11/02/2021

**T09 – Long Read Sequencing of RR strains**

*FINDSTR /B/C:">" MSY25\_adjik\_m4\_assembly.fasta*

Searches for string “>” in the file

*type \*.fasta > RR.fsa*

Concatenate assemblies into a single .fsa file

*makeblastdb -in RR.fsa -parse\_seqids -dbtype nucl -out RR*

Creates “RR” database

*blastn -db RR -query DUP240\_nr\_2.fasta -out DUP240\_vs\_RR.default.out*

BLAST non-redundant DUPs against RR database

*FINDSTR /B/C:">" RR.fsa*

*blastdbcmd -db RR -entry\_batch seqbatch\_reverse.txt -outfmt "%f" -out Result\_reverse.fasta*

*blastdbcmd -db RR -entry chrI\_M22 -range 177765-177878*

Extracts the sequence in range 177,756 – 177,878 on contig chrI\_M22 in RR database

MQASSENANTKLDTLSEPSA – wrong sequence (A/G mismatch at position 26, G is correct)

MQASSENASTKLDTLSEPSA (should be fixed now)

*makeblastdb -in DFPandREF.fa -parse\_seqids -dbtype nucl -out DFPandREF*

*blastn -db RR -query DFPandREF.fa -out DFPandREF\_vs\_RR.default.out*

*blastn -db RR -query DFPandREF.fa -outfmt 6 -out DFPandREF\_vs\_RR.out*

*FINDSTR /n/B/C:">" DFPandREF\_vs\_RR.default.out > meow.txt*

Generate a file with all > found in the default file (maybe can append the hit to the end)

*sed -n '/Expect/ =' DFPandREF\_vs\_RR.default.out > blastlines.txt*

Creates a file containing the line numbers of every line with the word “Expect” in it (i.e. a hit) from the .default.out BLAST file.

*./m.sh*

*blastdbcmd -db RR -entry\_batch seqbatch.txt -outfmt "%f" -out long\_and\_hitless.fasta*

*blastn -db DFPandREF -query long\_and\_hitless.fasta -out long\_and\_hitless.default.out*

PRM9 reference sequence in BY has undergone a mutation (insertion?) – is different from most other hits in other strains. Compare with clustal.

>>>> PRM9 from I14 is the new reference sequence for PRM9

**Calling DUP240s algorithm (tentative):**

1. If >=96% identity, then they’re the same gene
2. If between (some number) and 95% identical AND additional criteria are met, e.g.:
   1. Same length
   2. Large region of identity (e.g. >50%)
   3. Synteny: if flanked by the same genes

Then we call them the same genes

**From scratch:**

**INITIAL SEARCH (R0)**

YAR023C (DFP1) is from RM, PRM9 is from BY (897 bp long); the rest are from BY (total of 9, with MST28/27 lumped together and MST28 taken as the reference sequence)

*blastn -db RR -query REFonly.fa -word\_size 7 -out REFonly\_vs\_RR.default.out*

*blastn -db RR -query REFonly.fa -word\_size 7 -outfmt 6 -out REFonly\_vs\_RR.out*

*[run REF portion of the R code – extract hits to other DUPs +/- 100 bp flanks]*

*blastdbcmd -db RR -entry\_batch seqbatchREF.txt -outfmt "%f" -out hitsREF.fasta*

*[cross-reference just to assign names to hits]*

*blastn -db DFPandREF -query hitsREF.fasta -out hitsREF.default.out*

Identified:

* chrI\_PW5:174089-174938 = DFP24(100%, 705 bp), intact
* chrXII\_PW5:915442-916297 = DFP14(99%), intact
* chrI\_PW5:166644-167534 = some chimeric homology to DFP1=YAR023C (89%), disrupted!
* chrI\_YJM454:170270-171161 = some chimeric homology to DFP1, disrupted
* chrI\_YJM145:163896-164787 = same as above, disrupted
* chrI\_M22:170268-171159 = same as above, disrupted
  + Despite lack of start homology, there is **>50% homology at the end (last ~360 bp)** -> include this in the table as a 1\* for each of the 4 strains!
  + Synteny verified by blasting +/- 10 kb flanks of this segment,
* chrXII\_PW5:918700-919595 = DFP13(99%), intact
* chrI\_YJM454:174395-175290 = DFP13(100%, 696 bp), intact
* chrI\_YJM145:173944-174839 = DFP13(100%), intact
* chrI\_M22:174393-175288 = DFP13(100%), intact
* chrI\_PW5:170794-171689 = DFP13(99%), intact
* chrI\_CLIB413:183570-184474 = DFP20(100%, 705 bp), intact
* chrI\_YPS163:176389-177293 = DFP22(100%, 705 bp), intact
* chrI\_YJM978:191709-192622 = PRM8 (96%, 714 bp – same as PRM8), intact
* ~~chrI\_I14:175911-176824 = PRM8(93%, 714 bp, same as PRM8), intact, ?~~
  + Duplicate of chrI\_175884-176824 hit!
* chrI\_YJM454:177667-178580 = DFP14(100%, 714 bp), intact
* chrI\_YJM145:177216-178129 = DFP14(100%), intact
* chrI\_M22:177665-178578 = DFP14(100%), intact
* chrI\_I14:175884-176824 = PRM8(93%, 714 bp, same as PRM8), intact, ?
  + Fails 96% criterion, but has a **large (>50%) region of identity** **& same length -> count as PRM8**
  + Synteny verified (but unclear which chromosome it came from originally)
* chrI\_YJM978:186078-187028 = DFP17(99%), intact
* chrI\_YPS163:174083-175039 = DFP25(100%, 711 bp), intact
* chrI\_CLIB219:173267-174223 = DFP17(100%, 711 bp), intact
* chrVII\_YJM978:415638-416603 = DFP16(100%, 714 bp), intact

Addendum (*blastn -word\_size = 7*):

* chrI\_YJM454:180547-181462 = DFP11 (100%, 723 bp), intact
* chrI\_YJM145:180096-181011 = DFP11 (100%, 723 bp), intact
* chrI\_M22:180545-181460 = DFP11 (100%, 723 bp), intact
* chrI\_PW5:178365-179280 = DFP11 (99%, 723 bp)

Counts: DFP13 (**5**), DFP14 (**4**), DFP16 (**1**), DFP17 (**2**), DFP20 (**1**), DFP22(**1**), DFP24(**1**), DFP25(1), DFP1 homology (**4**), PRM8

Recovered with word\_size = 7: DFP11 (**4**)

TO-DO:

~~3) Verify Benchling vs. DFPandREF\_R1.fa added DFP sequences~~

~~4) Double-check lengths (to be same as in DFPandREF.fa in the other folder)~~

~~5) Once the next iteration is started, add the < 96% hits manually to the data.table~~

~~(these are: fishy YAR023C-like pseudogenes and PRM8’s from above)~~

~~6) Check synteny for PRM8 hits~~

**ITERATION 1 (R1)**

*blastn -db RR -query DFPandREF\_R1.fa -word\_size 7 -out DFPandREF\_R1\_vs\_RR.default.out*

*blastn -db RR -query DFPandREF\_R1.fa -word\_size 7 -outfmt 6 -out DFPandREF\_R1\_vs\_RR.out*

*[R code – R1 section]*

*blastdbcmd -db RR -entry\_batch seqbatchDFPandREF\_R1.txt -outfmt "%f" -out hitsDFPandREF\_R1.fasta*

*[cross-reference]*

*blastn -db DFPandREF -query hitsDFPandREF\_R1.fasta -out hitsDFPandREF\_R1.default.out*

Identified:

* chrI\_CLIB219:176701-177611 = DFP18 (100%, 711 bp)
* chrI\_PW5:176657-177562 = DFP12 (99%, 708 bp)
* ~~chrXII\_PW5:914325-915200 = DFP12 (99%, 708 bp)~~
  + Duplicate of below
* chrI\_YJM454:178822-179727 = DFP12 (100%, 708 bp)
* chrI\_YJM145:178371-179276 = DFP12 (100%, 708 bp)
* chrI\_M22:178820-179725 = DFP12 (100%, 708 bp)
* chrVII\_CLIB219:407038-407945 = DFP21 (100%, 723 bp)
* chrXII\_PW5:914295-915200 = DFP12 (99%, 708 bp)
* chrI\_I14:170443-171353 = DFP18 (98%, 711 bp)
* chrVII\_Y10:404177-405087 = DFP23 (99%, 726 bp)
* chrVII\_I14:402375-403285 = DFP23 (100%, 726 bp)
* chrVII\_YJM454:408919-409829 = DFP23 (100%, 726 bp)
* chrVII\_YPS163:397187-398097 = DFP23 (100%, 726 bp)
* chrI\_I14:167107-168017 = DFP16 (96%, 716 bp)
* chrI\_YJM978:189504-190414 = DFP15 (100%, 711 bp)

Missing after R1: DFP~~12~~, ~~15~~, ~~18~~, **19**, ~~21~~, ~~23~~

(just CLI3)

Look at the tree (nuc.) – YJM1 and YJM2 are kind of in separate branches, might have to include separately (argument: they’re in a DUP arrays, of similar length, same predicted topology)

Interesting observation: chrVII in YJM978, 2 copies of CLI3/DFP19, ~ same location. Need to look at sequences/synteny, what happened? 2 independent segmental duplications seems unlikely

TO-DO:

1. ~~Change expect threshold to see if we can capture DFP11, 12, and 19~~
2. Double-check PRM8 hits
3. ~~Iteration R2~~

**ITERATION 2 (R2)**

*blastn -db RR -query DFPandREF\_R2.fa -word\_size 7 -out DFPandREF\_R2\_vs\_RR.default.out*

*blastn -db RR -query DFPandREF\_R2.fa -word\_size 7 -outfmt 6 -out DFPandREF\_R2\_vs\_RR.out*

*[R code – R2 section]*

*blastdbcmd -db RR -entry\_batch seqbatchDFPandREF\_R2.txt -outfmt "%f" -out hitsDFPandREF\_R2.fasta*

*[cross-reference]*

*blastn -db DFPandREF -query hitsDFPandREF\_R2.fasta -out hitsDFPandREF\_R2.default.out*

Identified (at >500 bp threshold):

* chrI\_CLIB219:178508-180016 = DFP26 (**new DUP**, PRM8+PRM9, 690 bp), intact
* chrI\_YPS1009:186250-187812 = DFP26 but disrupted (too short)
* chrI\_CLIB413:182099-183661 = DFP26 (new DUP, PRM8+PRM9, 690 bp), intact

CLIB413 and YPS1009 alleles have an N-terminal extension (102 bp for CLIB413)

CLIB413 and CLIB219 are 93.6 % identical (Clustal)

TO-DO:

1. check synteny around old and these new PRM8/9 hits
2. For each hit, take the largest ORF it belongs to

*blastdbcmd -db RR -entry\_batch seqbatch\_na\_DFPandREF\_R2.txt -outfmt "%f" -out hits\_na\_DFPandREF\_R2.fasta*

*[cross-reference]*

*blastn -db DFPandREF -query new\_ORFseqs\_R3.fasta -out new\_ORFseqs\_R2.default.out*

*blastn -db DFPandREF -query new\_ORFseqs\_R3.fasta -outfmt 6 -out new\_ORFseqs\_R2.out*

At this point, database **DFPandREF** is **obsolete**! (no DFP26)

*makeblastdb -in DFPandREF\_R3.fa -parse\_seqids -dbtype nucl -out DFPandREF\_R3*

TO-DO:

1. ~~Make a new blastdb with weird PRM8 included~~
   1. ~~Problems: what if hit ~500 bp? Could be a novel but a chimeric DUP, i.e. just pident won’t suffice. We’ll see, I guess~~
2. In the DUP table, we want to say if there’s a “gene signature” present: screen for hits >650 and include them in our table even if there’s a “frameshift” (as \*)
3. Verify all things in c.dt are full-length ORFs

**Ultimate output:** the DUP table, with potentially new DUPs (e.g. DFP19, others?)

**Additionally:** an excel sheet

**ITERATION 3 (R3):**

Added DFP26 (new DUP), recovered DFP19

*blastn -db RR -query DFPandREF\_R3.fa -word\_size 7 -out DFPandREF\_R3\_vs\_RR.default.out*

*blastn -db RR -query DFPandREF\_R3.fa -word\_size 7 -outfmt 6 -out DFPandREF\_R3\_vs\_RR.out*

*[run R code – R3 section]*

*[ORF extraction]*

*blastdbcmd -db RR -entry\_batch seqbatch\_na\_DFPandREF\_R3.txt -outfmt "%f" -out hits\_na\_DFPandREF\_R3.fasta*

*[cross-reference]*

*blastn -db DFPandREF\_R3 -query new\_ORFseqs\_R3.fasta -out new\_ORFseqs\_R3.default.out*

*blastn -db DFPandREF\_R3 -query new\_ORFseqs\_R3.fasta -outfmt 6 -out new\_ORFseqs\_R3.out*

Verification:

*blastdbcmd -db RR -entry\_batch seqbatch\_newORFseqs\_R3.txt -outfmt "%f" -out newORFseqs\_R3\_verification.fasta*

Detected YAR029W in YPS163 (576 bp largest ORF) – need to add it to c.dt

* ORFcoords: chrI\_YPS163 175434-176009
* N-terminal extension (because BLAST hit cords are 175785-176009)

**To identify starred (\*) table entries:** pull c.dt seqs into R, find ORFs/translate, look for broken-up seqs

Update on: DUP table, which strains are “reference” strains for each DUP, info in starred DUPs (see above)

*blastdbcmd -db RR -entry\_batch seqbatch\_c-dt400.txt -outfmt "%f" -out c.dt400.fasta*

**Broken-up ORF from c.dt analysis:**

*blastdbcmd -db RR -entry\_batch seqbatch\_c-dt\_disrupted400.txt -outfmt "%f" -out c.dt\_disrupted400.fasta*

*[import into Benchling]*

List (length signifies max. predicted ORF length in the +/- 400 bp flanked region):

* chrI\_273614:168285-168981: REF\_PRM9, 684 bp - **OK**
* chrI\_BY:180051-180772: DFP1, 540 bp – **star (\*)**
* chrI\_CLIB219:175684-176349: DFP19, 666 bp – **full length DUP**
  + Double-check
* chrI\_CLIB219:179008-179694: DFP26, 687 bp – **full length DUP**
* chrI\_CLIB219:180069-180773: MST28, 309 bp – **star (\*)**
* chrI\_I14:169430-170095: DFP19, 666 bp – **full length DUP**
* chrI\_M22:170368-171059: DFP1, 444 bp – **star (\*)**
* chrI\_M22:174493-175188: DFP13, 696 bp – **full length DUP**
  + Double-check
* chrI\_PW5:166744-167434: DFP1, 444 bp – **star (\*)**
* chrI\_PW5:170894-171589: DFP13, 696 bp – **full length DUP**
* chrI\_RM:163819-164524: REF\_YAR028W, 447 bp – **star (\*)**
* chrI\_RM:165767-166460: REF\_PRM9, 651 bp – **star (\*)**
  + Premature stop (10 AA) at the end (w.r.t. 684 bp PRM9 alleles), as seen from Clustal
* chrI\_Y10:191099-191795: REF\_PRM9, 684 bp - **OK**
* chrI\_YJM145:163996-164687: DFP1, 444 bp – **star (\*)**
* chrI\_YJM145:174044-174739: DFP13, 696 bp – **full length DUP**
* chrI\_YJM454:170370-171061: DFP1, 444 bp – **star (\*)**
* chrI\_YJM454:174495-175190: DFP13, 696 bp – **full length DUP**
* chrI\_YJM978:170363-171086: DFP1, 540 bp – **star (\*)**
* chrI\_YJM978:188494-189159: DFP19, 666 bp – **full length DUP**
* chrI\_YJM981:174307-175011: REF\_YAR028W, 423 bp – **star (\*)**
* chrI\_YJM981:176252-176945: REF\_PRM9, 684 bp - **OK**
* chrI\_YPS1009:186750-187439: DFP26, 546 bp – **star (\*)**
* chrI\_YPS163:179552-180247: REF\_PRM9, 684 bp - **OK**
* chrVII\_YJM978:418064-418729: DFP19, 666 bp – **full length DUP**
* chrVIII\_PW5:9201-9908: DFP4, 303 bp – **star (\*)**
* chrVIII\_Y10:8073-8781: DFP4, 447 bp – **star (\*)**
* chrXII\_PW5:918800-919495: DFP13, 696 bp – **full length DUP**

TO-DO:

1. Determine whether chrI arrays are flanked by CDC15 and YAT1 as follows:
   1. BLAST CDC15 and YAT1 to determine their locations in 16 strains
   2. Use systemPipeR::predORF() to predict all orfs >650 in that region. Should only be DUP240s.
   3. If not, we may have missed some DUPs (unlikely), or we have some intruders in the DUP array that are >650 bp
2. Same but for chrVII arrays, flanked by ERV14 and TYW3

Flank verification:

*blastn -db RR -query chrI\_CDC15-YAT1.fa -outfmt 6 -out chrI\_CDC15-YAT1.out*

*blastn -db RR -query chrVII\_ERV14-TYW3.fa -outfmt 6 -out chrVII\_ERV14-TYW3.out*

*blastdbcmd -db RR -entry\_batch seqbatch\_ERV14-TYW3\_flanks.txt -outfmt "%f" -out RR\_chrVII\_arrays.fasta*

Cross-check complete! No new DUP240s found in either chrI or chrVII arrays.

(see is.na() entries in

**T09 – Long Read Sequencing Analysis – updating c.dt to include ORF coords**

*blastdbcmd -db RR -entry\_batch seqbatch\_c-dt400.txt -outfmt "%f" -out c.dt400.fasta*

*blastdbcmd -db RR -entry\_batch seqbatch\_c-dt400orfs.txt -outfmt "%f" -out c.dt400orfs.fasta*

*blastdbcmd -db RR -entry\_batch seqbatch\_029W\_600bp.txt -outfmt “%f” -out c.dt029W.fasta*

*blastdbcmd -db RR -entry\_batch seqbatch\_029Worfs.txt -outfmt “%f” -out c.dt029Worfs.fasta*