

```

#!/usr/local/bin/perl -w

use strict;
use warnings;

#use lib '/project/rodent/medaka_anja/pscripte';
#use ClusterFunc;

my($USAGE) = "$0 ***Dateiname***\n";

unless (@ARGV) {
    print $USAGE;
    exit;}

my $filename = $ARGV[0];

unless (open(GET_DATA,$filename)) {
    print STDERR "Cannot open file\"$filename\"\n\n";
    exit;}

#outputfile
#my $outputfilename = 'AnjaBlast';
#open(OUTPUT,">$outputfilename");

my $data = '';
#my $clone = '';
#my $short = 0;
#my $AC1 = '';

#while ($data = <GET_DATA>){
#   if ($data !~ /^ /){
#       $data =~ s/\n//;$data =~ s/^s+//;
#       if ($short){$AC1 = $data;$short = 0;}
#       else {
#           $clone = $data;
#           $data = <GET_DATA>;
#           $data =~ s/\n//;$data =~ s/^s+//;
#           $AC1 = $data;}
#       $data = <GET_DATA>;
#       $data =~ s/\n//;$data =~ s/^s+//;
#       if ($data =~ /data/){print "$clone\n $AC1\n $data\n";}
#       else {$short = 1;$clone = $data;}}
#}

#my $fileMG = 0;
my $fileACE = 0;
my $fileS = 0;
my @temp;
my $temp = '';
my $pattern = 'read: McF';

while ($data = <GET_DATA>){#in CloneAC_2ndedit lesen, für jedes
*.data nachschauen, wo abgeblieben
    if ($data =~ /found/){
        $data =~ s/\n//;
        @temp = split(/\s+/, $data);#MedGastrula
        # $temp[0] =~ s/^>//;
        $temp = translate_rearray($temp[4]);#McF
        # $fileACE = lookup_ACEcluster($temp);
        if ($temp !~ /^0$/){$fileACE = lookup_ACEcluster($temp);
            if (length($fileACE) < 2){$fileACE = lookup_singletons
($temp);}}
        # print "$data $fileACE $fileS\n";
        # $fileACE = lookup_singletons($temp[0]);

```

```

        print "$data $temp $fileACE\n";
    # $fileS = 0;
    # $fileACE = 0;
    }
    else {print "$data";}}

#Klone finden, fuer die unterschdl. ACs in versch. Contigs sind
#my @temp;
#my @temp2;
#my $num = 0;
#my $i = 0;
#my $data2nd = '';
#my $clone = '';
#my $notsame = 1;
#while ($data = <GET_DATA>){
# $data =~ s/\n//;$data =~ s/^\s+//;
# @temp = split(/\s+/, $data);
# if (scalar(@temp) < 4){$clone = $data;}
# else {
#     $data2nd = <GET_DATA>;
#     $data2nd =~ s/\n//;$data =~ s/^\s+//;@temp2 = split(/\s+/,
# $data2nd);
#     for ($i = 1;$i < 4; $i++){
#         if (!($temp[$i] eq $temp2[$i])){$notsame = 1;}}
#     if ($notsame){print "$clone\n $data\n $data2nd\n";$notsame =
# 0;}}
#}

exit;

sub lookup_MGcluster{

my ($AC) = @_ ;

$filename = 'AllData_clusters';
#$filename = 'test_CloneAC';
unless (open(GET_MG,$filename)) {
    print STDERR "Cannot open file\"$filename\"\n\n";
    exit;}

my $dataMG = '';
my @temp;
my $AC_temp = '';
my $cluster = '';
while ($dataMG = <GET_MG>){
    if ($dataMG =~ /^>/){
        $dataMG =~ s/\n//;
        $dataMG =~ s/^\s+//;
        @temp = split(/\s+/, $dataMG);
        $cluster = $temp[0];}
    else {
        @temp = split(/\s+/, $dataMG);
        foreach $AC_temp(@temp){
            if ($AC_temp =~ /$AC/i){return $cluster;}}}
}
return 0;
}#sub

sub lookup_ACEcluster{

my ($AC) = @_ ;

#$filename = 'ACE_tgicl.edit_bd2edit';
#$filename = 'TABLE.clusters-reads.bd2edit';

```

```

#$filename = 'All_OFP.trans';
#$filename = 'my1A.res';
#$filename = 'ACE_plates21to55.edit';
#$filename = 'ACE_platesAll.edit';
$filename = 'ACE_plates1to20.edit';

unless (open(GET_ACE,$filename)) {
    print STDERR "Cannot open file\"$filename\"\n\n";
    exit;}

my $dataACE = '';
my $cluster = '';
my @temp;
my $output_cluster = '';
my $pattern = 'read: McF';
my $cluster_patt = 'cluster: cluster';

while ($dataACE = <GET_ACE>){
    if (($dataACE =~ /^RD/) or ($dataACE =~ /^CO/)) {
# if (($dataACE =~ /$pattern/) or ($dataACE =~ /$cluster_patt/)){
# if ($dataACE =~ />/){
# if (($dataACE =~ /\#/) or ($dataACE =~ /Med/)){
    $dataACE =~ s/\n//;
    @temp = split(/\s+/, $dataACE);
    if ($dataACE =~ /^CO/){
        $cluster = "$temp[1]";
        # $cluster =~ s/>//;
    }
    else {
# if ($temp[4] =~ /Gastrula/){$temp[4] =~ s/Gastrula//;}
# $temp[1] =~ s/bd\d//;$AC =~ s/bd\d//;
        if ($temp[1] =~ /$AC/i){
            return $cluster;}}}
    }
return '0';
}#sub

sub lookup_singletons{

my ($AC) = @_ ;

#$filename = 'seqs_fasta_plate1to55.singletons';
#$filename = 'TABLE.singletons-reads.bd2edit';
#$filename = 'AllHiQualtraces_tgicl.singletons_bd2edit';
#$filename = 'ACE_plates21to55_singl';
#$filename = 'ACE_all_singl';
$filename = 'AllHiQualtraces_plates1to20.masked.singletons';

unless (open(GET_SI,$filename)) {
    print STDERR "Cannot open file\"$filename\"\n\n";
    exit;}

my $dataSI = '';
my $pattern = 'read: McF';
while ($dataSI = <GET_SI>){
    if ($dataSI =~ /^McF/){
        $dataSI =~ s/\n//;
        # $dataSI =~ s/bd\d//;$AC =~ s/bd\d//;
        if ($dataSI =~ /$AC/i){return 'Singleton';}}}
return 0;
}#sub

sub translate_rearray{

```

```
my ($Mc) = @_ ;
my $data = '';
my @temp;
#my $filename = 'All_Rearray';
my $filename = 'AllHiQualtraces.names.edit';

unless (open(GET_RE,$filename)) {
    print STDERR "Cannot open file\"$filename\"\n\n";
    exit;}

while ($data = <GET_RE>){
    @temp = split(/\s+/, $data);
    # $Mc =~ s/bd\d//;$temp[1] =~ s/bd\d//;
    if ($Mc eq $temp[0]) {
        close(GET_RE);
        # if ($temp[0] =~ /Gastrula/) {$temp[0] =~ s/Gastrula//;}
        return $temp[1];}
}#while
return "0";
}#sub
```