**NGS-Indel Coder v.1.0.0 Manual**

By Julien Boutte, PhD

Shannon Straub Lab (Hobart and William Smith Colleges, Geneva, NY, USA)

April 2019

**Contents:**

1. About NGS-Indel Coder, citation
2. Downloading NGS-Indel Coder, getting help
3. Input files format
4. NGS-Indel Coder command lines and options
5. Output files
6. Example
7. References
8. **About NGS-Indel Coder, citation**

NGS-Indel Coder was developed to detect and omit false positive indels inferred from assemblies of short read sequence data. This tool, divided in five parts proposed several options. This pipeline used several tools including 2MATRIX (Salinas and Little 2014) to code indels as binary characters and BLAST to detect exon positions (Altschul et al. 1990). Output files were generated for IQ-TREE software (Nguyen et al. 2015, Chernomor et al. 2016). Nevertheless, NGS-Indel Coder output fasta files can be used with any software coding indels using aligned fasta files.

List of python scripts:

1-parsing\_Samtools\_depth-files.py

2-Indel\_validation.py

3-Indel\_validation.py

4-Indel\_validation.py

5-indel\_deletion.py

6-IQTREE\_binary\_matrices\_creation.py

7-IQTREE\_DNA\_matrices\_creation.py

8-nexus\_files\_creation.py

9-identification\_boundaries.py

10-partitioned\_nexus\_files\_creation.py

11-Delete\_small\_partitions.py

When using NGS-Indel Coder please cite:

NGS-Indel Coder: A pipeline to code indel characters in phylogenomic data with an example of its application in milkweeds (*Asclepias*), Julien Boutte, Mark Fishbein, Aaron Liston, and Shannon C.K. Straub. Accepted in Molecular Phylogenetics and Evolution pending minor revisions.

1. **Downloading NGS-Indel Coder, getting help**

NGS-Indel Coder scripts can be downloaded from juboutte/NGS-Indel\_Coder/ - GitHub.

Mac OSX, Windows, and Linux environment are supported.

NGS-Indel Coder scripts require **python 2.7.12**.

2MATRIX software requires **perl**.

For all questions and concerns, please contact Dr. Julien Boutte.

Julien Boutte, Ph.D.

Postdoctoral Researcher Associate,

Department of Biology

Hobart and William Smith Colleges

300 Pulteney Street

Geneva, NY 14456 USA

Email: boutte.julien@gmail.com

Please use this subject tag: **NGS-Indel Coder Help**

1. **Input files format**

NGS-Indel Coder required several input files. At least one fasta file containing *n* aligned sequences and *n* read depth files (one read depth file per sequence in each aligned fasta file). More generally, NGS-Indel Coder used many aligned fasta files and one read depth file per sample studied. Aligned fasta files contained not necessary the same number of sequences.

**Warning**: Users can’t delete any nucleotide in the sequences after generating read depth files, but they can replace a nucleotide by a ‘N’ character. It is also possible to modify the alignment before using NGS-Indel Coder. Sequences can be modified prior to the creation of read depth files and aligned fasta files.

Example:

Considering 3 loci (locus\_1, locus\_2, locus\_3) and 5 species (species\_A, species\_B, species\_C, species\_D, species\_E).

locus\_1 contains species\_A, species\_B, species\_C and species\_D

locus\_2 contains species\_B, species\_C and species\_D

locus\_3 contains species\_A, species\_B, species\_C , species\_D and species\_E

Number of input files: (4+3+5) + 3 = 15 files corresponding to three aligned fasta files and 12 read depth files.

Read depth files can be generated using Samtools (Li et al. 2009). Command lines example is given part IV.

1. **NGS-Indel Coder command lines and options**

Relationships between the different parts of the pipeline are presented Figure 1.

Figure_1.pdf

Figure 1: Relationships between the different parts of the NGS-Indel Coder pipeline

1. Read Depth Files

Boutte et al. (2019) used BWA v.0.7.12 (Johnson et al. 2016), Samtools v.1.3.1 (Li et al. 2009) and Picard v.1.131 (http://broadinstitute.github.io/picard/) tools to create read depth files. The command lines are presented below.

bwa index Example\_species\_A.fasta

bwa mem Example\_species\_A.fasta Example\_species\_A\_1P.fastq Example\_species\_A\_2P.fastq > Example\_species\_A.sam

java -jar /opt/picard-tools/1.131/picard.jar NormalizeFasta \

I= Example\_species\_A.fasta \

O= Example\_species\_A\_cleaned.fasta

samtools faidx Example\_species\_A\_cleaned.fasta

java -jar /opt/picard-tools/1.131/picard.jar CreateSequenceDictionary R= Example\_species\_A\_cleaned.fasta O= Example\_species\_A\_cleaned.dict

java -jar /opt/picard-tools/1.131/picard.jar SortSam \

VALIDATION\_STRINGENCY=LENIENT \

I= Example\_species\_A.sam \

O= Example\_species\_A.bam \

SORT\_ORDER=coordinate

java -jar /opt/picard-tools/1.131/picard.jar AddOrReplaceReadGroups \

I= Example\_species\_A.bam \

O= Example\_species\_A\_cleaned.bam \

RGID=4 \

RGLB=lib1 \

RGPL=illumina \

RGPU=unit1 \

RGSM=20

java -jar /opt/picard-tools/1.131/picard.jar BuildBamIndex \

I= Example\_species\_A\_cleaned.bam

samtools depth -a Example\_species\_A\_cleaned.bam" > Example\_species\_A\_infos.txt

The **Example\_species\_A\_infos.txt** file contained all read depth information for the two samples of the species\_A (see part III). Users can parse this file using 1-parsing\_Samtools\_depth-files.py program. Usage: python 1-parsing\_Samtools\_depth-files.py Example\_species\_A\_infos.txt.

**Example\_species\_A.fasta** contained HybPiper supercontig sequences of the species\_A.

**Example\_species\_A\_1P.fastq** and **Example\_species\_A\_2P.fastq** correspond to cleaned paired reads.

It is possible to use single reads and references not generated by HybPiper (see BWA, Samtools and Picard manual or other software manuals that can produce read depth file).

1. Aligned Fasta files

This pipeline accepted any nucleotide aligned fasta files. However, sequences included in the aligned fasta files need to correspond to sequences used for the creation of read depth files. Nucleotides cannot be deleted. If users want to modify fasta files, please modify them prior to read depth file creation and alignment file creation.

**Warning (Part III - reminder)**: Users cannot delete any nucleotide in the sequences after generating read depth files, but they can replace a nucleotide by a ‘N’ character. It is also possible to modify the alignment before using NGS-Indel Coder. Sequences can be modified prior to the creation of read depth files and aligned fasta files.

1. Indel validation

This part includes four python scripts that will generate a final file necessary to generate temporary fasta files (see part iv). Command lines, temporary files, input files and outputfiles are indicated in the Example part.

Example:

python Scripts/2-Indel\_validation.py Example/I-input\_files/fasta\_files/Example.fasta Example/I-input\_files/depth\_files/

output file: temp\_part1.txt

move temp\_part1.txt to Example/II-temporary\_files

python Scripts/3-Indel\_validation.py Example/II-temporary\_files/temp\_part1.txt

output file: temp\_part2.txt

move temp\_part2.txt to Example/II-temporary\_files

python Scripts/4-Indel\_validation.py Example/II-temporary\_files/temp\_part2.txt 20

output file: MRD2\_MRD3\_T20.txt

move MRD2\_MRD3\_T20.txt to /II-temporary\_files

1. Indel deletion

To create indel binary files, NGS-Indel Coder generate temporary fasta files. These fasta files are only used to create indel binary files. They must not be used for phylogenomic analyses.

**Warning:** Fasta files are used only during the part v of the pipeline.

The script 5-indel\_deletion.py generates a temporary fasta file within which false putative indels are deleted.

Input files:

An aligned fasta file

MRD2\_MRD3\_TX.txt (generated during the Part iii, X=threshold selected)

Output file:

A temporary fasta file

Example:

python Scripts/5-indel\_deletion.py Example/II-temporary\_files/MRD2\_MRD3\_T20.txt Example/I-input\_files/fasta\_files/

output file: Example\_temp.fasta

move Example\_temp.fasta to Example/II-temporary\_files

1. Binary file creation

When the temporary fasta files is generated (part iv), it is possible to run 2MARTIX (Salinas and Little 2014) software to generate binary character files. Then, several python scripts are used to modify and generate final nexus and phylip files.

Input file:

The temporary fasta file generated during step iv.

Output file:

Three output files corresponding to IQTREE input files (.nex and .phy files).

Example:

perl 2matrix-master/2matrix.pl -i Example/II-temporary\_files/Example\_temp.fasta -n Example -o p

output file: Example.part and Example.phy

move Example.part and Example.phy to Example/II-temporary\_files

**Warning**: Example.part and Example.phy files are temporary files.

python Scripts/6-IQTREE\_binary\_matrices\_creation.py Example/II-temporary\_files /Example.phy Example/II-temporary\_files/Example.part

output file: Example\_indel.phy

**Example\_indel.phy is one of the three final files used by IQTREE software.**

move Example\_indel.phy to Example/II-temporary\_files if necessary.

perl 2matrix-master/2matrix.pl -i Example/I-input\_files/fasta\_files/Example.fasta -n Example2 -o p

output file: Example2.part and Example2.phy

move Example2.part and Example2.phy to Example/II-temporary\_files

python Scripts/7-IQTREE\_DNA\_matrices\_creation.py Example/II-temporary\_files/Example2.phy Example/II-temporary\_files/Example2.part

output file: Example2\_dna.phy

move Example2\_dna.phy to Example/II-temporary\_files if necessary.

**Example2\_dna.phy is one of the three final files used by IQTREE software.**

python Scripts/8-nexus\_files\_creation.py Example/II-temporary\_files/Example2\_dna.phy Example/II-temporary\_files/Example\_indel.phy MyFolder\_T20/

Comment: MyFolder\_T20/ corresponds to the folder that will contain Example2\_dna.phy and Example\_indel.phy when user will run IQTREE. This option is optional.

output file: Example2.nex

**Example2.nex is one of the three final files used by IQTREE software.**

move Example2.nex to Example/II-temporary\_files if necessary.

IQTREE command line example:

iqtree –nt 1 –bb 10000 –spp Example2.nex –m MFP+MERGE -AICc

Comment: Example2\_dna.phy and Example\_indel.phy are in the folder: MyFolder\_T20

Iqtree\_analyses/

Example2.nex

MyFolder\_T20/

Example2\_dna.phy

Example\_indel.phy

1. Exon/Intron

It is possible to identify intron and exon boundaries using a custom approach (using BLASTN results) to run a partitioned IQTREE analysis. This part of NGS-Indel Coder will replace the nexus file created step v.

Input file:

Output BLASTN result (database: transcript sequences used to create probes, query: the fasta file (part i)).

The X\_.nexus file created step v.

The aligned fasta file.

Output file:

A X.nexus output file (.nex file).

Example:

makeblastdb -in Example/II-temporary\_files/Example\_transcripts.fasta -out transcript\_db -dbtype nucl

blastn -query Example/I-input\_files/fasta\_files/Example.fasta -db transcript\_db -outfmt 7 -out res\_blastn.txt

**Warning: Example\_transcripts.fasta is not available in the example folder but res\_blastn.txt is available.**

output file: res\_blastn.txt

move res\_blastn.txt to Example/II-temporary\_files/

python Scripts/9-identification\_boundaries.py Example/II-temporary\_files/res\_blastn.txt Example

Comment: Example correspond to the name of your file (without extension, i.e. Example.fasta => Example)

output file: exon\_positions.txt

move exon\_positions.txt to Example/II-temporary\_files/

python Scripts/10-partitioned\_nexus\_files\_creation.py Example/II-temporary\_files/exon\_positions.txt Example/I-input\_files/fasta\_files/Example.fasta Example/II-temporary\_files/Example2.nex

output file: Example\_partition.nex

move Example\_partition.nex to Example/II-temporary\_files/

**Example\_partition.nex replaces Example2.nex. It is one of the three final files used by IQTREE software.**

IQTREE command line example:

iqtree –nt 1 –bb 10000 –spp Example\_partition.nex –m MFP+MERGE -AICc

Comment: Example2\_dna.phy and Example\_indel.phy are in the folder: MyFolder\_T20

Iqtree\_analyses/

Example\_partition.nex

MyFolder\_T20/

Example2\_dna.phy

Example\_indel.phy

1. Small partition deletion

Prior to run step vi, it is possible to delete small partitions ≤ x bp (100 bp by default). In this case, it is necessary to generate a new DNA phylip file (Example2\_dna.phy) and/or a new Binary phylip file.

Input file:

Output BLASTN result (database: transcript sequences used to create probes, query: the fasta file (part i)).

Initial fasta file

Output file:

An aligned fasta file cleaned (small partitions deleted).

Example:

makeblastdb -in Example/II-temporary\_files/Example\_transcripts.fasta -out transcript\_db -dbtype nucl

blastn -query Example/I-input\_files/fasta\_files/Example.fasta -db transcript\_db -outfmt 7 -out res\_blastn.txt

**Warning: Example\_transcripts.fasta is not available in the example folder but res\_blastn.txt is available.**

mv res\_blastn.txt to Example/II-temporary\_files/

python Scripts/9-identification\_boundaries.py Example/II-temporary\_files/res\_blastn.txt Example

Comment: Example correspond to the name of your file (without extension, i.e. Example.fasta => Example)

move exon\_positions.txt to Example/II-temporary\_files/

Comment: if users already ran step vi, it is possible to start line: python Scripts/ 11-Delete\_small\_partitions.py Example/II-temporary\_files/exon\_positions.txt Example/I-input\_files/fasta\_files/Example.fasta.

11-Delete\_small\_partitions.py Example/II-temporary\_files/exon\_positions.txt Example/I-input\_files/fasta\_files/Example.fasta Example\_partitioned.fasta 100

Comments: Example\_partitioned.fasta is the name of the new fasta created (don’t forget “.fasta”). 100 is the length to consider a partition as a small partition.

output file: Example\_partitioned.fasta

move Example\_partitioned.fasta to Example/II-temporary\_files/

This new fasta file could be used as the initial fasta file (user can restart steps i to v/vi). Users can also replace only the DNA .phy file and the .nex file associated and conserve the Binary .phy file create prior to delete small partition (recommended).

1. **Output files**

NGS-Indel Coder pipeline generates several output files. The most important files correspond to two .phy and one .nex files to run IQ-TREE analyses. See part IV. NGS-Indel Coder command lines and options.

1. **Example**

The Example folder contains input files, intermediate files and output files generated using one threshold: T20. All the example command lines are indicated part IV.

Input files:

Aligned fasta file:

Example.fasta

Read depth files:

Asclepiasaffstandleyi1321-10025.txt, Asclepiasquadrifolia1387-10025.txt, Asclepiasalbicans003-10025.txt, Asclepiasscaposa977-10025.txt, Asclepiasamplexicaulis1401-10025.txt, Asclepiassolanoana256-10025.txt, Asclepiasarenaria1322-10025.txt , Asclepiasstenophylla1417-10025.txt, Asclepiascoulteri823-10025.txt, Asclepiassubulata423-10025.txt, Asclepiascurtissii609-10025.txt, Asclepiastuberosa1403-10025.txt, Asclepiaselata856-10025.txt, Asclepiasvirletii476-10025.txt, Asclepiasemoryi952-10025.txt, Gomphocarpusphysocarpus957-10025.txt, Asclepiaserosa70-10025.txt, Stathmostelmafornicatum72012-10025.txt, Asclepiashirtella1399-10025.txt, euphorbiifolia72014-10025.txt, Asclepiaslongifolia940-10025.txt, linaria720006-10025.txt, Asclepiasnyctaginifolia584-10025.txt, puberula72004-10025.txt, Asclepiasoenotheroides1325-10025.txt, viridis72007-10025.txt, Asclepiasperennis500-10025.txt

Output files (temporary files included, names chosen during example phase). Final files are indicated in red:

|  |  |  |
| --- | --- | --- |
| Example.part  Example.phy  Example\_partitioned.fasta  Example2.nex  Example2.part  Example2.phy | Example2\_dna.phy  Example\_indel.phy  Example\_partition.nex  Example\_temp.fasta  MRD2\_MRD3\_T20.txt  exon\_positions.txt | res\_blastn.txt  temp\_part1.txt  temp\_part2.txt |

1. **References**

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. J. Mol. Biol. 215, 403–410. https://doi.org/10.1016/S0022-2836(05)80360-2

Boutte, J., Fishbein, M., Liston, A., Straub, S.C.K, Accepted in Molecular Phylogenetics and Evolution pending minor revisions.

Chernomor, O., von Haeseler, A., Minh, B.Q., 2016. Terrace Aware Data Structure for Phylogenomic Inference from Supermatrices. Syst. Biol. 65, 997–1008. https://doi.org/10.1093/sysbio/syw037

Johnson, M.G., Gardner, E.M., Liu, Y., Medina, R., Goffinet, B., Shaw, A.J., Zerega, N.J.C., Wickett, N.J., 2016. HybPiper: Extracting Coding Sequence and Introns for Phylogenetics from High-Throughput Sequencing Reads Using Target Enrichment. Appl. Plant. Sci. 4, 1600016. https://doi.org/10.3732/apps.1600016

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., 1000 Genome Project Data Processing Subgroup, 2009. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 25, 2078–2079. https://doi.org/10.1093/bioinformatics/btp352

Nguyen, L.-T., Schmidt, H.A., von Haeseler, A., Minh, B.Q., 2015. IQ-TREE: A Fast and Effective Stochastic Algorithm for Estimating Maximum-Likelihood Phylogenies. Mol. Biol. Evol. 32, 268–274. https://doi.org/10.1093/molbev/msu300

Salinas, N.R., Little, D.P., 2014. 2matrix: A Utility for Indel Coding and Phylogenetic Matrix Concatenation. Appl. Plant. Sci. 2, 1300083. https://doi.org/10.3732/apps.1300083

Saurabh, K., Holland, B.R., Gibb, G.C., Penny, D., 2012. Gaps: An Elusive Source of Phylogenetic Information. Systematic Biology 61, 1075–1082. https://doi.org/10.1093/sysbio/sys043