping)  
  
length\_cutoff\_pr = 12111 (Length cutoff used for seed reads used for pre-assembly)

pa\_HPCdaligner\_option = -v -dal128 -t20 -e.70 -l1000 -s1000

ovlp\_HPCdaligner\_option = -v -dal128 -t32 -h60 -e.96 -l500 -s1000

pa\_DBsplit\_option = -x500 -s400

ovlp\_DBsplit\_option = -x500 -s400

falcon\_sense\_option = --output\_multi --min\_idt 0.70 --min\_cov 4 --local\_match\_count\_threshold 2 --max\_n\_read 200 --n\_core 20 --output\_dformat

overlap\_filtering\_setting = --max\_diff 100 --max\_cov 100 --min\_cov 1 --bestn 10 --n\_core 20

4) MaSuRCA v3.2.8 (Zimin et al., 2013) (<ftp://ftp.genome.umd.edu/pub/MaSuRCA/>) was used for *de novo* assembling of filtered reads.

Usage: default settings with recommended parameters

# ran on server through PBS

# input type = filtered read /PacBio: filtered\_read.fasta

# input type = short reads / illumine: Dgun\_SK\_trimmed\_R.fastq Dgun\_SK\_trimmed\_L.fastq

# Configration file main changes:

DATA is specified as type {PE and PACBIO}

GRAPH\_KMER\_SIZE = 0

GRID\_BATCH\_SIZE=300000000

LHE\_COVERAGE=40

MEGA\_READS\_ONE\_PASS=0

JF\_SIZE = 10200000000

5) Pilon v1.22 (Walker et al., 2014) (<https://github.com/broadinstitute/pilon> ) used for polishing MaSURCA assembly.

Usage: recommended pipeline

# ran on server through PBS

# input type = filtered read; illumine reads and assembled genome: Aligned and indexed BAM files and illumine reads (via Bowtie2 <http://bowtie-bio.sourceforge.net/bowtie2/index.shtml>)

--diploid

6) Arrow v2.3.3 (Chin et al., 2013) (<https://github.com/PacificBiosciences/pbbioconda>) used for polishing Canu, Falcon and Quickmerge assemblies.

Usage: recommended pipeline

# ran on server through PBS

# input type = filtered read and assembled genome; aligned BAM files (via pbmm2 <https://github.com/PacificBiosciences/pbmm2>) used for polishing with GenomicConsensus

7) Quickmerge v0.2 (Chakraborty et al., 2016) (<https://github.com/mahulchak/quickmerge> )

Usage: recommended pipeline and parameters

# ran on server through PBS

# input type = assembled genome fasta

-hco 5.0 -c 1.5 -l 400000 -ml 20000

**B) Genome Assembly Assessment:**

8) QUAST v5.02 (Gurevich et al., 2013) (<http://bioinf.spbau.ru/quast>) used for assembly statistics assessment.

Usage: default settings and parameters

# ran on server through PBS

# input type = assembled genome fasta and illumine pair end reads

9) BUSCO v4.0.5 (Simão et al., 2015) ( <http://busco.ezlab.org> ) used for genome assesment

Usage: default settings with recommended parameters

#ran on server

#for genome completeness

-m genome -l diptera\_odb10

#for transcriptome completeness

-m transcriptome -c 40 --augustus\_species fly --augustus\_parameters='--progress=true'

#for protein completeness

-m proteins -c 40 --augustus\_species fly --augustus\_parameters='--progress=true'

#for identifying contaminated contigs

-m genome -l bacteria\_odb10

10) IGV v2.11.1 (Thorvaldsdóttir et al., 2013) (<https://software.broadinstitute.org/software/igv/> ) used for visualization of genome alignments. Pre required files type were attained from pbmm2 (<https://github.com/PacificBiosciences/pbmm2>) and Samtools v1.15 (Li et al., 2009) (<https://github.com/samtools/> ).

Usage: default pipeline

#ran on Linux system

#input type = filtered read and assembled and indexed and aligned genome files “fasta, .fai, .bam and .bai”

11) SyMAP v5.0.6 (Soderlund et al., 2011) (<http://www.agcol.arizona.edu/software/symap/> ) used to perform and visualization of genome alignments

Usage: default settings

#ran on Mac OS

#input type = assembled genome fasta and MAKER annotation GFFs

13) SYRI v1.5 (Goel et al., 2019) (<https://github.com/schneebergerlab/syri> ) to identify structural variation

Usage: default pipeline

#ran on Mac OS

#input type = assembled genome fasta

13) MUMmer v3.1 (Kurtz et al., 2004) (<http://mummer.sourceforge.net/> ) used for alignments

Usage: default pipeline

#ran on Mac OS and server

#input type = assembled genome fasta as query

#nucumer parameters  
--maxmatch -c 100 -b 500 -l 50

**C) Haplotypes and contaminated contigs detection:**

First, we used BUSCO with bacterial lineage “-l bacterial” to identify contigs of bacterial origin.   
Second, we used BLASTn to identify contigs of *Drosophila* gut microbiome origin

Finialy, we used Kraken2 to identify contigs of other contaminants origin in our whole-genome assembly of *Drosophila gunungcola.* After decontamination, contigs were identified as haplotypes using Purge\_haplotigs.

14) BLASTn (Altschul et al., 1997) (<https://blast.ncbi.nlm.nih.gov/Blast.cgi>)

Usage: default  
#ran on Mac OS and server

#input type = assembled genome fasta as query and Drosophila gut microbiome as subjects.

15) Kraken2 v2.0.9-beta (Wood et al., 2019) (<https://github.com/DerrickWood/kraken2>)

Usage: default  
#ran on server

#input type = assembled genome fasta

#downloaded standard taxonomy from K2 database

#kraken2 database = K2\_Standard\_20200919

#classification of contigs

kraken2-build --standard --threads 40 –db K2\_Standard\_20200919

16) Purge\_haplotigs v1.1.1 (Roach et al., 2018) (<https://bitbucket.org/mroachawri/purge_haplotigs/src/master/>)

Aligned via minimap2 v2.24 (Li, 2018) (<https://github.com/lh3/minimap2> ) and sort via samtools v1.15 (Li et al., 2009) (<https://github.com/samtools/> ).

Usage: default pipeline

#ran on server

#input type assembled genome fasta and filtered subreads fasta

#Step 0 (to get align .bam file via minimap 2 and smtools ‘.sam’ to align ‘.bam’ of the assembled genome)

minimap2 -t 16 -ax map-pb --secondary=no

samtools sort -m 40G

# step 1

purge\_haplotigs hist -t 20

#step 2 :the vale was selected from the graph as recommended

purge\_haplotigs cov -l 6 -m 35 -h 95

#step 3

purge\_haplotigs purge -t 20

**D) Genome Annotation:**

20) FastQC v0.11.9 (Andrews, 2010) (<https://www.bioinformatics.babraham.ac.uk/projects/fastqc/> ) used for quality assessment of short reads.

Usage: default

#ran on MAC OS

# input type = fastq files of pair end reads

21) Fastp v0.19.1 (Shifu et al., 2018) was used for trimming short reads.

Usage: default

#ran on server  
# input type = fastq files of pair end reads

21) Trinity v2.8.5 (B. J. Haas et al., 2013) (<https://github.com/trinityrnaseq/trinityrnaseq> ) for assembling short reads

#ran on server  
# input type = fastq files of pair end reads

Trinity --seqType fq --CPU 40 --max\_memory 180G --verbose --jaccard\_clip

22) Repeat Masker v4.1.0 (Chen, 2004) (<https://www.repeatmasker.org/> ) for masking of repeats in the genome.

Usage: default/recommended pipeline

#ran on server  
# input type = assembled primary decontaminated genome fasta

# The query species was selected to be drosophila

RepeatMasker Combined Database: Dfam\_3.1 and RepBase-20181026. Run with rmblastn version 2.9.0+

RepeatMasker -xsmall --species drosophila

ProcessRepeats -species drosophila -gff -excln

22) MAKER v3.01.03 (Cantarel et al., 2008) (<https://www.yandell-lab.org/software/maker.html> ) used for genome annotation. We ran five rounds of MAKER to predict gene and protein models out of which 3rd predicted the best protein models.

1) First round of gene prediction was performed directly from transcript (generated using Trinity) and protein evidence (nine *Drosophila* relative species downloaded from flybase (<https://flybase.org/>).

MAKER was optimized and ran following the tutorials:

1)<http://weatherby.genetics.utah.edu/MAKER/wiki/index.php/MAKER_Tutorial_for_WGS_Assembly_and_Annotation_Winter_School_2018#Add_functional_annotations_and_meta-data>

2) <https://gist.github.com/darencard/bb1001ac1532dd4225b030cf0cd61ce2>

Usage: recommended pipeline

#ran on server

#input modified control files

mpiexec -n 40 path\_to/maker\_opts.ctl path\_to/maker\_bopts.ctl path\_to/maker\_exe.ctl

Round 1

#Step 1: To generate three control files: maker\_opts.ctl, maker\_exe.ctl, maker\_bopts.ctl

maker -CTL

#Step 2: Modify to provide required data types maker\_opts.ctl

#-----Genome (these are always required)

genome= /path\_to/Dgun\_SK.fasta

organism\_type=eukaryotic

#-----Re-annotation Using MAKER Derived GFF3

### Kept default

#-----EST Evidence (for best results provide a file for at least one)

est=/path\_to/assembled\_mRNA-seq\_trinity.fasta

#set of ESTs or assembled mRNA-seq in fasta format

### Kept default

#-----Protein Homology Evidence (for best results provide a file for at least one)

protein=/path\_to/related\_9Drosophila\_species\_protein.faa

### rest kept default

#-----Repeat Masking (leave values blank to skip repeat masking)

model\_org=drosophila

repeat\_protein=/path\_to/maker/data/te\_proteins.fasta

rm\_gff=/path\_to/Dgun\_SK.fasta.complex.reformat.gff3

prok\_rm=0

softmask=1   
### rest kept default

#-----Gene Prediction

est2genome=1 #infer gene predictions directly from ESTs, 1 = yes, 0 = no

protein2genome=1 #infer predictions from protein homology, 1 = yes, 0 = no

### rest kept default

#-----Other Annotation Feature Types (features MAKER doesn't recognize)

### kept default

#-----External Application Behavior Options

### kept default

#-----MAKER Behavior Options

### kept default

#Step 3: setting up of blast mapping parameter in maker\_bopts.ctl.file

### kept default

#Step 4: modifying maker\_bopts.ctl.file

### setting path/location for maker executables

#step 5: run MAKER round 1

#step 6: The output “rnd1.maker.output.log” of the first round was used to extract the EST, protein and repeat in an individual .gff files, which were provided as input files for this second round. SNAP and AUGUSTUS (via BUSCO) training were also done. For this step the following tutorial instruction/steps was followed: <https://gist.github.com/darencard/bb1001ac1532dd4225b030cf0cd61ce2>

Round 2:

#step 7: modifying maker\_opts.ctl for 2nd round

#-----Genome (these are always required)

genome=/path\_to/Dgun\_SK.fasta

organism\_type=eukaryotic

#-----Re-annotation Using MAKER Derived GFF3

### kept default

#-----EST Evidence (for best results provide a file for at least one)

est\_gff=/path\_to/rnd1.all.maker.est2genome.gff

### rest kept default

#-----Protein Homology Evidence (for best results provide a file for at least one)

protein\_gff=/path\_to/rnd1.all.maker.protein2genome.gff

### rest kept default

#-----Repeat Masking (leave values blank to skip repeat masking)

rm\_gff=/path\_to/rnd1.all.maker.repeats.gff

prok\_rm=0

softmask=1   
### rest kept default

#-----Gene Prediction

snaphmm=/path\_to/snap/round1/NCT\_rnd1.zff.length50\_aed0.25.hmm

augustus\_species=Dgunungcola\_r1

est2genome=1

protein2genome=1

trna=1

unmask=0   
### rest kept default

#-----Other Annotation Feature Types (features MAKER doesn't recognize)

### kept default

#-----External Application Behavior Options

### kept default

#-----MAKER Behavior Options

### kept default

Next (3,4,5) Rounds:

#Itteration of step 6 and 7 was done.

17) TransDecoder v5.5.0 (B. Haas & Papanicolaou, 2016) ( <https://github.com/TransDecoder/TransDecoder> ) to find coding regions within predicated transcripts.

Usage: default  
#ran on server

#input type = MAKER predicted transcript fasta

TransDecoder.LongOrfs -t

18) OrthoFinder v2.2.7 (Emms & Kelly, 2019) (<https://github.com/davidemms/OrthoFinder> )To find orthologs of predicted proteins.

Usage: default  
#ran on server

#input type = DATA file containing MAKER predicted protein fasta and relative species fasta

orthofinder -f /path\_to/DATA

**E) Mitochondrion genome Annotation:**

To identify mitochondrion genome first MUMmer v3.1 (Kurtz et al., 2004) (<http://mummer.sourceforge.net/> ) used for alignments with *D.melanogaster* mitochondrion as a reference was done. Then the selected contig was blasted online using BLASTn (Altschul et al., 1997) (<https://blast.ncbi.nlm.nih.gov/Blast.cgi>)

18) MITOS-2 (Donath et al., 2019) (<http://mitos2.bioinf.uni-leipzig.de/index.py> ) online mito genome annotation tool

Usage: recommended settings

# ran online  
#Reference= RefSeq89 Metazoa

#Genetic Code= 5 Invertebrate

# input type mitochondria fasta

19) SnapGene Viewer v5.3.2 (Insightful Science) ( <https://www.snapgene.com> )

Usage: default

# ran on MAC OS

# input type= mitochondria fasta

19) MEGA v11.08-1 (Tamura et al., 2021) (<https://www.megasoftware.net/>)

Usage: default

# ran on MAC OS

# input type= mitochondria fasta

**F) Phylogenetic analysis:**

From predicted protein sequences:

Protein seq was used to create (Figure 1) phylogenetic rooted tree via OrthoFinder v2.2.7 and visualizes online on NCBI tree viewer (Federhen, 2012) ( <https://www.ncbi.nlm.nih.gov/tools/treeviewer/>).

From mitochondria DNA sequences:

Mitochondrion (all protein-coding) sequences of *D. gunungcola* and 17 species of *Drosophila* were aligned via opting for clustalW alignment and the tree was plotted via opting maximum likelihood through MEGA v11.08-1 (Tamura et al., 2021)

**G. Finding ORs and IRs**

20) HMMER (v.3.3.2;<http://hmmer.org>, last accessed Spetmber 2022) (Eddy 2011)

- *hmmsearch*  <path to the protein seq ( annotated from *D. gunungcola*) > <path to Pfam database .hmm file>

21) PseudoPipe (Zhang et al., 2006) <http://pseudogene.org/pseudopipe/>

#ran on server

Followed the author's instructions.

Exons were extracted using Perl script:

#!/usr/bin/perl -w

open (FILE, $ARGV[0]);

while ($line=<FILE>) {

@bk=split(/\t/, $line);

if ($bk[2] eq 'exon') {

$chr=$bk[0];

$exon{$chr}.="$bk[0]\t$bk[2]\t$bk[3]\t$bk[4]\n";

}

}

close FILE;

foreach my $key (sort { $a <=> $b } %exon) {

$fileName="chr".$key."\_exLocs";

#print "$fileName\n";

open (OUT, ">$fileName");

print OUT "$exon{$key}";

close OUT;

}

**References:**

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research, 25*(17), 3389-3402.

Andrews, S. (2010). FastQC: a quality control tool for high throughput sequence data. In: Babraham Bioinformatics, Babraham Institute, Cambridge, United Kingdom.

Cantarel, B. L., Korf, I., Robb, S. M., Parra, G., Ross, E., Moore, B., et al. (2008). MAKER: an easy-to-use annotation pipeline designed for emerging model organism genomes. *Genome research, 18*(1), 188-196.

Chakraborty, M., Baldwin-Brown, J. G., Long, A. D., & Emerson, J. J. (2016). Contiguous and accurate de novo assembly of metazoan genomes with modest long read coverage. *Nucleic Acids Research, 44*(19), e147-e147. <https://doi.org/10.1093/nar/gkw654>

Chen, N. (2004). Using Repeat Masker to identify repetitive elements in genomic sequences. *Current protocols in bioinformatics, 5*(1), 4.10. 11-14.10. 14.

Chen, S., Zhou, Y., Chen, Y., & Gu, J. (2018). fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics, 34(17), i884-i890.

Chin, C.-S., Alexander, D. H., Marks, P., Klammer, A. A., Drake, J., Heiner, C., et al. (2013). Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nature methods, 10*(6), 563-569.

Chin, C.-S., Peluso, P., Sedlazeck, F. J., Nattestad, M., Concepcion, G. T., Clum, A., et al. (2016). Phased diploid genome assembly with single-molecule real-time sequencing. *Nature methods, 13*(12), 1050-1054.

Donath, A., Jühling, F., Al-Arab, M., Bernhart, S. H., Reinhardt, F., Stadler, P. F., et al. (2019). Improved annotation of protein-coding genes boundaries in metazoan mitochondrial genomes. *Nucleic Acids Research, 47*(20), 10543-10552. <https://doi.org/10.1093/nar/gkz833>

Eddy SR. 2011. Accelerated Profile HMM Searches. PLoS Comput Biol 7:e1002195.

Emms, D. M., & Kelly, S. (2019). OrthoFinder: phylogenetic orthology inference for comparative genomics. *Genome biology, 20*(1), 1-14.

Federhen, S. (2012). The NCBI taxonomy database. *Nucleic Acids Research, 40*(D1), D136-D143.

Goel, M., Sun, H., Jiao, W.-B., & Schneeberger, K. (2019). SyRI: finding genomic rearrangements and local sequence differences from whole-genome assemblies. *Genome biology, 20*(1), 1-13.

Gurevich, A., Saveliev, V., Vyahhi, N., & Tesler, G. (2013). QUAST: quality assessment tool for genome assemblies. *Bioinformatics, 29*(8), 1072-1075. <https://doi.org/10.1093/bioinformatics/btt086>

Haas, B., & Papanicolaou, A. (2016). TransDecoder (find coding regions within transcripts). *GitHub 3.1*.

Haas, B. J., Papanicolaou, A., Yassour, M., Grabherr, M., Blood, P. D., Bowden, J., et al. (2013). De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. *Nature protocols, 8*(8), 1494-1512.

Hufnagel, D. E., Hufford, M. B., & Seetharam, A. S. (2020). SequelTools: a suite of tools for working with PacBio Sequel raw sequence data. *BMC Bioinformatics, 21*(1), 429. <https://doi.org/10.1186/s12859-020-03751-8>

Koren, S., Walenz, B. P., Berlin, K., Miller, J. R., Bergman, N. H., & Phillippy, A. M. (2017). Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome research, 27*(5), 722-736.

Kurtz, S., Phillippy, A., Delcher, A. L., Smoot, M., Shumway, M., Antonescu, C., & Salzberg, S. L. (2004). Versatile and open software for comparing large genomes. *Genome biology, 5*(2), 1-9.

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics, 34*(18), 3094-3100.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., et al. (2009). The sequence alignment/map format and SAMtools. *Bioinformatics, 25*(16), 2078-2079.

Roach, M. J., Schmidt, S. A., & Borneman, A. R. (2018). Purge Haplotigs: allelic contig reassignment for third-gen diploid genome assemblies. *BMC Bioinformatics, 19*(1), 1-10.

Simão, F. A., Waterhouse, R. M., Ioannidis, P., Kriventseva, E. V., & Zdobnov, E. M. (2015). BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics, 31*(19), 3210-3212. <https://doi.org/10.1093/bioinformatics/btv351>

Soderlund, C., Bomhoff, M., & Nelson, W. M. (2011). SyMAP v3.4: a turnkey synteny system with application to plant genomes. *Nucleic Acids Research, 39*(10), e68-e68. <https://doi.org/10.1093/nar/gkr123>

Tamura, K., Stecher, G., & Kumar, S. (2021). MEGA11: Molecular Evolutionary Genetics Analysis Version 11. *Molecular Biology and Evolution, 38*(7), 3022-3027. <https://doi.org/10.1093/molbev/msab120>

Thorvaldsdóttir, H., Robinson, J. T., & Mesirov, J. P. (2013). Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Briefings in bioinformatics, 14*(2), 178-192.

Walker, B. J., Abeel, T., Shea, T., Priest, M., Abouelliel, A., Sakthikumar, S., et al. (2014). Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. *PloS one, 9*(11), e112963.

Wood, D. E., Lu, J., & Langmead, B. (2019). Improved metagenomic analysis with Kraken 2. *Genome biology, 20*(1), 1-13.

Zhang, Z., Carriero, N., Zheng, D., Karro, J., Harrison, P. M., & Gerstein, M. (2006). PseudoPipe: an automated pseudogene identification pipeline. Bioinformatics, 22(12), 1437-1439.

Zimin, A. V., Marçais, G., Puiu, D., Roberts, M., Salzberg, S. L., & Yorke, J. A. (2013). The MaSuRCA genome assembler. *Bioinformatics, 29*(21), 2669-2677.