PMiFish (MiFish パイプライン) マニュアル

Version 2.4.1 (2020/7/20)

0. 環境の準備

＜Windows の場合＞

・ActivePerl のインストール　https://www.activestate.com/activeperl

・Gzip のインストール　http://gnuwin32.sourceforge.net/packages/gzip.htm

※Gzip はインストール時に自動で PATH が通らないので、その設定をする必要があります。

コンピューター→システムの詳細設定→詳細設定タブ中にある環境変数→変数 Path を選択し、編集　→　最後に Gzip.exe が含まれるフォルダの場所（例　C:¥Program Files (x86)¥GnuWin32¥bin）を付け足し保存

・Usearch_v11（https://www.drive5.com/usearch/download.html）をダウンロードし Tools フォルダに入れる。

※PMiFish を実行した際に、"Can't spawn …"のようなエラーメッセージが出てきた場合、付録 3 をご確認ください。

＜Mac の場合＞

・Mac 版の Usearchv11（https://www.drive5.com/usearch/download.html)をダウンロードし Tools フォルダに入れる。

※パイプライン実行中に permission denied と言われたら、ターミナル上で chmod +x usearch 名を実行する。

※最新の Mac の OS では、32bit 版の Usearch は動かないとのこと。有料版の 64bit 版を購入するか、昔の OS で動かすしかありません。

◯系統解析をする場合

PMiFish ではオプションで科ごとに系統樹を作成することができます。作成には MEGAX の Command Line 版が必要となります。

・megacc（MEGA の Command Line 版）をインストール　http://www.megasoftware.net/

※megacc はアラインメント・系統樹作成に使用

解析の前に MEGA を起動し、PROTOTYPE モード（MEGAX の場合、起動後の画面で右下に ANALYZE か PROTOTYPE か選択できます）で、アラインメント及び系統樹の設定ファイルをそれぞれ作成する（詳しくは MEGAX のマニュアル参照)。作成したファイル（.mao)を Tools フォルダに入れる。

1．フォルダ内の構造

DataBase・・・・リファレンスデータ（fasta 形式）と primer 配列を記入したファイル
Dictionary ・・・日本語、英語の種名辞書データ、および科名辞書
Results ・・・・結果が出力されるフォルダ
Run ・・・・・・解析したい fastq ファイルもしくは gz ファイルを入れるフォルダ
Scripts・・・・・ステップごとのスクリプトが入っているフォルダ（ステップごとに解析したい時に使用）
Tools ・・・・・Usearch および MEGA の設定ファイル(.mao)を入れておくフォルダ
PMiFish.pl・・・解析を走らせるスクリプト
PA_with_DB.pl・PMiFish.pl の解析後に実行することで、科レベルの系統樹を作成
結果は、5-2 Summary_table 内に出力される
Setting.txt・・・各種設定を記入するファイル

2　解析方法

STEP1　Setting.txt で各種設定を確認（毎回する必要はありません）
STEP2　解析したいファイル（fastq or gz）を Run フォルダに入れる。
STEP3　PMiFish.pl がある階層でコマンドプロンプト(Windows10 は PowerShell、Mac の場合はターミナル)を立ち上げ（Windows の場合フォルダ内のなにもない場所で Shift+右クリック→コマンドウィンドウをここで開く)、
perl　PMiFish.pl を入力
※ステップごとに解析をしたい場合は、PMiFish.pl がある階層で
perl　./Scripts/＿＿＿＿＿.pl を実行
STEP4　結果が Results フォルダに出力される。

3　出力結果

※Setting.txt で"Temporary = YES"にすると 1-1 から 1-4 の fq ファイルは自動で消去されます。(1-1 から 1-4 で作成されるファイルは容量が大きいため削除をお勧めします)
※Setting.txt で"Compress = YES"にすると fastq を gz に圧縮します。(ただし、圧縮に時間がかかります)

1－1	Merge_paird_reads	R1 と R2 をつなげたファイル
1－2	Strip_primers	Primer を除去したファイル
1－3	Quality_filter	Quality_Check をしたファイル

1－4　Rarefaction　　Rarefaction を実施したファイル
（＊Setting.txt で Rarefaction = 数字もしくは MIN、Timing=1 の場合作成）

2－1　Find_Unique　　同じ配列をもつ Read をまとめたファイル
2－2　Denoise　　PCR エラーやキメラ配列のチェック
_unoise3_result.txt・・・Denoise の結果ファイル
_zotu.fa ・・・・・・・・PCR エラーの配列を除去したファイル（キメラは含まれる）
2－3　Separate_chimera　　キメラ配列とそれ以外の配列を分けたファイル
_zotu_chimeras.fa・・・・キメラ配列
_zotu_nonchimeras.fa ・・キメラ配列を除去した配列
2－4　Rarefaction　　Rarefaction を実施したファイル
（＊Setting.txt で Rarefaction = 数字もしくは MIN、Timing= 2 の場合作成）

3－1　Usearch_global　　Usearch_global の結果ファイル

4－1　Annotation　　Usearch_global の結果をまとめた一連のファイル
_all_annotated_seq.fas・・サンプルのすべての Unique 配列
_Detail.html・・・・・・・結果をまとめた詳細
_Representative_seq.fas ・・代表配列（もっとも Read 数が多いもの）をまとめたファイル
_Summary.txt・・・・・・結果をまとめた概要
_Synonym_list.html・・・・リファレンスデータ中に同じ配列を持つ異なる種リスト

5－1　Fasta_for_Phylogenetic_Analysis
各サンプルの代表配列をまとめた一連のファイル
all_representative_seqs.fas・・・すべてのサンプルから代表配列をまとめたファイル
merged_list.txt・・・・・・・・同じ配列をまとめたときのリスト
merged_seq.fas ・・・・・・・ all_representative_seqs.fas で同じ配列をまとめたファイル
merged_seq_with_family_name.fas・・・・merged_seq.fas に科名が付加されたもの

５－２　Summary_table　全サンプルの結果をまとめた表
cluster_list_of_U 数字.txt・・・・・クラスタリングを行った結果
Summary_table.tsv・・・・・・・・各サンプルをまとめた表

５－３　Fasta_classified_by_Family　科ごとにまとめた Fasta ファイル
list.txt・・・・・・どのファイルに何種含まれるかをまとめたリスト
___.fas・・・・・・科ごとの fasta ファイル(N.A.fas は科名が不明のもの)

６－２　Phylogenetic_trees
科ごとに系統樹を作成したファイル（Setting.txt で Phylogeney = YES の場合作成）
___.meg ・・・・・・・・・アラインメントしたファイル
___.nwk ・・・・・・・・・系統樹

log.html	各ステップ後のリード数をまとめた html ファイル
Portal.html	結果をまとめた html ファイル

◯更新履歴

■version2.4 (2018/12/15)

・1_2_Strip_primers で Primer 配列を認識して削除もできるように変更。これにより、複数のプライマーで増幅したリードが含まれるサンプルにも対応
・Rarefaction のタイミングを Denoise 後でもできるように変更
・ポータルに log、Summary_Table、Representative_Sequence へのリンクを作成
・削除するユニーク配列のリード数を固定値だけではなく、割合（Find_unique 後の総リード数に対する割合）でも指定できるように変更（Setting.txt の Depth Filter で設定）

■version2.3 (2018/07/09)

・相同性が 97%以下（数値は Setting.txt の UIdentity による）のものについておこなうクラスタリングの方式を変更(Usearch の-cluster_smallmem 使用)。これまでと結果は変わらないが、速度向上。
・2_1_Find_unique で配列数が 0 になると、2_2_Denoise が進まないエラーを修正。

■version2.2 (2018/07/01)

・すでにプライマー領域を削除している fastq ファイルも解析できるように変更
（Setting.txt の Primers を No にする）

・Denoise で削除していたエラー配列を、予測した真の配列のリード数に加えるオプションを追加
（Setting.txt の Correct_error を YES にする。NO の場合エラー配列のリード数はカウントされない）

・Portal からいける Detail に Accession NO.を追加（NCBI へのリンクが張ってある）

・Portal に解析を初めた時間・終了した時間を記載するように変更

■version2.2 以前の変更点

・Usearchv8 から Usearchv10 へ変更

・系統樹作成以外の作業を Usearch のみで実行するように変更

・Usearchv10 の unoise3 オプションを使うことで、PCR エラーやキメラ配列を除去

・これまで 97%以下（数値は Setting.txt の UIdentity による）の相同性で同じ種名と判断されたものはひとつにまとめられていましたが、同じ種名のものでクラスタリングを実行（クラスタリングの基準は UIdentity と同じ。複数種が含まれる可能性があるため実施）

・Portal の形式の変更

・log を出力するように変更

＜作成者＞
後藤　亮
千葉県立中央博物館
rogotoh@chiba-muse.or.jp

付録１　-Setting.txt の内容について-

※実際に解析に使用される部分を赤文字で示す。＃から始まる行は説明文。

※ファイル名にスペースは使用できない。

＃リファレンスファイルの指定

＃DataBase フォルダ中にあるリファレンスファイル（fasta フォーマット）を指定する

DB = ファイル名

例）DB = MiFish_DB.fas

#プライマーの設定

#DataBase フォルダ中にあるプライマー配列が記入されたファイルを指定する

#すでにプライマー配列が削除されている fastq ファイルを解析する場合は”NO”

Primers =ファイル名 or NO

例 1）Primers = Primers.txt

例 2）Primers = NO

#プライマー領域中に最大いくつのミスマッチを許容するか指定する

#プライマーの長さのみを利用して、プライマー領域を削除する場合は”length”

MaxDiff = 数値 or length

例 1 ）MaxDiff = 2

例 2 ）MaxDiff = length　＃ 1 つのプライマーペアしか使用してないならこっちがお勧め

#複数のプライマーペアで増幅されたリードが含まれる場合、それを分けるかどうか

#Yes の場合、以降の解析は別々に解析される。

Divide = YES or NO

例 1 ）分けて解析する場合　Divide= YES

#Rarefaction の設定

#リード数を希釈する場合は、数値もしくは MIN。しない場合は NO

Rarefaction = 数値 or MIN or NO

例 1）リード数を 10000 にする場合　Rarefaction = 10000

例 2）リード数を解析するサンプル中でもっとも少ないものに合わせる場合

Rarefaction = MIN

例 3 ）リード数を希釈しない場合　Rarefaction = NO

#Rarefaction を実行するタイミングを設定する

Timing = 1 or 2

例 1）Quality filter の後に実行する場合　Timing = 1

例 2）Denoise の後に実行する場合　　Timing = 2

#解析に使用する配列の長さを指定する

Length = 数値

例 1）50bp 以上の配列を使用する場合　Length = 50

#同じ配列がいくつあれば解析にしようするかを設定する

Depth = 数値 or 百分率

例 1）同じ配列が 4 以上あるものを使用する場合　Depth = 4

例 2）百分率を指定した場合、各サンプルの全リード数（2_1_Find_unique 後）に対して、指定した百分率から算出された値が Depth に設定される

Depth = 0.01%

この場合、全リード数が 10000 のときは、Depth = 1 (10000×0.0001)に、

全リード数が 100000 のときは、Depth = 10 (100000×0.0001)に設定される

#Denoise でエラーとされた配列を補正して以降の解析に使用するかどうか

Correct_error = YES or NO

例 1）補正して使用する場合　Correct_error = YES

例 2）容赦なく捨てる場合　　Correct_error = NO

#相同性の設定

UIdentity = 数値

例 1）相同性が 97%以上のものを採用する場合　UIdentity = 97

#拾い上げる最低の相同性の設定

LIdentity = 数値

例 1）相同性が 80%より低いものは不明とする　LIdentity = 80

#標準名の辞書を設定（辞書は Dictionary フォルダ内に置く）
Dictionary = ファイル名 or NO
例１）Dictionary = Sname_Jname.togodb.nonredundant.txt
例２）使用しない場合　Dictionary = NO

#科名辞書を設定（辞書は Dictionary フォルダ内に置く）
Family = ファイル名 or NO
例１）Family = Family_name_Fish.txt
例２）使用しない場合　Family = NO

#科ごとに系統樹を作成するかどうか
#作成するには MEGAX が必要
Phylogenetic = YES or NO
例１）使用する場合　　Phylogenetic = YES
例２）使用しない場合　Phylogenetic = NO

#Preprocessing のファイルを解析後削除するかどうか
#ファイル容量が膨大になるので通常は YES をお勧めします。
Temporary = YES or NO
例１）削除する場合　　Temporary = YES
例２）削除しない場合　Temporary = NO

#fq ファイルを gz ファイルに圧縮するかどうか
#圧縮には時間がかかるので通常は NO をお勧めします。
Compress = YES or NO
例１）圧縮する場合　　Compress = YES
例２）圧縮しない場合　Compress = NO

付録２　-Primer.txt の内容について-

＃DataBase フォルダ内に保存します
＃プライマーペア名は Forward と Reverse で必ず同じにする必要があります
＃プライマーの名前にスペースは使用できません
＃解析に使用しないプライマーは削除するか＃（半角）を最初につける（削除すると再び使う時に面倒なので＃を使うことをお勧めします）必要があります
＃縮重プライマーも使用可能です
＃順番が上のプライマーが優先されます。例えば、許容する置換数（Setting.txt の MaxDiff で設定）をクリアするプライマーペアが複数あった場合、順番が上のプライマーペアで増幅したものとみなされます。
＃プライマー配列中の indel は考慮しませんが、5‘末端側は± 1 塩基の長さを許容します（結果の中で Shifted と表記されるリード数がこれに該当します）

例）
Forward primers
>Primer1
NNNNNNATTTCGATGTRGTAAGTC
>Primer2
NNNNNNAATCCATGATTCCCGTA
#>Primer3
#NNNNNNCCCGTAGCTTTAAAAGCGC

Reverse primers
>Primer1
NNNNNNGTATTTACTAGTYAAACC
>Primer2
AAGCTGATGGATGGGAAAT
#>Primer3
#AAAGCGGATCTGAAGTAR

上記の場合、＃が先頭にある Primer3 は解析に使われません。

付録３　-トラブルシューティング-

＃Windows でうまく動かない

64bit のパソコンを使い、32bit 版の Usearch を動かそうとした場合、たまに“Can’t spawn…”というエラーメッセージが出て動かないことがあります。この場合下記のマイクロソフトのサイトから「Visual Studio 2015 の Visual C++ 再頒布可能パッケージ 」をダウンロードしてインストールしてください。

https://www.microsoft.com/ja-jp/download/details.aspx?id=48145