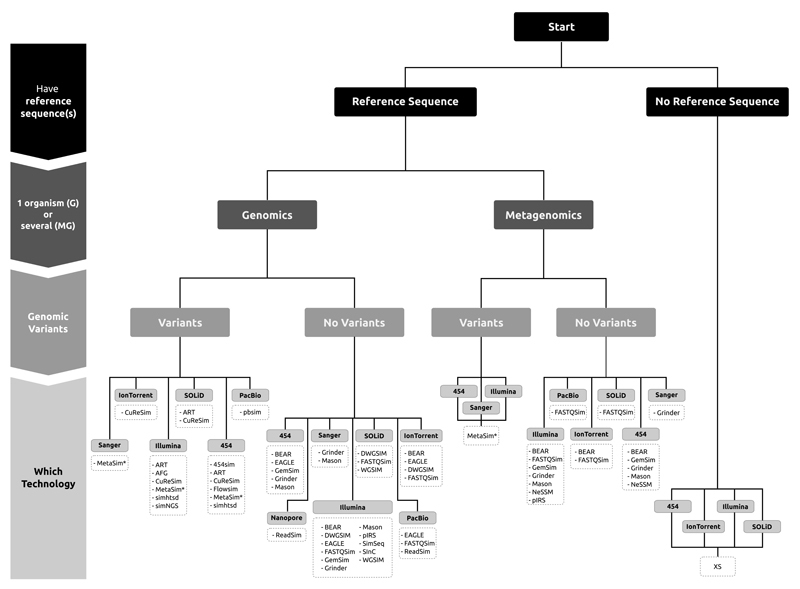
****

**NORGAL TOOL**

Norgal (de Novo ORGAneLle extractor) avoids this requirement by identifying a high frequency subset of k-mers that are predominantly of mitochondrial origin and performing a de novo assembly on a subset of reads that contains these k-mers.

Current methods to extract mtDNA from WGS data require a short seed sequence to initiate assembly.

The mitogenome will have a higher read depth compared to the nuclear genome and this difference in the read depth levels can be used to separate the reads into two groups.

The separation of the two types of reads is done by counting occurrences of subsequences of length k in the reads - k-mers - and classifying reads that have k-mers that are found more times than the nuclear read depth as being of non-nuclear origin.

Norgal uses raw short NGS reads from WGS data as input and outputs either a full or partial mitogenome.

usage: python norgal.py -i reads1.fq reads2.fq -O MyProjectFolder

Norgal v.1.0.0: Mitochondrial genomic DNA extraction from NGS reads.

optional arguments:

-h, --help show this help message and exit

-i FILE FILE Input paired fastq-files. (Required)

-o NAME Folder for output. (Required)

-r FILE Optional nuclear genome assembly of organism. Can be contigs, scaffolds, single nuclear gene etc.

-t INTEGER Number of threads. Default=2

-c INTEGER Depth-cutoff. Default=auto

-m INTEGER Minimum length of scaffold to be considered. Default=10,000bp

-b INTEGER Number of contigs to report with BLAST-hits. Default=10

-e FLOAT E-value cut-off for BLAST-search. Default=1e-5

--trim Remove adapters. Linux-only function. Default=disabled

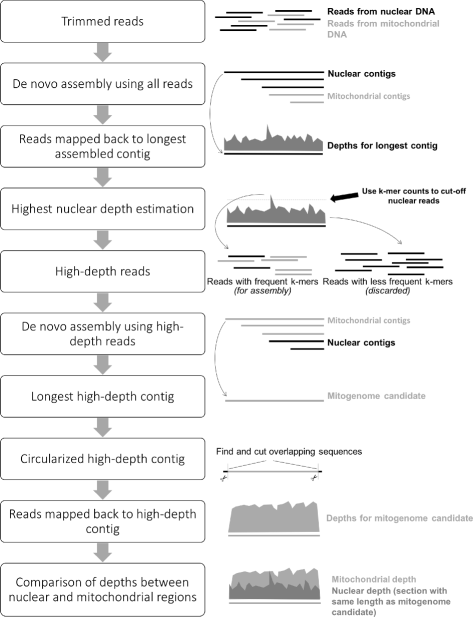
--blast Use best BLAST-hit from a mitochondrial/plastid database instead of longest assembled scaffold. Default=disabled

--delete Remove large temporary files (ie slim-mode)

**Script to calculate the execution time of norgal tool:**

nohup time -**f "\t%E real,\t%U user,\t%S sys"** -o /home/genome1/mitochondria\_assembly/results/norgal\_results/output.txt python /home/genome1/mitochondria\_assembly/tools/norgal/norgal.py -t 8 -i /home/genome1/mitochondria\_assembly/raw/ERR091374\_pass\_1.fastq.gz /home/genome1/mitochondria\_assembly/raw/ERR091374\_pass\_2.fastq.gz -o /home/genome1/mitochondria\_assembly/results/norgal\_results/ERR091374

This program will run the norgal program and the execution time in the format of real, system and user time will be stored into the output.txt at the defined path.



1. Trim and remove adapters from NGS reads using *AdapterRemoval* and perform a de novo assembly using *MEGAHIT* with default settings and the k-mer range: 21, 49, 77 and 105. Norgal assumes that the longest assembled sequence (contig) is nuclear.
2. Map the reads back to the longest assembled sequence using *bwa mem* and calculate the read depths for each position, if the longest assembled contig is longer than 100,000 base pairs, only the first 100,000 base pairs are used as it should be enough to determine the depth. The read depths of the mapped reads to this contig are used to determine the nuclear depth threshold (ND threshold) which is defined as the mean of all non-zero read depths from the 25*th* to the 75*th* percentile range multiplied with five.
3. Count k-mers of size 31 in all reads and only keep a subset of reads that contains at least one 31-kmer with a frequency that is greater than the ND threshold. This is done using the program *BBTools*.
4. Perform a de novo assembly using *idba\_ud* with the reads containing the frequent k-mers and extract either the longest contig or optionally the longest contig with a predicted cytochrome c oxidase subunit 1 (COI) gene. The binned reads with high-frequency k-mers are used for an assembly with *idba\_ud* with default settings which does multiple assemblies with different k-mer sizes in the range: 20, 40, 60, 80 and 100. This second assembly only contains contigs that have a high read-depth of at least the ND threshold.
5. Examine circularity of the longest contig, determine read depth, identify potential mitochondrial and chloroplast contigs, and output plots comparing depths between this contig and the longest contig from the assembly in step 1. Another option is to select the longest contig that has the best hits to full RefSeq mitochondrial or pastid genomes. The extracted contig is tested for circularity by comparing the ends of the contig and finding overlaps. Any overlapping base pairs are cut and the final sequence is reported as a potential mtDNA candidate. Norgal outputs a graph with the read depths as well as the read depths of a section of the nuclear DNA. This graph with the two sets of read depths may be used for validation of the mtDNA candidate, so if the depths over the mtDNA candidate is around 10-100 higher than the depths over the nuclear region, it increases the evidence that the candidate is from the mitogenome.

**MITObim (**MITOchondrial baiting and iterative mapping)

MITObim is capable of assembling mitochondrial genomes without the need of a reference genome of the targeted species by relying solely on mitochondrial genome information of more distantly related taxa as a starting point.

MITObim use relatively small sequences as a seed, such as a single gene sequence from the target mitogenome or from a more distantly related organism. MITObim uses cytochrome-oxidase subunit 1 (COI) sequences as the seed references.

**STEPS:**

1. Pre-Processing: NGS reads (read length 100 bp) were error corrected using the error correction tool (personal communication Ruibang Luo, BGI) of the SOAPdenovo2 software and quality trimmed using Mimicking Intelligent Read Assembly (MIRA) v3.4.1.1 for subsequent analyses.
2. step one identifies conserved regions between genomic reads of the target organism to the mitochondrial genome of a reference species by a mapping assembly from which a new reference is derived. This new reference may initially be gapped or consist of several contigs, as non-conserved parts of the target organism will not have mapped to the mitochondrial genome of the reference species.
3. Step two of MITObim is an in silico-baiting step: using the newly created reference sequence as bait, all reads from the total genomic readpool, which are partly or fully overlapping with the bait (minimum overlap is n k-mers of length k) are retrieved and integrated into a new data set. This baiting step with relatively long k-mers (k = 31) is crucial for the specificity of the whole process as the subsequent mapping assembly is performed with comparatively low stringency, i.e. allowing mismatches in up to 15% of the bases of the total length of a Smith–Waterman alignment overlap.
4. In the third step, the reads identified in the previous step are mapped back to the gapped reference sequence using MIRA v3.4.1.1. This will lead to an extension of the reference sequence and a reduction of gaps. To be incorporated as an extension into a mapping assembly, a read requires an overlap of at least 30 bases at the edge of a reference.

Diagram

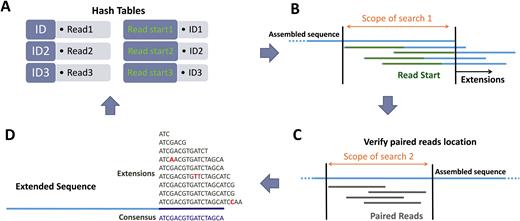
Description automatically generated

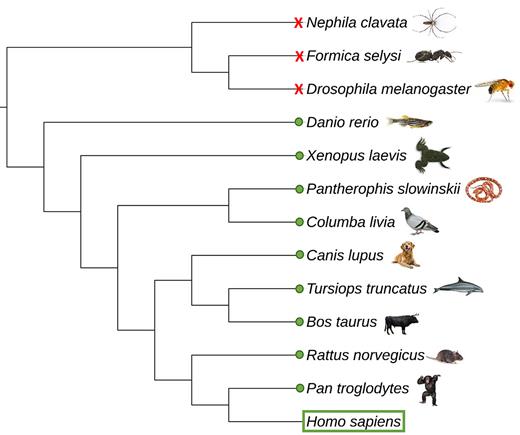
**Note**: The initial mapping assembly in step one may have increased memory requirements, as it is dealing with the entire genomic readpool. This increased memory consumption can be bypassed by an initial fishing step (-quick option) reducing the readpool to only reads with a certain k-mer overlap to the reference already before the initial assembly. This approach initially slightly decreases specificity, but nevertheless performs well with not too distantly related references available and can be performed as a standard primary test.

**NOVOPlasty**

* NOVOPlasty assembled all tested circular genomes in less than 30 min with a maximum memory requirement of 16 GB and an accuracy over 99.99%.  NOVOPlasty is the sole de novo assembler that provides a fast and straightforward extraction of the extranuclear genomes from WGS data in one circular high quality contig.
* NOVOPlasty is written in Perl.
* The genome coverage represents the percentage of the reference genome that was assembled minus ambiguous nucleotides.

* The accuracy represents the percentage of correctly assembled nucleotides relative to the ‘perfect’ validated alignments.
* NOVOPlasty is a seed-extend based assembler. The assembly must be initiated by a seed, which is iteratively extended bidirectionally. This seed sequence is not used for initiating the assembly, but to retrieve one sequence read of the targeted genome from the NGS data set. This strategy can handle a wider range of seed inputs without incorporating mismatches into the assembly.
* The seed sequence can be one sequence read, a conserved gene or even a complete organelle genome from a distant species.
* NOVOPlasty starts with storing the sequences into a hash table, which allows quick accessibility of the reads.
* The end and start of the seed are scanned for overlapping reads in the hash table and stored separately.
* All putative extensions are identified and subsequently cross-checked with the paired reads to verify if they are positioned correctly.
* Relatively similar sequences are grouped together, and every base extension is resolved by a consensus between the overlapping reads. When there is more than one possible consensus extension (i.e. more than one group of sufficient size), the assembly splits and two new contigs will be created.
* NOVOPlasty does not try to assemble every read but will extend the given seed until the circular genome is formed. The assembly will circularize when the length is in the expected range and both ends overlap by at least 200 bp.
* When a repetitive region is detected, the circularization will be postponed until the assembly exits the repetitive region.





**Seed compatibility test for the de novo assembly of the human mitochondria with 12 different mitochondrial genomes as seed sequence. A green dot means that the mitochondrial genome of that species can be used as a seed for the mitochondrial assembly of H. sapiens. Red X means unsuccessful**.

**ARC (Assembly by Reduced Complexity):**

ARC was only used for the mitochondrial assemblies since it still relies on reference genomes and when a very close reference was lacking, the assemblies resulted in a lower genome coverage than with the de novo approach.