“Chopping” build sequences using Gene Assembly by Sub-Pool PCR (GASP)

0. To run the script:

a) Place input files in the input-seqs directory
b) Edit the configuration file plate-based-assembly-from-ols-pool-config. See #2 below for an important note on plate numbering.
There is one configuration entry per file. Make sure that you set:
initialPlateNum(current entry) = initialPlateNum(previous entry) + (number of plates used up by sequences in previous entry)
c) cd to script directory in terminal and type “python gasp.py”
d) pay attention to outputs in the form of:
oligo files (oligo sequences generated corresponding to each input file),
primer files (primer list corresponding to each input file),
report files (build seq by build seq delineation of primer sets to use, for each input file),
and to **agilent-upload.txt** which contains all oligo sequences generated by the script run in a form which can be copied and pasted into excel
e) don’t forget to save a copy of the configuration file used for this run

1. Sri’s original configuration settings were

```
"avgoverlapsize": 20,
"deltaGThresholdForOverlaps": -3,
"selfDimersThreshold": 3,
"insertionSizeToKillRESite": 2,
"lengthleeway": 10,
"overlaptemps": [
        55,
        65
],
"positionleeway": 10,
"seqsToAvoidInOverlapRegions": []
“skip”: []
```

In order to tolerate a wider range of sequences, you can modify these to

```
"avgoverlapsize": 20,
"deltaGThresholdForOverlaps": -3,
"selfDimersThreshold": 5,
"insertionSizeToKillRESite": 2,
"lengthleeway": 15,
"overlaptemps": [
        55,
        65
],
"positionleeway": 10,
"seqsToAvoidInOverlapRegions":[],
“skip”:[]
```

(or something in between)

Making such changes to the configuration setting should not be catastrophic. Increasing the length leeway increases the amount of time, but increasing the threshold for primer dimers would decrease it. Likewise, it should be possible to set positionleeway = 15.

2. Importantly, do not EVER set

```
"initialPlateNum": 1
```

for any of the input files in the configuration file. Doing this will create an error where the building block primers (i.e., “construction primers”) are chosen to be the same as the position specific primers. Plate #1 is reserved for a special purpose. The script should report an error if you try to do this.

3. If the script reports an error like "Error: incorrect number of restriction sites per oligo"

This probably means that something like the following has happened.

Good (example of desired oligo output):
CCCTTTAATCAGATGCGTCGTGGGTAGAAGGCAGTTACACGCAGTGTTGCGTCTTTTACACAGCATGAGCGGGTCTCACATTGCCTTCCTGTTGACCTTCGTTGTCACCTGAGTTGAGTCCAGTCTGTCTGAGTCCTTGATG**CACTGCGGGATGAGAAAGCAAGGAGTGGTCGCCCTTATTACTACCA**

Bad (example of oligo output when this error occurs):
CCCTTTAATCAGATGCGTCGCAGTG**TGTATTCACGAAGCC**GCAGTGTTCACTTGAGGTCGGTTTGAAGGCGGGTCTCACATTGCCTTTCCTGTCATCACATGTCTCCAGTGCTTGAAACACCTTCCAAGGAACCTCTTGCTT**CACTGCGCGAGAAACAACTTAACGGTGGTCGCCCTTATTACTACCA**

In this case, the underlined region is what is added, purple is the BtsI restriction site to be used in the restriction processing step before final assembly, blue is the rev_comp of that site, orange is a construction primer (in this case), and bold-black are the concatenated plate specific and position specific primers.

Note that in the "bad" case there is an extra purple forward restriction site inside the plate specific and position specific primer region. This occurred by generation of a new BtsI site when the plate specific and position specific primers, which individually were chosen by the primer design algorithm to lack the BtsI site, were concatenated. We see from this example that junctions can sometimes, but rarely, add a restriction site. This is usually not that bad, especially if the primers are removed later.

In such cases, there is old code (not yet incorporated into current GASP script) to add a base between junctions: there is an option to add a buffer base between the junction iff a restriction site is created.

4. If the script reports an error like "Error: We were not able to find an overlap structure consistent with the parameters in the config file."

This reflects failure of the program on one of the sequences you specified. Either remove that sequence from the input file, or try making the overlap search parameters more lenient, for instance by increasing selfDimersThreshold or lengthleeway in the configuration file.

5. If the script reports an error like "Error: No restriction enzyme found for building block..."

This reflects the fact that you need to change the sequence, or add restriction enzymes to the REvector entry corresponding to the file containing that sequence in the configuration file.

Sometimes this error will occur even if you have eliminated the restriction site from the sequence to be synthesized - this is because addition of the construction primers on either side of the sequence to be synthesized can cause generation of restriction sites (at the edges of the sequence) which exist neither in the desired sequence nor in the construction primers themselves. Try changing the bases at the edges of the sequence to be synthesized in order to avoid this problem.

6. Explanation of terms in configuration file

```
{"configDictList":[ // each entry in this list corresponds to a single input file with many seqs
        {
        "initialPlateNum": 3, // never set this = to 1
        "buildSequencesFile": "input-seqs/All_MC_Scaffolding_Seq_PostProc.txt", // input file name
        "RESpacing": [ // offset of cut site from recognition site for type II enzyme
                2,
                5,
                4
        ],
        "REVector": [ // sites it can choose for processing step if SearchForRE = True
                "BtsI",
                "BsmBI",
                "BspQI"
        ],
        "SearchForRE": "True", // set this to “False” if you want to use a fixed RE
        "REToUse": "BtsI", // fixed RE to use, if applicable
        "forwardPrimersLibraryFile": "primer-library/forward_finalprimers.fasta", // from Sri
        "reversePrimersLibraryFile": "primer-library/reverse_finalprimers.fasta", // from Sri
        "fixedPrimerSetThroughoutFile": "False", // true if all seqs in file get same primer set
        "fixedIndex": 0, // plate position index of primer set when using fixed primer set
        "fixedPlateNumber": 1, // plate # of primer set when using fixed primer set
        "avgoverlapsize": 20, // keep this fixed
        "deltaGThresholdForOverlaps": -3, // keep this fixed
        "selfDimersThreshold": 5, // default = 3, deltaG threshold for “primer dimer” effect
        "insertionSizeToKillRESite": 2, // keep this fixed
        "lengthleeway": 15, // default = 10, length of oligo can be + or - this #
        "overlaptemps": [ // keep these fixed, range of overlap melting temps for PCR
                55,
                65
        ],
        "positionleeway": 10, // oligo center position can be + or - this
        "seqsToAvoidInOverlapRegions":["CATCCGTCGAAGACGGATG"] // set to [] if can
        “skip”: ["ATbias_Seq1"] // names of sequences not to process in the file
        }
]
}
```

7. Neglecting to specify seqsToAvoidInOverlapRegions [specific to Shih lab origami staple synthesis work]

For staple constructs, we can set seqsToAvoidInOverlapRegions to the hairpin sequence if we want to strictly exclude instances of that sequence from intersecting *at all* with the overlap regions. This greatly slows down the search process and sometimes causes it to fail with otherwise good parameters.

So what happens if we leave this parameter blank, while keeping the 2ary structure threshold, dimer threshold and length leeway to near-normal values? Manual inspection suggests that we get a typical overlap structure like this, where bold = flanking primers, purple = hairpin sequence, red = overlap regions.

1st oligo:
GGGTGGGTAAATGGTAATGCTAAGCGCCCTTCTAATACCCGCAGTGCCATCGGCATCAGGTTAAAACAACATAGCGTAAAATTCATCCGTCGAAGACGGATGAAGAAAGGATGGGGTCCTGTTTGTATACAGGATGATGGAC**CACTGCGGCGAAGATAAAACCGCTACGCTATATCCGGGGAATCGAT**

2nd oligo:
GGGTGGGTAAATGGTAATGCTAAGCGCCCTTCTAATACCCGCAGTGCCTGTTTGTATACAGGATGATGGACGCTGTTGTAAACATCCGTCGAAGACGGATGTTGAGAAAGCACTAAATTGCCCCACATCGGTCAATTATG**CACTGCGGCGAAGATAAAACCGCTACGCTATATCCGGGGAATCGAT**

3rd oligo:
GGGTGGGTAAATGGTAATGCTAAGCGCCCTTCTAATACCCGCAGTGTTGCCCCACATCGGTCAATTATGAATTTTTGTATACATCCGTCGAAGACGGATGTAGGGAAAGGGAGCCCCAGCAAGGATTCGCCGCAGAGTCT**CACTGCGGCGAAGATAAAACCGCTACGCTATATCCGGGGAATCGAT**

and so forth...

Note that in most instances the hairpin does not, in fact, intersect with the overlap regions to a significant degree, due to the existence of a 2ary structure constraint on the overlap regions which is independent of the explicit screening option, leading the overlaps to have less than 3 kcal / mol of 2ary structure free energy. In cases where the hairpin does intersect the overlap region, the *whole* hairpin stem would *likely* not be placed in an overlap region due to the 2ary structure constraint.