Manual

For testing purposes FASTQ, FASTA and BLAST outputs for amino acid corrections have been provided.

Get miniBarcoder

```
>>>git clone https://github.com/asrivathsan/miniBarcoder
```

Move into miniBarcoder directory

```
>>>cd miniBarcoder
```

Get dataset from Dryad:

```
>>>sh get_manuscript_datasets.sh datasetid
```

where dataset id can be “DatasetA” “DatasetB” or “DatasetC”

Descriptions of files in each dataset:

DatasetA:

1. A_demultfile.csv : Demultiplexing file for DatasetA
2. A_N055.fasta: Complete fasta file for DatasetA
3. A_N055.fastq: Complete fastq file for DatasetA
4. DatasetA_Sanger_references.fas: Sanger references for DatasetA
5. 1_A_MAFFT_barcode_all_nt_megablast: megablast output of MAFFT barcodes used for amino acid correction
6. 1_A_MAFFT_barcode_all_nt_megablast.fasta: Accession fasta retrieved from 1_A_MAFFT_barcode_all_nt_megablast

DatasetB:

1. B_demultfile.csv: Demultiplexing file for DatasetA
2. B_N052.fasta: Complete fasta file for DatasetB
3. B_N052.fastq: Complete fastq file for DatasetB
4. REFERENCE_B_f.fasta: Illumina references for DatasetB, end1
5. REFERENCE_B_r.fasta: Illumina references for DatasetB, end2
6. 1_B_MAFFT_barcode_all_nt_megablast: megablast output of MAFFT barcodes used for amino acid correction
7. 1_B_MAFFT_barcode_all_nt_megablast.fasta: Accession fasta retrieved from 1_B_MAFFT_barcode_all_nt_megablast

DatasetC:

1. C_demultfile.csv: Demultiplexing file for DatasetC
2. C_N030.fasta: Complete fasta file for DatasetC
3. C_N030.fastq: Complete fastq file for DatasetB
4. DatasetC_References_illumina.fasta: Illumina references for DatasetC
5. 1_C_MAFFT_barcode_all_nt_megablast: megablast output of MAFFT barcodes used for amino acid correction
6. 1_C_MAFFT_barcode_all_nt_megablast.fasta: Accession fasta retrieved from 1_C_MAFFT_barcode_all_nt_megablast

QUICKSTART (for rapid generation - this uses 100X coverage only).

If running MAFFT and MAFFT+AA only: Datasets files to be in current folder under "DatasetA", "DatasetB" and DatasetC (file names used are for dataset A)

1. MAFFT

a. To get uncorrected barcodes

```
>>>python miniBarcoder.py -f DatasetA/A_N055.fasta -d DatasetA/A_demultfile.csv -o
DatasetA/1_A_MAFFT_barcode_100x -l 600
```

Note: Subsetting to 100 gave reasonable results: we have observed 486-489/490 barcodes with marginally higher error rates (substitution error upto 0.005% and indel errors ~0.75-0.78% in the few tests we have conducted). For exhaustive analyses as done for the publication run miniBarcoder.py with "-D 0" for all datasets: This however took >8 hours for Dataset A

b. For ease of downstream analyses copy output of previous step (all_barcodes.fa) in output directory to main directory

```
>>>cp DatasetA/1_A_MAFFT_barcode_100x/all_barcodes.fa DatasetA/1_A_MAFFT_barcodes.fasta
```

c. Filter barcodes by number of ambiguities (<=n), here 1% of >600 bp barcodes.

```
>>>python scripts/filter_by_Ns.py -i DatasetA/1_A_MAFFT_barcodes.fasta -n 6
```

Output MAFFT barcodes are in DatasetA/1_A_MAFFT_barcodes_Nfilter.fa

2. MAFFT+AA

a. run error correction on filtered barcodes

```
>>>python aacorrection.py -bf DatasetA/1_A_MAFFT_barcode_all_nt_megablast.fasta -bo
DatasetA/1_A_MAFFT_barcode_all_nt_megablast -b DatasetA/1_A_MAFFT_barcodes_Nfilter.fa -o
DatasetA/2_A_MAFFT_corr_barcodes.fasta
```

Output of MAFFT+AA in DatasetA/2_A_MAFFT_corr_barcodes.fasta

For dataset C of 313 bp: to miniBarcoder.py add "-m 1" and change to "-l 200" and "-D 0) (read subsetting not recommended as this is more error prone)
to aacorrection.py add "-l 300 -L 330"
to filter_by_Ns.py "-n 3"

IF RUNNING RACON:

3. RACON barcode

Batch script performing fastq retrieval, graphmap and racon. (See breakdown of steps for this dataset if any issues at the end of this document)

```
sh racon_consensus.sh DatasetA/A_N055.fastq DatasetA/A_N055.fasta DatasetA/1_A_MAFFT_barcode_100x
DatasetA/1_A_MAFFT_barcodes_Nfilter.fa DatasetA/3_A_racon_barcodes_all
```

Output is stored in DatasetA/3_A_racon_barcodes_all.fa

4. RACON+AA

run error correction on racon barcodes

```
>>>python aacorrection.py -bf DatasetA/1_A_MAFFT_barcode_all_nt_megablast.fasta -bo
DatasetA/1_A_MAFFT_barcode_all_nt_megablast -b DatasetA/3_A_racon_barcodes_all.fa -o
DatasetA/4_A_racon_barcodes_all_corr.fa
```

Output is stored in DatasetA/4_A_racon_barcodes_all_corr.fa

5. CONSOLIDATED BARCODES :

```
>>> python scripts/consolidate.py -m DatasetA/2_A_MAFFT_corr_barcodes.fasta -r
DatasetA/4_A_racon_barcodes_all_corr.fa -o DatasetA/5_A_mafft_racon_con.fa
```

Output is stored in DatasetA/5_A_mafft_racon_con.fa

IF INTERESTED IN ASSESSING CORRECTED BARCODES (requires MAFFT)
For MAFFT/RACON+AA barcodes:

```
python scripts/assess_corrbarcodes_wref.py -m DatasetA/2_A_MAFFT_corr_barcodes.fasta -r
DatasetA/DatasetA_Sanger_references.fas -t DatasetA/2_A_MAFFT_corr_barcodes_wrefs -o
DatasetA/2_A_MAFFT_corr_barcodes_wrefs_stats
```

Output is in DatasetA/2_A_MAFFT_corr_barcodes_wrefs_stats

IF INTERESTED IN ASSESSING UNCORRECTED BARCODES (requires dnadiff, part of MuMMER)
For MAFFT/RACON barcodes

```
python scripts/assess_uncorrbarcodes_wref.py -m DatasetA/1_A_MAFFT_barcodes_Nfilter.fa -r
DatasetA/DatasetA_Sanger_references.fas -t DatasetA/1_A_MAFFT_barcodes_Nfilt    er_wrefs -o
DatasetA/1_A_MAFFT_barcodes_Nfilter_wrefs_stats
```

Output is in DatasetA/1_A_MAFFT_barcodes_Nfilter_wrefs_stats

---

run_racon_consensus.sh.

This is batch script with performs the following steps. Names here are for Dataset A.

This batches the following or dataset A which can be separately conducted as:
a. Get trimmed fastq files for demultiplexed datasets corresponding to the fasta files

```
>>>python scripts/get_fastqs.py -fq DatasetA/A_N055.fastq -dr
DatasetA/1_A_MAFFT_barcode_100x/demreads -se
DatasetA/1_A_MAFFT_barcode_100x/A_N055.fasta_reformat_out_COIpred -o
DatasetA/1_A_MAFFT_barcodes_fastqs
```

b. Create separate fasta files from MAFFT uncorrected barcodes to use as references for graphmap and racon.

```
>>>python scripts/split_fasta_to_each.py -i DatasetA/1_A_MAFFT_barcodes_Nfilter.fa -o
DatasetA/1_A_refs_for_graphmap
```

c. Run graphmap in batch

```
>>>for f in DatasetA/1_A_refs_for_graphmap/*; do graphmap align --max-error 0.05 -r $f -d
DatasetA/1_A_MAFFT_barcodes_fastqs/"${f##*/}"stq -o DatasetA/1_A_MAFFT_barcodes_fastqs/"${f##*/}".sam
; done
```

d. Run Racon in batch

```
>>>for f in DatasetA/1_A_refs_for_graphmap/*.fa; do
nops="${f##*/}"stq;
nops="${nops%.fa*}";
racon --sam DatasetA/1_A_MAFFT_barcodes_fastqs/"$nops".fastq
DatasetA/1_A_MAFFT_barcodes_fastqs/"${f##*/}".sam $f
DatasetA/1_A_MAFFT_barcodes_fastqs/"${f##*/}"_racon.fasta;
done
```

e. Concatenate and clean up racon output

```
>>> cat DatasetA/1_A_MAFFT_barcode_all_fastqs/*racon.fasta | sed 's/Consensus_//g' >
DatasetA/3_A_racon_barcodes_all.fa
```