

Supplementary Text. Three Perl scripts that were used for the analysis described in the Methods. The name of each program is printed in bold.

Program 1: gene_movement_donor_sites

```
#!/usr/bin/perl

use lib '/home/wicker/WICKERSoft/';
use Biomodule;

if (@ARGV <2) {
    print "\n1.arg = all_grass_colinear_genes_list_final\n";
    print "2.arg = <species>_non_colinear_genes_list\n";
    exit;
}

open (IN, "$ARGV[1]");
@global = <IN>;
close(IN);

#open(OUT, ">donor_site_list");

open(IN, "$ARGV[0]");

while (<IN>) {

    ($bdid, $bdpos, $osid, $ospos, $sbid, $sbpos, $rem) =
/^(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S.*)/;
    if ($bdid) {

        ($osch, $osgen) = ($osid =~ /Bradi(\d+)g(\d+)/);
        $osgen =~ s/^[0]+//g;

        $donor_yes = 0;
        $entry = "\nTEST:$bdid\t$bdpos\t$osid\t$sbid\t$rem\n";

        open (IN2, "$ARGV[1]");

        while (<IN2>) {

            ($bdid2, $bdpos2, $osid2, $ospos2, $sbid2, $sbpos2, $rem2) =
/^(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S.*)/;

            if ($bdid2) {
                ($osch2, $osgen2) = ($osid2 =~ /Bradi(\d+)g(\d+)/);
                $osgen2 =~ s/^[0]+//g;

                ($sbch2, $sbgen2) = ($sbid2 =~ /Sb(\d+)g(\d+)/);
                $sbgen2 =~ s/^[0]+//g;

                $diff_os = abs($osgen-$osgen2);
                $diff_sb = abs($sbgen-$sbgen2);

                $hit = '';
                if ($osid eq $osid2) { $hit = "+";}

                if ($sbid eq $sbid2) { $hit .= "-";}

                if ((($osch2 eq $osch) && ($diff_os < 1000)) || (($sbch2 eq
$sbch) && ($diff_sb < 1000))) {

                    $donor_yes = 1;
                }
            }
        }
    }
}
```

```

        $entry .= "$bdid2\t$bdpos2\t$osid2\t$sbid2\t$hit\n";
    }

}

}

close(IN2);

if ($donor_yes ==1) {
    print "$entry\n";
}

}

}

```

Program 2: gene_movement_mk_colinear_gene_list

```

#!/usr/bin/perl

use lib '/home/wicker/WICKERSsoft/';
use Biomodule;

if (@ARGV <1) {
    print "\n1.arg = global synteny table\n";
    exit;
}

$ch = $ARGV[1];

open(IN, "$ARGV[0]");

while (<IN>) {
    ($bdid, $bdpos, $osid, $ospos, $sbid, $sbpos, $rem) =
/^(\S+)\s+(\S+)\s+(0s\S+)\s+(\S+)\s+(Sb\S+)\s+(\S+)\s+\d+\s+(\S.*)/;
    if ($bdid) {

        ($bdch, $bdgen) = ($bdid =~ /\..(\d+)g(\d+)/);
        ($osch, $osgen) = ($osid =~ /\..(\d+)g(\d+)/);
        ($sbch, $sbgen) = ($sbid =~ /\..(\d+)g(\d+)/);

        $osch =~ s/^0//;
        $sbch =~ s/^0//;
        $bdch =~ s/^0//;

        $bdgen =~ s/^[0]+//;
        $osgen =~ s/^[0]+//;
        $sbgen =~ s/^[0]+//;

        &adjust_chr;

        &match_chr;

        $all_synt = 0;

        #if (($os_sb_synt ==1) && ($os_bd_synt ==0) && ($bd_sb_synt ==0)) {

        if ($os_bd_synt ==1) {

```

```

        print "$bdid\t$bdpos\t$osid\t$ospos\t$sbid\t$sbpos\t$rem\n";
    }

}
}

```

```

# adjust ch numbers to compensate for Sbic chromosome fusions
sub adjust_chr {

```

```

# assign virtual Sbic chromosomes 11 + 12
if (($sbch == 1) && ($sbgen > 16880) && ($sbgen < 31290)) {
    $sbch = 11;
}

```

```

if (($sbch == 2) && ($sbgen > 11385) && ($sbgen < 32815)) {
    $sbch = 12;
}

```

```

# assign virtual Bdis chromosomes 6 - 12

```

```

#Bd1
if (($bdch == 1) && ($bdgen >= 16450) && ($bdgen < 29130)) {
    $bdch = 7;
}

```

```

if (($bdch == 1) && ($bdgen >= 52490) && ($bdgen <= 59820)) {
    $bdch = 7;
}

```

```

if (($bdch == 1) && ($bdgen >= 29130) && ($bdgen < 52490)) {
    $bdch = 6;
}

```

```

#Bd2
if (($bdch == 2) && ($bdgen >= 14080) && ($bdgen < 40150)) {
    $bdch = 12;
}

```

```

# Bd3
if (($bdch == 3) && ($bdgen >= 12460) && ($bdgen < 20730)) {
    $bdch = 8;
}

```

```

if (($bdch == 3) && ($bdgen >= 34610) && ($bdgen <= 43080)) {
    $bdch = 8;
}

```

```

if (($bdch == 3) && ($bdgen >= 20730) && ($bdgen < 34610)) {
    $bdch = 10;
}

```

```

# Bd4
if (($bdch == 4) && ($bdgen >= 8170) && ($bdgen < 8970)) {
    $bdch = 9;
}

```

```

if (($bdch == 4) && ($bdgen >= 26920) && ($bdgen < 39020)) {
    $bdch = 9;
}

```

```

if (($bdch == 4) && ($bdgen >= 8970) && ($bdgen < 26920)) {
    $bdch = 11;
}

# define syntenic pairs
@os_sb_pairs = qw (1-3 2-4 3-1 4-6 5-9 6-10 7-2 8-7 9-12 10-11 11-5 12-8);
@os_bd_pairs = qw (1-2 2-3 3-1 4-5 5-12 6-6 7-7 8-8 9-9 10-10 11-11 12-4);
@bd_sb_pairs = qw (1-1 2-3 3-4 5-6 6-10 7-2 8-7 9-12 10-11 11-5 12-9);
} # end adjust_chr -----

# check for chromosome synteny ++++++
sub match_chr {
    $os_sb_synt = 0;
    SYNT: foreach $pair (@os_sb_pairs) {
        ($os,$sb) = ($pair =~ /(\d+)-(\d+)/);
        if (($osch == $os) && ($sbch == $sb)) {
            $os_sb_synt = 1;
            last SYNT;
        }
    }

    $os_bd_synt = 0;
    SYNT: foreach $pair (@os_bd_pairs) {
        ($os,$bd) = ($pair =~ /(\d+)-(\d+)/);
        if (($osch == $os) && ($bdch == $bd)) {
            $os_bd_synt = 1;
            last SYNT;
        }
    }

    $bd_sb_synt = 0;
    SYNT: foreach $pair (@bd_sb_pairs) {
        ($bd,$sb) = ($pair =~ /(\d+)-(\d+)/);
        if (($bdch == $bd) && ($sbch == $sb)) {
            $bd_sb_synt = 1;
            last SYNT;
        }
    }

} # end match_chr -----

```

Program 3: gene_movement_mk_colinear_gene_list2

```
#!/usr/bin/perl

use lib '/home/wicker/WICKERsoft/';
use Biomodule;

if (@ARGV <1) {
    print "\n1.arg = global syteny table\n";
    print "\n1.arg = donor_site_list\n";
    exit;
}

open(IN, "$ARGV[0]");
@global = <IN>;
close(IN);

open (COL, ">colinear_genes");
open (NON, ">non_colinear_genes");

for($i=0;$i<(@global);$i++) {

    ($osch1,$osgen1,$sbch1,$sbgen1) = ($global[$i] =~
/^\\S+\\s+\\S+\\s+0s(\\d+)g(\\d+)\\s+\\S+\\s+Sb(\\d+)g(\\d+)\\s/);

    $osgen1 =~ s/^[0]+//g;
    $sbgen1 =~ s/^[0]+//g;

    $count_os = 0;
    $count_sb = 0;

    # check low numbers of neighbour genes
    for($j=($i-5);$j<$i;$j++) {

        ($osch2,$osgen2,$sbch2,$sbgen2) = ($global[$j] =~
/^\\S+\\s+\\S+\\s+0s(\\d+)g(\\d+)\\s+\\S+\\s+Sb(\\d+)g(\\d+)\\s/);

        $osgen2 =~ s/^[0]+//g;
        $sbgen2 =~ s/^[0]+//g;

        $diff_os = abs($osgen1-$osgen2);
        $diff_sb = abs($sbgen1-$sbgen2);

        if (($diff_os < 400) || ($osch2 ne $osch1)) {
            $count_os++;
        }

        if (($diff_sb < 400) || ($sbch2 ne $sbch1)) {
            $count_sb++;
        }

        #          print "$osch2\\t$osgen2\\t$sbch2\\t$sbgen2\\t$count_os\\t$count_sb\\t$osgen1-
        $osgen2 = $diff_os\\n";

    }

    print "$global[$i]";

    # check higher numbers of neighbour genes
    for($j=($i+1);$j<$i+5;$j++) {

        ($osch2,$osgen2,$sbch2,$sbgen2) = ($global[$j] =~
/^\\S+\\s+\\S+\\s+0s(\\d+)g(\\d+)\\s+\\S+\\s+Sb(\\d+)g(\\d+)\\s/);
```

```

$osgen2 =~ s/^[0]+//g;
$sbgen2 =~ s/^[0]+//g;

$diff_os = abs($osgen1-$osgen2);
$diff_sb = abs($sbgen1-$sbgen2);

if (($diff_os < 400) || ($osch2 ne $osch1)) {
    $count_os++;
}

if (($diff_sb < 400) || ($sbch2 ne $sbch1)) {
    $count_sb++;
}

#          print "$osch2\t$osgen2\t$sbch2\t$sbgen2\t$count_os\t$count_sb\t$osgen1-
$osgen2 = $diff_os\n";

    }

    if (($count_sb < 4) || ($count_os < 4)) {
        print NON "$global[$i]";

        print "non colinear \t$count_os\t$count_sb
\t++++++++\n";
    } else {
        print COL "$global[$i]";
    }

}

# ++++++
sub check_num {

my($k);

}

```