# Assembly process description

**#Long-read assembly pipelines**

The single-molecule sequencing (SMS) data are assembled following a hierarchical approach: (a) select a subset of longer reads as seed data and correct through canu/falcon;(b) use the error-corrected reads for a draft assembly through different assemblers; (c)polish the draft assembly through Quiver/Arrow and Pilon.

**##Different assemblers**

**Canu** is a comprehensive and scalable pipeline for SMS data assembly (available at https://github.com/marbl/canu, v1.5). In the correction step, Canu first selects longer seed reads with the settings 'genomeSize=430000000' and 'corOutCoverage=80', then detects raw reads overlapping through a high sensitive overlapper MHAP (mhap-2.1.2, option 'corMhapSensitivity=low/normal/high'), and finally performs an error correction through falcon_sense method (option 'correctedErrorRate=0.025'). In the next step, with the default parameters, error-corrected reads are trimmed unsupperted bases and hairpin adapters to get their longest supported range. In the last step, Canu generates the draft assembly using longest 80 coverage trimmed reads.
Reference: Koren S, et al. "Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation." Genome research 27.5(2017):722.

**Falcon** is a hierarchical and haplotype aware genome assembler (available at https://github.com/PacificBiosciences/FALCON, v0.3.0). In the correction step, Falcon first selects longer seed reads with the setting 'length_cutoff = 3000', then detects raw reads overlapping through daligner overlapper (pa_HPCdaligner_option, '-v -B128 -e.70 -l4800 -s100 -k18 -w8 -h480 -M8 -T4'), and finally performs an error correction through falcon_sense method (falcon_sense_option, '--output_multi --min_idt 0.70 --min_cov 3 --max_n_read 200 --n_core 4'). In the assembly step, Falcon selects pre-assembly reads with the setting 'length_cutoff_pr = 8000', then detects reads overlapping (ovlp_HPCdaligner_option, '-v -B128 -e.96 -l2400 -s100 -k18 -h1024 -M8 -T4'), and finally performs a directed string graph with the setting 'overlap_filtering_setting = --max_diff 120 --max_cov 120 --min_cov 3 --n_core 4 --bestn 8'.
Reference: Chin, Chen Shan, et al. "Phased Diploid Genome Assembly with Single Molecule Real-Time Sequencing." Nature Methods 13.12(2016):1050.

**Wtdbg** is a SMS data assembler by constructing fuzzy Brujin graph (available at https://github.com/ruanjue/wtdbg). Wtdbg first generates a draft assembly with the command 'wtdbg -i pbreads.fasta -t 64 -H -k 21 -S 1.02 -e 3 -o wtdbg'. Using error-corrected reads from canu gets a better assembly performance. And then a consensus assembly is obtained with the command 'wtdbg-cns -t 64 -i wtdbg.ctg.lay -o wtdbg.ctg.lay.fa -k 15'.

Reference: https://github.com/ruanjue/wtdbg

**##Merge assemblies**

**Quickmerge** merges different assemblies to produce a more contiguous assembly (available at https://github.com/mahulchak/quickmerge). Quickmerge uses contigs from canu as query input, and contigs from wtdbg as ref input. The two contigs are aligned through mummer (v4.0.0, available at https://github.com/mummer4/mummer) with the nucmer parameters '-b 500 -c 100 -l 200 -t 12' and delta-filter parameters '-i 90 -r -q', and then merged through quickmerge with the parameters '-hco 5.0 -c 1.5 -l 100000 -ml 5000'.

Reference: Chakraborty, M, et al. "Contiguous and accurate de novo assembly of metazoan genomes with modest long read coverage." Nucleic Acids Research 44.19(2016):e147-e147.

**##Polish assembly**

The draft assembly is polished to obtain the final assembly. The first round polishing adopts quiver/arrow algorithm using SMS data with the 40 threads. The second polishing adopts pilon algorithm (v1.22, available at https://github.com/broadinstitute/pilon) using illumina data with the parameters '--mindepth 10 --changes --threads 4 --fix bases'.

**Genome annotation software and parameters**

**1) Repeat annotation：**

| | Software version or database version | The main parameters |
|---|---|---|
| LTR-FINDER | v1.05 | default |
| PILER | V1.0 | default |
| RepeatScout | v1.0.5 | default |
| PASTEClassifier | V1.0 | default |
| Repbase(database) | 19.06 | null |
| RepeatMasker | v4.0.5 | -nolow -no_is -norna -engine wublast -qq -frag 20000 |

## 2) gene prediction

| Method | | Software | The main parameters |
|---|---|---|---|
| gene prediction | Ab initio based | Genscan_3.1<br>Augustus_3.1<br>GlimmerHMM_v1.2<br>SNAP(version 2006-07-28) | default |
| | Homology based | GeMoMav1.3.1 | |

| | | | |
|---|---|---|---|
| | RNA-seq | Hisat v2.0.4<br>Stringtie v1.2.3<br>TransDecoderv2.0<br>GeneMarkS-Tv5.1<br>PASA v2.0.2 | default |
| | Integration | EVMv1.1.1 | default |

**3) Pseudogene prediction**

| | | |
|---|---|---|
| Pseudogene prediction | GenBlastA v1.0.4 | -e 1e-5 |
| | GeneWise2.4.1 | -both -pseudo |

**4) ncRNA prediction**

| | | | | |
|---|---|---|---|---|
| ncRNA | tRNA | tRNAscan-SE_v1.3.1 | | default |
| | snoRNA | Blast_v2.2.31<br>Infenal_v1.1.1 | Rfam_v12.1<br>miRBase_v21 | 1e-5 |
| | snRNA | | | |
| | rRNA | | | |

**5) Gene function annotation**

| | | | | |
|---|---|---|---|---|
| Function annotation | NR,KOG,GO,KEGG,TrEMBL,swissprot | BLAST v2.2.31 | NR,KOG,GO,KEGG,TrEMBL,swissprot | 1e-5 |

**6) Motif prediction**

| | | | |
|---|---|---|---|
| Motif prediction | InterProScan v5.8-49.0 | PROSITE,HAMAP,Pfam,PRINTS,ProDom,SMART,TIGRFAMs,PIRSF,SUPERFAMILY,CATHGene3D,PANTHER | default |