What's in my food (WIMF) pipeline v2.0

***Olivier Emery**, PhD in Life Sciences*
Digital Epidemiology Lab
(Ecole Polytechnique Fédérale de Lausanne - EPFL)
Campus Biotech Geneva, Switzerland

December 13rd 2020

Contents

WIMF description **2**

Disclaimer **2**

WIMF main principles **3**

What's new in WIMF v2? **4**

WIMF v2 workflow summary **6**

WIMF input format **6**

Generating DNA sequences for WIMF **6**

Requirements **7**

Installation **7**

Basic usage **8**

Advanced usage **8**
Use of ingredient list 8
Changing default parameters 9
Editing the dictionary 9

Interpretation of results **10**

Description of result folders and main files **15**

Limitations **18**

Perspectives **18**

Extra remarks and protocol recommendations **19**

Troubleshooting guide **20**
Installation problems 20
Problems running WIMF 21

APPENDIX: Overview of WIMF analyses **23**

WIMF description

WIMF (for **W**hat's **I**n **M**y **F**ood) is an open source experimental automated bioinformatic pipeline to identify which plants and animals are present in food based on DNA sequences obtained from food samples sequenced with the MinIon instrument (Oxford Nanopore Technologies).
Version 1.0 was first developped in August 2019 in the context of the Open Food Repo DNA project which is a part of the Open Food Repo initiative. The Food Repo DNA workshop held on May 25th 2019 in collaboration with the biohacking association Hackuarium and the DNA-based food certification company SwissDeCode allowed to produce the sequencing results using the gene markers 16S rDNA gene for animals and the rbcL gene for plants.
For WIMF version 2.0, the wet laboratory protocol was modified to include a second marker for plants the matK gene and PCR primers used for rbcL amplification were modified to obtain a longer fragment. Furthermore, added functionality was implemented (See the **What's new in WIMF v2?** section for more details). The data used to develop this second version was provided by SwissDecode and corresponds to 96 multiplexed samples for which PCR reactions using the three gene markers were performed and, when enough DNA was available, sequenced using the MinIon sequencer.

Disclaimer

The WIMF (What's In My Food) pipeline is provided "as is" and "with all faults." It is not affiliated with the WIMP (What's In My Pot) pipeline neither with Oxford Nanopore Technologies. The author makes no representations or warranties of any kind concerning the safety, suitability, lack of viruses, inaccuracies, typographical errors, or other potentially harmful components of WIMF neither of the content of the tools used in WIMF or the links provided in this document. There are inherent dangers in the use of any software, and you are solely responsible for determining whether WIMF and its components are compatible with your equipment and other software installed on your equipment. You are also solely responsible for the protection of your equipment and backup of your data, and the author will not be liable for any damages you may suffer in connection with using, modifying, or distributing WIMF. Importantly, WIMF does not give quantitative results (e.g. a food composed of 50% ingredient A and 50% ingredient B will usually not correspond to 50% reads coming from ingredient A and 50% reads coming from ingredient B for several reasons among which: differential yield of DNA extraction between the ingredients and between various preparation forms, differential amplification of the products from the two ingredients = PCR bias etc...) and should not be used as a proof in determining whether an indicated ingredient is present neither to show that an unexpected ingredient is present.

WIMF main principles

WIMF tries to identify plant and animal species based on specific food DNA sequences (i.e. sequences of nucleotides represented by the A,C,G and T letters which form the DNA "alphabet"). All living things possess DNA and the diversity of species is also reflected in the diversity of their DNA sequences. The entire set of DNA sequences of an organism is called the genome and includes all genes (which are the blueprints for producing proteins) as well as regions not coding for genes in the case of Eukaryotes.
Several specific gene sequences have been found to vary enough between species to allow to assign a specific species to a specific version of this gene sequence: these are known as gene markers. The challenge to find a reliable gene marker is that its sequence needs to be similar enough to be found in all considered species but also different enough between species to allow unambiguous species assignments. The gene markers rbcL and matK are found only in plants while the 16S rRNA gene is found only in animals.
To generate the input data for WIMF, the first step is to extract DNA from homogenized food samples. The second step is to multiply the DNA sequences of the three specific gene markers chosen (i.e. rbcL/matK/16S) using polymerase chain reaction (PCR). Briefly, PCR is a molecular biology technique which allows to multiply a specific DNA region exponentially, a process also known as DNA amplification. After the PCR reactions, the initial DNA extracted from food is enriched with the copies (also known as "amplicons") of the three gene markers.
Next this DNA source with the amplified DNA fragments is sequenced (i.e. the letters are read from the DNA into a computer) using the MinIon instrument which is the most affordable DNA sequencer available. The main drawback of the MinIon is that it has a relatively high sequencing error rate, with a reported average of about 10% wrong nucleotides. Importantly, this error is much higher than the differences we need to be able to detect in the gene marker sequences to distinguish different species (the sequences of a given gene marker from two different species are typically >95% identical).
This problem can partially be overcome thanks to the combined used of PCR amplification and sequencing. Although each gene marker from each species is present in multiple identical copies after the PCR, the sequencing of these identical copies will lead to several different sequences due to the introduction of sequencing errors. WIMF will first filter the sequencing data using an average read PHRED score cutoff of Q10 (i.e. at most 1/10 bases wrong) in order to only consider high quality sequencing reads in the expected size range of the amplicons. Then WIMF will use SeekDeep in order to cluster highly similar reads (because we expect identical reads clusters for each species in the case there would be no sequencing errors) in order to be perform a partial sequencing error correction: with many identical copies and sequencing errors more or less spread along the normally identical sequences, a majority rule at each position (i.e. assigning the letter which is most prevalent at this position of a given gene cluster) can allow to resolve ambiguities. The clustering step is RAM intensive and depends on the number of reads to cluster. In case there are more reads than what SeekDeep can handle without causing a memory error on a 8Gb RAM computer, a second more stringent quality filtering of reads is performed in order to obtain a smaller number of reads to cluster.
Each read cluster is then compared against BLAST databases containing gene marker sequences already associated to different species obtained from NCBI. Finally, the species corresponding to the best result for each marker (if any) is reported by WIMF along with the number of reads contained in the clusters associated to that species and other statistics.
WIMF v2 also allows to use a list of ingredients (organisms) in order to favor these species among the best BLAST hits. This helps to get the expected organism when the best hit corresponds to a closely related species - but not the exact one - and that the expected species is among the best results.

What's new in WIMF v2?

- The **second plant marker for the matK gene** in addition to rbcL gene
  This allows to perform plant species assignment using the two different markers and to report the contribution of the two markers. This contribution is measured as the percentage of reads from each marker used to make the final species assignment relative to all reads - from matK and rbcL - assigned to this particular species. While WIMF v1 uses a single BLAST database, WIMF v2 uses three databases separately (one for each marker) and summarizes the results at the end. There are an estimated 102'440 animal species represented by 444'199 sequences in the 16S database, 60'190 plant species represented by 166'120 sequences in the matK database and 67'437 plant species in the rbcL database (90'595 plant species represented when considering both rbcL and matK)
- **Ingredient list to assist species assignment**
  A typical problem to identify species using a gene marker arises when the marker sequence is very similar and this can particularly happen in closely related species. In that case, the first BLAST hit reported by WIMF may not be the species we expect but a genetically close one at this marker. Furthermore, sequencing from MinIon produces errors which can also lead to wrong species assignment when errors correspond to positions with high discriminative power to distinguish species. To resolve this, a new feature to favor species from the ingredient list has been implemented. The user can supply a list of organisms which will be used to scan the best BLAST results ordered by bitscore. If one of the organisms of the ingredient is among the best BLAST results, it is used for species assignment instead of the first BLAST hit. A parameter named bitscore tolerance allows to be more or less stringent when defining results considered as "best BLAST results".
- **Translation of scientific (latin) names to common english using a dictionary**
  The dictionary is regenerated at each run from a simple text file containing the translations which can easily be edited if needed. It contains 15'138 entries. In case a latin name does not have its translation in the dictionary, it is printed in latin.
- Introduction of a **noise cutoff**
  Species which were determined from a fraction of all reads used for final species assignments below 2% are removed from the report. This value can be modified in the configuration file (e.g. setting it to zero will not filter any noise).
- Implementation of the **configuration file**
  A plain text file with all option values and their description was implemented to allow an easier use of multiple options and enhanced reproducibility by enabling to archive custom configuration files which can then be re-used later or shared between users.
- **BLAST database optimization**
  Although there are three BLAST databases in WIMF v2, these only use 371Mb of disk space (WIMF v1 database was 1.4Gb). The reason is that the database in WIMF v1 also included sequences of full chloroplast genomes (these can be over 100'000 base pairs long) while the marker genes matK and rbcL are smaller than 2000 bp and, similarly for the animal gene marker, the database included full mitochondrion genomes while the 16S gene is smaller than 2000 bp. To optimize the space used by the database, only sequences equal or smaller to 2000 bp were kept.
- **Automated dependency check**
  WIMF v2 checks at the beginning of each run that the required dependencies are available and errors to indicate which software is/are not available are printed to guide the user.
- **Warnings in less reliable species assignments**
  For sample analyses in which less than 200 reads were ultimately used for all species assignments, a warning is produced in red indicating that this analysis has an unexpectedly low number of reads and not to trust too much the obtained result (Note that they may be correct but have a higher chance of being wrong). An additional warning is produced in case the assignment of at least one species is based on less than 130 reads.

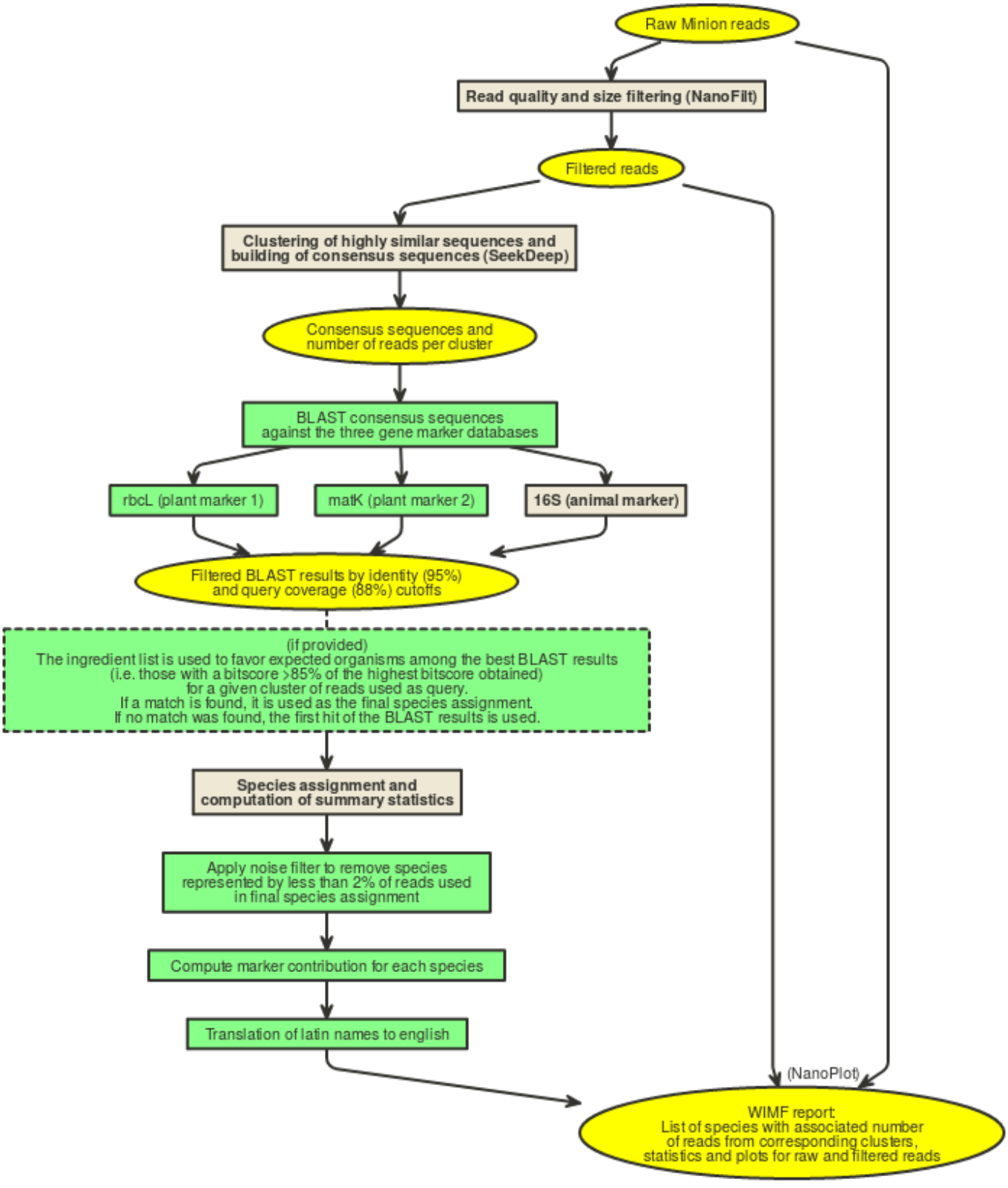

Figure 1: WIMF v2 workflow summary

WIMF v2 workflow summary

WIMF v2 workflow for "typical" samples (i.e. analysis quality filtered at average read quality > Q10) is shown in **Figure 1**. Rounded boxes indicate the different data/results at each point of the pipeline. The rectangular boxes represent data manipulation processes and arrows the direction of the workflow to and from data and processes. Green rectangular boxes correspond to new features in WIMF version 2.0, dashed lines indicate optional processes. Other checkpoints (i.e. minimal number of raw or filtered reads) or treatment of particularly high quality samples (i.e. which were filtered at >Q12 or even >Q13 before analysis) are not shown here for simplicity.

WIMF input format

WIMF takes as input demultiplexed amplicon sequencing data in **uncompressed FASTQ** format as delivered by the MinIon standard software suite, with one directory per sequencing run containing several directories (the latter with names starting with "BC" - for "Barcode" - containing all reads from one sample). See below how to obtain the appropriate sequencing data. **Tip** Rename your BCXX (XX is 01 or 02 or 03 etc...) directory to BCXX_[FOOD_SAMPLE_NAME] with [FOOD_SAMPLE_NAME] replaced by a short name corresponding to the food sequenced with the barcode XX. By doing so you will avoid scrolling through your list of barcodes to find the corresponding food when analyzing your results. When renaming, **always leave `BC` at the beginning of the sequencing run directory name**, this is used by WIMF to recognize directories containing the reads.

Generating DNA sequences for WIMF

Briefly, DNA is extracted from each sample and two specific genomic sequences present in all plants and one genomic sequence present in all animals are amplified (i.e. multiplied exponentially) via polymerase chain reaction (PCR), these three sequences are:

1. Plant gene marker 1: a region of 650 base pairs (bp) of the Ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit ("rubisco") ***rbcL*** gene. Note that the pair of primers used were modified relative to the WIMF v1 lab protocol in which the size of the amplicon was of 390 bp in order to possibly obtain better species assignments.
2. Plant gene marker 2: a region of 850 bp of the maturase K ***matK*** gene.
3. Animal gene marker: a region of 250bp for the mitochondrial **16S rRNA** gene.

Hence the resulting amplified sequences (i.e. "amplicons") have an expected size ranging from 250 to 850 base pairs (bp). Samples are barcoded so that PCR products from all samples can be pooled and sequenced simultaneously on a single sequencing run using only one flow cell and later separated (i.e. "demultiplexed") per sample.

Requirements

In order to run WIMF requires the installation of the following freely available softwares and their dependencies:

- Python3 (v3.5 or above)
- Git See how to install git here.
- NanoPlot https://github.com/wdecoster/NanoPlot
- PoreChop (github) https://github.com/rrwick/Porechop
- NanoFilt https://github.com/wdecoster/nanofilt
- SeekDeep v3.0.0 https://github.com/bailey-lab/SeekDeep
- NCBI BLAST+ v2.11.0 NCBI install instructions

Important: Python 3 must be set as default Python interpreter, PoreChop, NanoPlot, NanoFilt, SeekDeep, NCBI BLAST+ utilities need to be accessible from any directory. See the **Troubleshooting guide** on page 12 of this document for help on how to comply with these requirements.

In terms of hardware WIMF v2 requires at least 8Gb of RAM and a sufficient amount of free disk space for the local BLAST nucleotide database (371 Mb) and for temporary files of each run analysis (2 to 3 times the disk space used for the FASTQ files of the run to be analyzed). WIMF has been developped and tested on Linux (Ubuntu 18.04.5 LTS 64bit) on a PC with an Intel® Core™ i9-9880H CPU @ 2.30GHz × 16 (but only 4 were used) and 32Gb of RAM. Some programs in WIMF make use of the 4 cores and corresponding parameters may need to be changed from their default values in order to work on a computer with less than 4 cores or to run faster on computers with more than 4 cores.

Installation

After checking that your hardware meets the requirements and installing the required softwares (see **Requirements** and **Troubleshooting guide** sections), enter the directory where you would like to install WIMF and clone the `WIMF_v2` directory to your computer:

```
git clone https://github.com/salathegroup/Food-Repo-DNA-analysis-WIMF.git
```

Enter the install directory (`WIMF_v2`) and run the `config.sh` script in order to configure your WIMF install path and to make WIMF accessible from any directory on your computer:

```
cd Food-Repo-DNA-analysis-WIMF/WIMF_v2
./config.sh
```

To test that the installation was successful, enter the following command which should print the WIMF help if WIMF is correctly installed:

```
wimf -h
```

In case this prints an error, you may need to look at the **Troubleshooting guide** (p.12).

Basic usage

In order to run WIMF on your sequencing run directory (containing the barcoded demultiplexed FASTQ files in directories with names starting with "BC"), execute the following command (replacing `$PATH_TO_SEQUENCING_RUN_DIRECTORY` with the path of the sequencing run directory):

```
wimf -i "$PATH_TO_SEQUENCING_RUN_DIRECTORY"
```

Which corresponds to WIMF with default parameters. The full analysis of a run with 48 samples took about 25h. The results will be stored in a directory called `$PATH_TO_SEQUENCING_RUN_DIRECTORY_`**WIMF** which will be located next to the sequencing run directory, see the "Description of results" section below for details. In case you want to cancel the run press CTL+C.

Important notes In case WIMF is launched for the second time on the same sequencing directory (i.e. with existing WIMF outputs), the previous WIMF outputs will first be deleted before producing outputs from the last command. In case you would like to keep the WIMF outputs of the first run (which can be of interest if you use specific parameters for the different WIMF runs in order to later compare them, see advanced usage section below), just rename the WIMF output directory before launching WIMF for the second time. If you would like to run WIMF on only a subset of the samples, create a directory in which you copy the directories starting with "BC" that you would like to analyze and launch WIMF indicating this directory path instead of `$PATH_TO_SEQUENCING_RUN_DIRECTORY`. In case of interruption of the analysis because of an error, temporary files are kept.

Advanced usage

Use of ingredient list

The use of a list of expected organisms ("ingredients") is recommended as it allows to solve some ambiguities, just beware of strictly respecting the format see below. In order to use a list of ingredients contained in a text file named MY_INGREDIENT_LIST.txt present in the directory from which WIMF is run, run the following command:

```
wimf -i "$PATH_TO_SEQUENCING_RUN_DIRECTORY" -l MY_INGREDIENT_LIST.txt
```

Note: if the ingredient list is in another directory, you can specify the full path of the file.

Important: The ingredient list MUST have the following format and be saved using a simple text editor (e.g. Notepad/Gedit but not MS Word):

```
BC01,Carrot,Leek,Paprika,Celery,Parsley,Lovage,Nutmeg,Pepper
BC02,Soybean,Wheat,Canola,Mustard,Celery,Cow,Palm,Corn
# etc...
```

One line is used per sample. The first item is the sample barcode folder ("BCXX"), followed by all expected organisms comma-separated without spaces. All MUST start with a capital letter. Pay attention to name organisms and not meat type (e.g. "Cow" and not "Beef", "Pig" and not "Pork" etc). Optional comment lines must start with # (so that they are ignored when parsing the file). Note that the use of the ingredient list automatically involves to translate these names to latin and hence the dictionary is used. In case of doubt (certain fish species can have different common names, certain legumes are derived from the same species etc.), consult the content of the file dictionary.csv which is located in the "resources" directory in your WIMF install path. If this is the case but you would like to fix this, you can interrupt the analysis using CTL+C and, for example, correct a typo in an organism of your list after checking the dictionary and re-run the analysis so that the species is taken into account. WIMF will let you know at the beginning which entries were not found in the dictionary but will continue the analysis without your intervention. It is not mandatory

to define a list for all samples. If a sample is not present in the sequencing run folder, a warning is produced but the analysis continues.

Changing default parameters

WIMF default settings are stored in the configuration file located at

```
WIMF_INSTALL_PATH/resources/config.txt
```

All parameters and their usage are described in this file. To use a custom configuration, make a copy of this file under a different name using a simple text editor (i.e. notepad/gedit), modify values and save.

You can run this command to use your own configuration file:

```
wimf -i "$PATH_TO_SEQUENCING_RUN_DIRECTORY" -c "$PATH_TO_CONFIG_FILE"
```

The command

```
wimf -h
```

will display WIMF help and describe the different command options.

Editing the dictionary

The dictionary is the sole source of translation from latin scientific names to common english names. In case a species provided by the user via the ingredient list is not found in the dictionary, a warning is produced to let the user know that the entry was not in the dictionary. This can happen if there was a typo or a missing capital letter at the beginning or for other linguistic reasons (e.g. some fish species can be named differently depending on locations). The dictionary file is:

```
WIMF_INSTALL_PATH/resources/dictionary.csv
```

Please note that if there are several english definitions for a given latin species name, only the last one is used. Only use plain text editors (Notepad/gedit and similar) to edit and save the file.

Interpretation of results

At the end of the analysis of all samples (note that it took 29h to complete the analysis of 48 samples, see Appendix to get the time spent per sample), some general information about the samples is printed:

```
************************************************************
****************WIMF ANALYSIS FINISHED**********************
***************Time Elapsed: 28h:52m:28s********************
************************************************************

45 sample(s) went through the analysis with quality filter >Q10 ("typical" samples)
These are stored in SeqRun_96_WIMF/Filter1Samples

2 sample(s) went through the analysis with quality filter >Q12 ("high quality" samples)
These are stored in SeqRun_96_WIMF/Filter2Samples

1 sample(s) with less than 100 filtered reads which were not analyzed (besides raw+filtered
data statistics and plots), a complete list is available in the following file:
SeqRun_96_WIMF/LOGS/LowFiltReads.log
These are stored in SeqRun_96_WIMF/LowFiltReadsSamples

8 sample(s) with less than 200 raw reads which were not analyzed (besides raw data statistics
and plots), a complete list is available in the following file:
SeqRun_96_WIMF/LOGS/LowRawReads.log
These are stored in SeqRun_96_WIMF/LowRawReadsSamples

38 sample(s) with less than 10 raw reads which were not analyzed, a complete list is available
in the following file:
SeqRun_96_WIMF/LOGS/NotAnalyzed.log

Compacting log from SeqRun_96_WIMF/LOGS/WIMF.log

The available HTML reports can be found in the folder SeqRun_96_WIMF/FullReport
```

Look at the HTML report from a sample located at `$PATH_TO_SEQUENCING_RUN_DIRECTORY_WIMF/FullReport` to see which species plant and/or animal species were detected based on the WIMF pipeline. These reports indicate which species were found (or provide only data plots and statistics if no assignments could be made) and the proportion of reads corresponding to each species (See **Figure 2**). Further statistics on the read clusters and BLAST results are available (See **Figure 3**) as well as the raw data and filtered data plots. In case of suspicious data, warnings are shown. In case there are warnings in the report, inspect the raw sequencing data plot at the bottom of the report. We expect the gene markers which were amplified by PCR to show up at their expected amplicon size. For example, see **Figure 4** and **Figure 5** to see how the raw sequencing data should normally look like when the PCR and sequencing of the gene marker for 16S and matK, respectively, were successful. On the contrary, **Figure 6** shows a sample in which no amplification could be observed (and for which there was a very low number of reads used for final species assignments).

For average quality samples (as well as for very and extremely high quality samples), the `SpeciesTable.csv` file will provide scientific species names and the % of reads matching to that species (relative to all reads mapped to different species used at the end for all species assignments, after applying a noise cutoff).

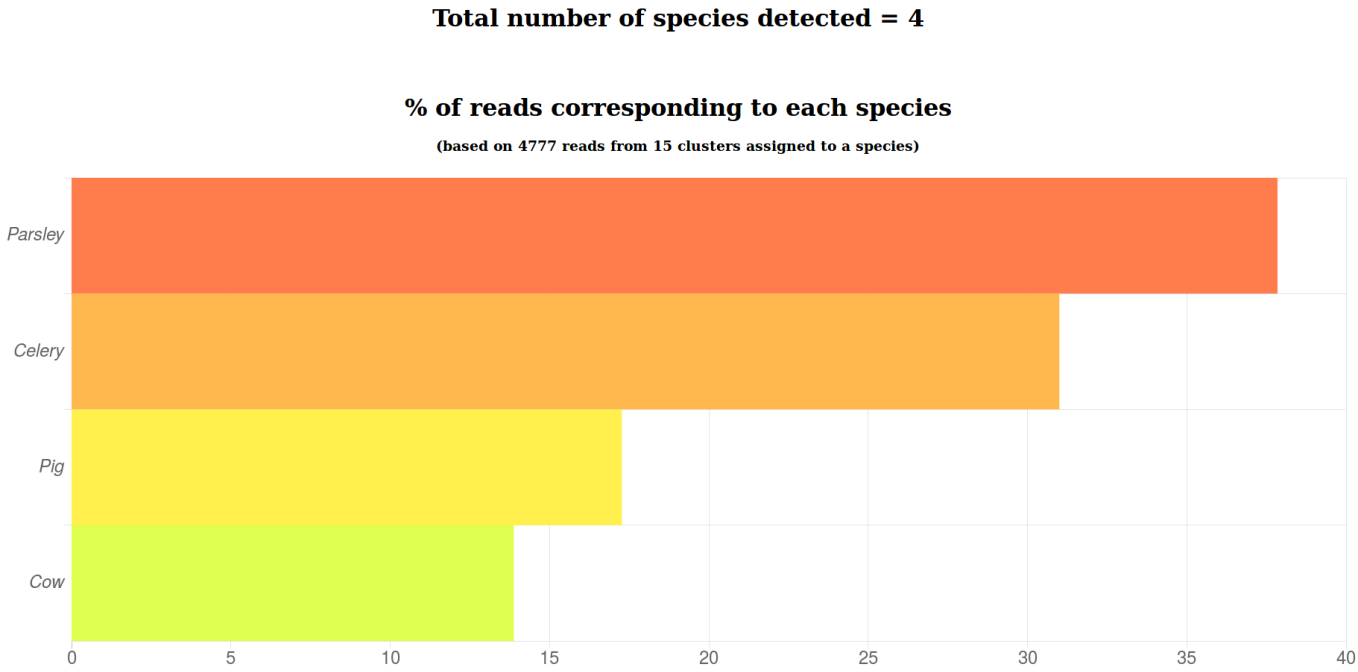

Figure 2: Species assignments plot with the corresponding % of reads per species (Sample BC30)

Species clusters summary statistics

Species english name	Species latin name	Total number of reads	Number of clusters	% of identical matches*	Query coverage*	Query length*	Number of mismatches*	Number of gap openings*	E-value*	Bit Score*	Marker	Species marker contribution (%)
Parsley	*Petroselinum crispum*	1808	9	99.2699	99.9336	700.676	1	4.03982	2.89491e-143	1262.14	matK	100.00
Celery	*Apium graveolens*	1481	2	98.1472	99.0149	855.028	5.92573	8.92573	2.0054e-127	1482.42	matK	100.00
Pig	*Sus scrofa*	825	2	98.6022	100	237.24	1	2	2.08533e-111	417.64	16S	100.00
Cow	*Bos taurus*	663	2	99.2712	99.3484	232.045	1.34842	0.348416	3.15318e-115	419.213	16S	100.00

* weighted average from NCBI BLAST results across clusters for this species, with the number of reads per cluster as weight

Figure 3: Species assignments statistics (Sample BC30)

Figure 4: Amplification for the 16S rRNA gene marker can be seen at around 250bp

Figure 5: Amplification for the matK gene marker can be seen at around 850bp

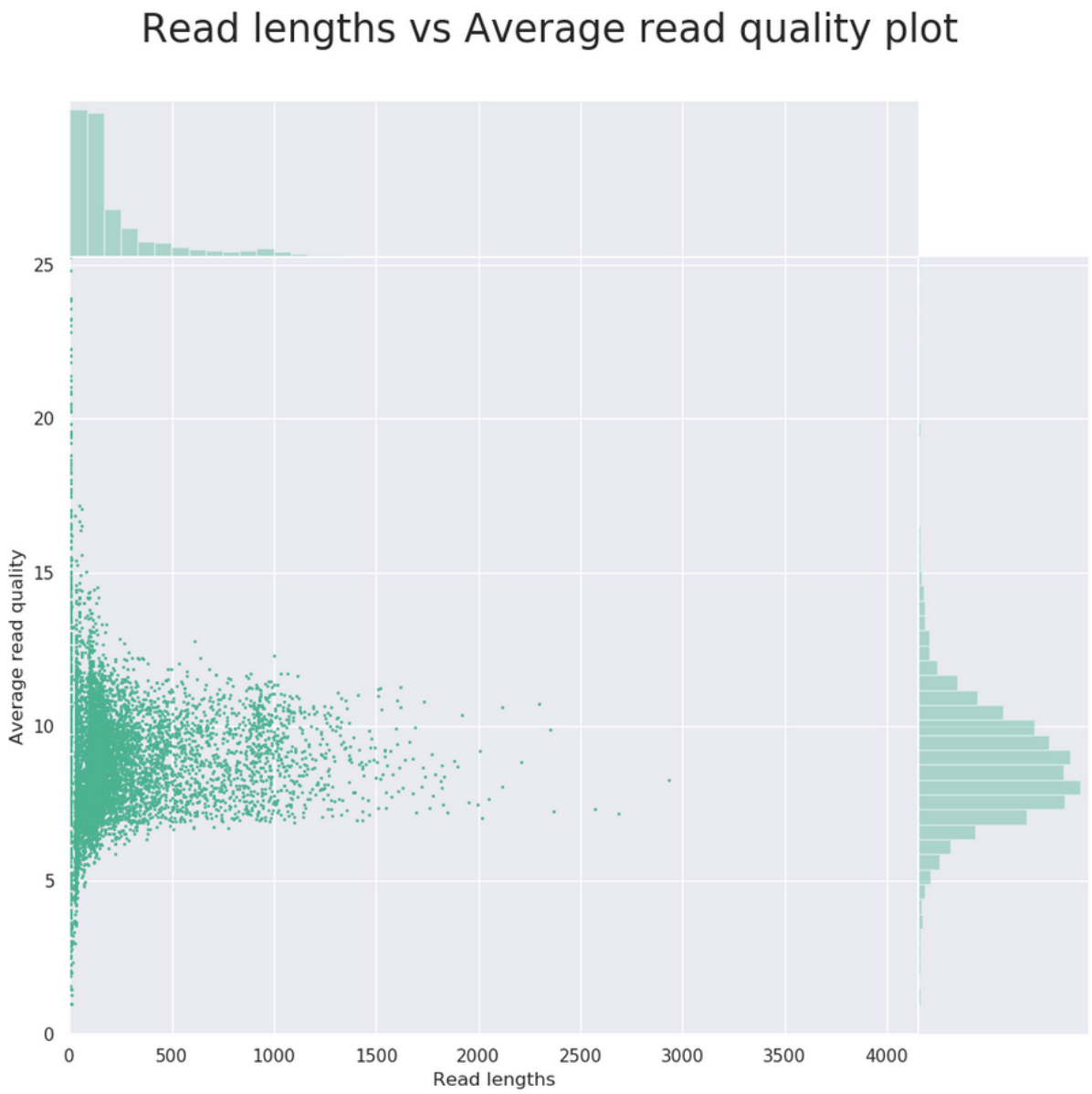

Figure 6: Problematic sample: no amplification can be seen at expected amplicon sizes

Description of result folders and main files

As the WIMF analysis is performed, the outputs are produced in `$PATH_TO_SEQUENCING_RUN_DIRECTORY_WIMF` directory which contains the folder and files structure shown in **Figure 7**. You can explore this files during the analysis in case you would like to see reports as they arrive (Note that during clustering WIMF needs an important amount of RAM and it may be wiser to close the internet browser during this step to avoid out of memory errors).

Figure 7: WIMF results folder and files structure

- **FullReport/** Produced at the very end of the analysis, this folder contains all available HTML reports. Start by visualizing reports from here using an internet browser.
- **Filt1Samples/** contains typical samples which passed the full analysis after the first quality filter.
  - **BCXX** sample XX directory containing the following subdirectories:
  - **01_QC_NanoPlot/** contains NanoPlot outputs for raw reads (plots+statistics) - **02_Filtering_NanoFilt/** contains filtered reads (**filtSeqs.fastq** and **filtSeqs.fasta**) as well as NanoPlot outputs for filtered reads
  - **03_clusterConsensus/** contains SeekDeep qluster outputs including consensus sequences for the clusters obtained (**output.fastq** and **output.fasta**)
  - **04_BLASTclusters/** which contains the following subdirectories:
  - **16S/ matK/ and 16S/** each contains the result files corresponding to the respective gene marker
    - **BLASTresults.out** raw BLAST results (maximum 100 per query) using cluster consensus sequences as queries against the corresponding (rbcL/matK/16S) database
    - **firstHits.txt** the top BLAST hits (one per query) from **BLASTresults.out**
    - **BLASTresultsfiltered_TMP.out** filtered BLAST results (coverage > 88% and identity >95%)
    - **ingrFiltResults.txt** (only when ingredient list used) shows BLAST results obtained after applying ingredient list option.
    - **clustersSummary.txt** based on BLAST first hits (or on), each line corresponds to one species and is composed of its name, the number of reads supporting this species, the numbers of clusters and BLAST statistics (in order: % identity to hit, query coverage, alignment length, number of mismatches, number of gap openings, e-value, bit score) determined by a weighted average using the number of reads per cluster as weights for each cluster associated with this species. An internal filter removes hits for which the query coverage is below 88% and the identity to the hit below 95% before producing this file (most of the time this removes clusters supported by only one read - = singletons - of rather low identity). Note that a file with the same name in the parent folder combines values from all markers with a noise cutoff applied and marker contribution.
    - **speciesTable.csv** comma separated file containing the name of each species detected and their percentage of reads relative to all reads used (i.e. sum of reads that make the clusters for all species indicated in this file), may be used as input for webpage graphs (note that a file with the same name in the parent folder combines values from all markers with a noise cutoff applied)
    - **uniqueIDs.txt** cluster consensus sequences IDs (is used to get first BLAST hits)
  - **REPORT** contains WIMF report in HTML format (**report.html**) as well as a subdirectory (**HTML_files/**) containing the files used by the HTML report
- **Filt2Samples/** contains very high quality samples which passed the full analysis only after the second quality filter, refer to **Filt1Samples/** for description of subdirectories
- **Filt3Samples/** contains very high quality samples which passed the full analysis only after third quality filter, refer to **Filt1Samples/** for description of subdirectories
- **LowFiltReadsSamples/** contains samples which did not pass the full analysis because there were not enough filtered reads, includes a report only on the raw and filtered data (without species determination).
- **LowRawReadsSamples/** contains samples which did not pass the full analysis because there were not enough raw reads, these include a report only on the raw data.
- **LOGS** contains log files
  - WIMF.log: same as standard output, read it with `cat WIMF.log`.
  - WIMFcompact.log: a compacted version of WIMF.log, to be read with basic text editors.

- `LowFiltReads.log`: list of samples with not enough filtered reads to carry the full analysis (may not be present depending on the data analyzed)
- `LowRawReads.log`: list of samples with not enough raw reads to carry the full analysis (may not be present depending on the data analyzed)

Notes: **`LowFiltReadsSamples/`** only have plots for raw and filtered data, **`LowRawReadsSamples/`** only plots for raw data. If you would like to share would like to share a report and see it in another computer, share the **`REPORT`** directory including the subdirectory **`HTML_files/ HTML_files/`** which includes plots and javascript definitions to build the barplot of species.

Limitations

Unfortunately, the rbcL marker used was apparently not amplified/sequenced in the sequencing run used to develop WIMF version 2 although rbcL was previously amplified and sequenced in the development of WIMF version 1 using another pair of primers (the goal by changing primers was to obtain a longer amplicon which could potentially differenciate more species). It is not clear why there was no amplification for rbcL (wrong pair of PCR primers used? Problems in the PCR thermal program? Problem with the design of primers? Error during protocol etc.). However, rbcL was sometimes detected with a few reads (presumably from not amplified DNA) with species identification from matK and rbcL indeed pointing to the same species. It would be important to find out if the primers used work or if there was a problem in their thermal PCR program. It was hence not possible to assess properly the combined use of matK and rbcL (for example if a species would be detected by both markers we could give it a higher confidence score). Without using an expected list of species, some genetically close species at a given marker may be assigned instead of the expected one, although the expected one may also present in the BLAST results but just not the first hit with the highest score. The use of the expected list of species allows to use the expected species for species assignment, only if they are present among the best BLAST results. Although the identification of species using the expected list of ingredients allows to resolve many ambiguities, if several species which are genetically close at the selected gene marker are part of the same sample in the ingredient list (note that there were no examples among the samples used), the first one found among the best BLAST results will be used for species assignments and may not be the correct one. Finally, the timing needed to run an analysis may be limiting for certain applications.

Perspectives

WIMF allows to detect plant and animal species from plant DNA sequenced with the MinIon instrument. Specific applications will guide the choice of sequences to use to populate BLAST databases. This will depend on how specific and accurate the application needs to be. WIMF works as a standalone laptop computer application it is hence useable anywhere without an internet connection. However, it could be interesting to make a server version of WIMF accessible from the internet with a login. This would avoid the installation process to the end user and would enable a decentralized access, and potentially higher shared computing power. Another interesting option would be to try to obtain and analyze the sequencing data from the MinIon during sequencing: this could reduce time from sequencing to delivering results and a "sequence until" approach could allow to only sequence as much as needed (and use sequencing pores more effectively). Another perspective would be to extend WIMF to also screen for bacterial species with an existing 16S rRNA gene marker that has been heavily used previously in many metagenomic amplicon sequencing studies. This would be of particular interest for fermented foods or to potentially detect pathogenic bacteria due to impromper storing or cold chain failure. Similarly, a marker for species of the *Fungi* kingdom (e.g. ITS for **i**nternal **t**ranscribed **s**pacer) could allow to distinguish mushroom species as well as yeast-related organisms such as molds that may be present folowing food spoilage.

Extra remarks and protocol recommendations

The BLAST databases were not fully curated, it is possible that some names appearing in WIMF do not correspond to a species. This can happen if the header of the FASTA sequence is not properly formatted and that the fields corresponding to the latin species names are misplaced. In addition, there is some redundancy with sometimes several sequences per species. It was not evaluated here how these can impact results but we can imagine that having too many identical sequences in the database pointing to the same species may lower the chance of finding an expected species from the list (when used) since each query is limited to 100 BLAST results. In this version, a given plant species may be present in only one of matK/rbcL databases. This can allow to find species not represented by both markers but it may be preferable to always have a pair of markers in order to compare their results if the goal is more sensitivity than specificity.
There are possible code optimizations to avoid redundancy (e.g. making a common function for the different filters that takes as arguments the quality threshold as well as a common function for the clustering of the reads filtered at different quality thresholds using the error quality ranges as arguments) and to make the code more consistent across the whole script or faster. One possibility to save time would be not to BLAST a cluster of reads which has already been found in one of the other databases (this would also avoid some unecessary processing of BLAST outputs). No attempts were made to parallelize processes except using options provided by the different softwares used when possible. The HTML report provides numerous statistics but could be improved in terms of visualization / viewing design. In particular, there seems to be an issue with NanoPlot to show the range of lengths specified (up to maximum expected amplicon size), another plotting system may be more appropriate to fully customize the plots. To make WIMF more user friendly, it would be needed to develop a user interface which could be deployed on multiple operating systems. The SeekDeep qluster module posseses many features to deal with clustering of similar reads which were not fully explored during this project. WIMF has been developed starting with demultiplexed FASTQ files obtained with the standard software from Oxford Nanopore Technologies and it was not assessed during this work how many reads could not be assigned to a given sample. It may be interesting to test the DeepBinner open source demultiplexing tool in order to see how well it performs relative to the default software from Oxford Nanopore Technologies. WIMF v2.0 performs the full analysis from beginning to the end, future versions should divide the parts of the analysis (e.g. quality control/filtering/clustering/BLAST...) so that the user can decide to stop at any point and resume the analysis from this point without having to recompute files. Due to the highly heterogeneous nature of the demultiplexed FASTQ files (some had only a few raw reads, others had over 300'000 with various read lengths and qualities), it is difficult to find a one-for-all solution that works for all samples. WIMF was developed based on the sequencing data from a single run, tuning parameters may be needed for your own samples.
It was a choice to focus on reads with quality above Q10 and lengths corresponding to expected amplicon sizes. While this worked for most samples, it is possible that in some cases where WIMF says that there are too few filtered reads to carry on the analysis that taking reads of lower qualities or of smaller length could allow to extract relevant information. However this was not investigated here and incorporating lower quality and smaller read lengths could result in more ambiguity to determine the true original amplicon sequence. Although options have been implemented to allow the user to set custom settings, these were not heavily tested.
With the actual laboratory protocol, add a PCR product purification step in order to sequence specifically sequences from the amplicons, sequences of the PCR products of each primer pair could be migrated on gel and DNA purified from the band cut from the gel (unless for some reason this approach is not suited for Minion sequencing chemistry, comercial kits exist to do this and this would eliminate longer DNA molecules which we are not interested in sequencing). It would be great to record the background (i.e. previous experience with DNA extractions or not) of workshop participants in order to see if samples for which sequencing failed can be explained by this. Based on SwissDecode data, some experiments performed better some worse so the experimentor's background may not explain everything. Think to record the order in which the different samples were blended (in order to explain possible contaminations). To avoid cross contamination of samples: either test and improve the cleaning of the blender to remove all residual materials or, even better, use a bead beater with individual tubes for each sample.

Troubleshooting guide

Installation problems

Setting Python3 as the default Python interpreter

To check your version of Python, type the following command in your terminal:

```
python --version
```

Which should return a version of Python above 3.5, for instance:

```
Python 3.5.2
```

If you get a version of Python below 3.5 with the previous command, Python3 is either not installed or, alternatively, it is possible that Python3 is installed but not used as default Python interpreter. You can check if Python3 is installed with this command (notice the "3" appended to `python`):

```
python3 --version
```

If you do get a Python3 version above 3.5 with the previous command, this means that you need to set Python3 as your default Python interpreter (see below). If you get a "`command not found`" error, Python3 needs to be installed. To set Python3 as default Python interpreter, you need to add the following line:

```
alias python=python3.5
```

to your ~/.bashrc profile (edit the `~/.bashrc` file using your favorite basic text editor and save changes) so that when you type `python` in the terminal, this becomes equivalent to entering the command `python3.5` . In order for this change to take effect immediately (without the need of rebooting) you can enter:

```
source ~/.bashrc
```

Notes: do not put `alias python=python3.5.2` if you have Python v3.5.2 installed but instead use only the first two digits as above, similarly if you have Python v3.7.3 installed enter `alias python=python3.7` and not `alias python=python3.7.3`

Making programs accessible from any directory

If for example BLAST is installed at `/home/user/WIMF_v2/BLAST_install/ncbi-blast-2.11.0+/bin` you need to execute this (adapt this command to the directory where your BLAST install is located):

```
echo "export PATH=$PATH:/home/user/WIMF_v2/BLAST_install/ncbi-blast-2.11.0+/bin" >> ~/.bashrc
source ~/.bashrc
```

You may need to do this for some of the required softwares depending on how they were installed if you see that they lead to a "command not found" error message.

Installing gcc7 and g++7 compiler (required to install SeekDeep)

Here are the commands I ran to install gcc7 and g++7 compilers in order to install SeekDeep (see Requirements section), it worked on my system but it is not guaranteed to work on yours:

```
sudo add-apt-repository ppa:george-edison55/cmake-3.x
sudo apt-get update
sudo apt-get install cmake
sudo add-apt-repository ppa:jonathonf/gcc-7.1
sudo apt-get update
sudo apt install gcc-7
sudo apt install g++-7
```

Problems running WIMF

Some samples were skipped during the analysis, analysis was interupted

While it is a normal behaviour of WIMF not to analyze certain samples (e.g. samples with very few reads), it is possible that WIMF skips certain samples which could indeed be analyzed if it stopped unexpectedly. If this happens, you will see a `BC_XX` directory (with XX being numbers) in the main WIMF results directory. This will indicate at which sample the analysis stopped (provided temporary files are kept which is the default behavior unless using the `-e` option). In this case, you can use the following command from the installation directory (`WIMF_v2`) to retrieve a compact log of the run in order to obtain error messages and to find the cause of the problem (possibly in this troubleshooting guide):

```
./logCompactor.sh $PATH_TO_SEQUENCING_RUN_DIRECTORY/LOGS/WIMF.log
```

Note: replace `$PATH_TO_SEQUENCING_RUN_DIRECTORY` with your actual path. The logCompactor will produce a file called `WIMFcompact.log` next to `WIMF.log`. The compact log file is generated by default for all runs which completed successfully. In case of interruption the commands above are needed to obtain a compact log.

Permission denied error

If you get the following error message while attempting to run `wimf -i "$PATH_TO_SEQUENCING_RUN_DIRECTORY"`:

```
bash: wimf: Permission denied
```

Make the script executable on your computer with the following command from the 'WIMF_v2' installation directory:

```
chmod u+x wimf
```

And retry to launch WIMF with `wimf -i "$PATH_TO_SEQUENCING_RUN_DIRECTORY"`

To obtain the full path of the sequencing directory, navigate in the terminal (with `cd`) until you are inside the sequencing run directory, then execute

```
pwd
```

The printed path corresponds to `$PATH_TO_SEQUENCING_RUN_DIRECTORY`

SeekDeep memory overlimit

If you get the following error message or if your computer freezes (cannot move mouse cursor normally, slows down):

```
20432 Killed  SeekDeep qluster --fastq ${RUN_DIRECTORY}_WIMF_QC/$BC_DIRECTORY/02_Filtering_...
```

This means that you went beyond the memory limit of 8Gb of RAM. Try to relaunch WIMF while keeping all other programs closed (internet navigator, PDF reader etc). If you still get the error, try to lower the threshold of the maximum number of reads to produce clusters (default empirically set to 10'050 reads using a PC with 8Gb of RAM)

SeekDeep Error in aligning

The following error is pretty rare (hundreds of samples were tested and it happened only with a pair of reads from four samples):

```
Clustering at 96.5% identity, quality thresholds Q12, Q10... static void njhseq::alignCalc
::runNeedleDiagonalSave(const string&, const string&, uint32_t, uint32_t, njhseq::alnParts
&), error in aligning:
CTAGCATCGTCGCTGTAACGATCAAGACTGAGTAGACCGTCAGTCCATACAGTTGTCCATGTACCAGTGAAGATTCAGCAGCTACCGCGG
CCCTGCTTCCTCAGGTGGAACTCAGGTTGAGAGTTACTCGAAATGCTGCCAAAATATCGGTATCTTTGGTTTCATAGTCAGGGTATAATA
AGTCAATTTATAATCTTTAACTTCAGCTTTGAATCCAACACTTGCTTTAGTCTCTGTTTGTGGTGACATG
AACACATTACCTACAGTAAGGTAAACATGTTAGTAACAGAACCTTCTTCAAAAGGTCCTACAGGGGGTAGCTACATAAAGGCAATAAAAT
CGACTTTTTTTCTTCTCCAGCAACAGGCTCGATGTGGTAGCATCGCCCTTTGTAACGATCAAGACTGGGTAAGACCGTCCAGTCCATACA
GTTGTCCATGTACCAGTAGAAGATTGGCAGCTACCGCGGCCCCTGCTTCCTCAGGTGGAACTCCAGGTTGAGGAGTTACTCGAAATGCTG
CCAAAATATCAATTATCTTTGGTTTTCATAGTCAGGAGTATAATAAGTCAATTTATAATCTTTAACACACCGAAACTTTGAATCCAACAC
TTGCTTTAGTCTCTGTTTGTGGTGACATG
```

The reason for this error is not clear, for some reason SeekDeep could not align the two sequences that are printed. The solution is to remove those two reads from the dataset. You can use `grep` on the concatenated FASTQ file to find each read corresponding to each sequence (note that the two sequences are separated by a newline in the error message shown above) using each sequence as the pattern in `grep` and delete them manually (don't forget to save the FASTQ file after removing these two reads).

APPENDIX: Overview of WIMF analyses

33/48 (68.75%) Analysis OK
13/48 (27.08%) Low number of assigned reads (<200)
1/48 (2.08%) No species assignment despite many reads (no amplification)
1/48 (2.08%) Not enough raw reads
Time Elapsed for BC01 : 00h:08m:58s Low number of assigned reads (<200)
Time Elapsed for BC02 : 00h:38m:48s Analysis OK
Time Elapsed for BC03 : 00h:30m:15s Analysis OK
Time Elapsed for BC04 : 00h:25m:51s Low number of assigned reads (<200)
Time Elapsed for BC05 : 00h:14m:55s Analysis OK
Time Elapsed for BC06 : 01h:56m:42s Low number of assigned reads (<200)
Time Elapsed for BC13 : 00h:09m:29s Analysis OK
Time Elapsed for BC14 : 00h:16m:51s Analysis OK
Time Elapsed for BC15 : 00h:08m:49s Low number of assigned reads (<200)
Time Elapsed for BC16 : 00h:53m:28s Analysis OK
Time Elapsed for BC17 : 00h:16m:01s Analysis OK
Time Elapsed for BC18 : 00h:17m:51s Analysis OK
Time Elapsed for BC25 : 00h:29m:56s Analysis OK
Time Elapsed for BC26 : 00h:45m:07s Analysis OK
Time Elapsed for BC27 : 01h:13m:49s Analysis OK
Time Elapsed for BC28 : 00h:35m:39s Analysis OK
Time Elapsed for BC29 : 00h:54m:32s Analysis OK
Time Elapsed for BC30 : 00h:22m:45s Analysis OK
Time Elapsed for BC37 : 00h:25m:13s Analysis OK
Time Elapsed for BC38 : 00h:04m:48s Analysis OK
Time Elapsed for BC39 : 00h:24m:25s Analysis OK
Time Elapsed for BC40 : 00h:38m:42s Low number of assigned reads (<200)
Time Elapsed for BC41 : 00h:21m:35s Analysis OK
Time Elapsed for BC42 : 00h:08m:55s Low number of assigned reads (<200)
Time Elapsed for BC49 : 01h:45m:33s Low number of assigned reads (<200)
Time Elapsed for BC50 : 00h:21m:21s Analysis OK
Time Elapsed for BC51 : 00h:38m:25s Low number of assigned reads (<200)
Time Elapsed for BC52 : 00h:23m:55s Analysis OK
Time Elapsed for BC53 : 00h:20m:28s Low number of assigned reads (<200)
Time Elapsed for BC54 : 00h:23m:39s Analysis OK
Time Elapsed for BC61 : 00h:24m:35s Analysis OK
Time Elapsed for BC62 : 00h:17m:53s Analysis OK
Time Elapsed for BC63 : 00h:30m:41s Analysis OK
Time Elapsed for BC64 : 00h:09m:20s Low number of assigned reads (<200)
Time Elapsed for BC65 : 02h:30m:54s No species assignment despite many reads (no amplification)
Time Elapsed for BC66 : 00h:07m:31s Analysis OK
Time Elapsed for BC73 : 00h:18m:36s Analysis OK
Time Elapsed for BC74 : 00h:37m:06s Analysis OK
Time Elapsed for BC75 : 00h:10m:53s Analysis OK
Time Elapsed for BC76 : 00h:00m:16s Not enough raw reads
Time Elapsed for BC77 : 00h:29m:26s Low number of assigned reads (<200)
Time Elapsed for BC78 : 00h:09m:12s Low number of assigned reads (<200)
Time Elapsed for BC85 : 00h:18m:45s Analysis OK
Time Elapsed for BC86 : 00h:34m:59s Analysis OK
Time Elapsed for BC87 : 00h:37m:59s Analysis OK
Time Elapsed for BC88 : 00h:21m:09s Analysis OK
Time Elapsed for BC89 : 04h:06m:45s Low number of assigned reads (<200)
Time Elapsed for BC90 : 00h:49m:07s Analysis OK