

BioBricks++: Simplifying Assembly of Standard DNA Components

Austin Che

June 9, 2004

Abstract

Construction of complex biological systems can require assembling many modules together. However, existing assembly schemes are lacking in generality, ease of use, or power to perform some desirable operations. Currently, biological modules are most easily specified and manipulated as DNA sequences. A general system, called BioBricks++, for assembling and manipulating DNA modules is proposed. BioBricks++ was inspired by the BioBricks assembly scheme but provides for more possible module operations.

BioBricks++ uses commercially available restriction enzymes and standard biological techniques for assembling modules. The key to the method is in the specification of the standard DNA module. Modules are packaged with a standard prefix and suffix DNA sequence containing several restriction enzyme sites, which are used for different module operations.

The following operations can be performed on all BioBricks++ modules. The most fundamental operation is the arbitrary assembly of any two modules. In addition, the assembly of the two modules can be made seamless, with no extra intervening sequence inserted between the modules. Modules can also be easily reversed with no extra bases added during the operation. Another useful capability is being able to remove bases from either end of a module, allowing for operations such as protein fusions or addition of tags.

1 Overview

The following describes BioBricks++, a proposed alternative to the BioBricks assembly scheme using offset and nicking restriction enzymes. We first provide the motivations for this new scheme.

1.1 Advantages

- The arbitrary prefix/suffix assembly of parts is possible, similar to the capabilities of BioBricks.
- Parts can be easily and seamlessly reversed which cannot be done in BioBricks.

- A small number of bases can be removed from either end of a part, allowing for operations such as protein fusions.
- Parts are all stored in a single kind of plasmid unlike the 3 different plasmids needed for the 3-antibiotic selection.
- Long overhangs are used on both ends during assembly, allowing for ligation independent cloning. Also, chromosomal or other random DNA contamination will no longer be an issue with the long overhangs.
- Assembly can be made seamless by using blunt end ligation between parts, unlike the mixed site formed under BioBricks assembly. Sticky end ligation is also possible for possibly higher efficiency ligations, with a smaller mixed site (4bp rather than 8bp in the BioBricks system).
- We don't need to use different ends for different parts as done in the BioBricks scheme to allow for the separation of the RBS and CDS.
- This new system was designed with forward compatibility in mind. It is possible to cleanly move parts into new plasmids.
- Only three recognition sequences are reserved for parts (1 6-bp and 2 7-bp). All are asymmetric, so we have to double that number for double stranded DNA. This is still better than the 4 symmetric 6-bp sequences in BioBricks, placing fewer restrictions on parts. In random DNA, a restricted sequence appears with a greater than 50% probability in about 710bp for BioBricks and 945bp for BioBricks++.
- Assembly without purification should be possible. Instead of antibiotic selection to select for the destination plasmid, the parent plasmids are selected against by cutting it into many pieces using offset cutters. The offset cutters make each piece have a different overhang making it unlikely to come back together. Also, the overhangs have been designed to be compatible with a biotin purification strategy.
- Assembly of short parts may be easier as the length of the long overhangs impose a lower limit on the length of a cut out fragment.
- No oligos specific to any parts need to be synthesized to perform any of the operations. This removes the necessity to *think* about any of the operations. All operations are done identically each time. The amount of time, cost, and materials required is independent of the size of the module, i.e. $O(1)$, assuming that cloning takes a constant amount of time. In contrast, direct DNA synthesis or PCR assembly is at least $O(n)$.

1.2 Disadvantages

- Different plasmids are needed for different operations. Also, all N.BstNBI sites (asymmetric 5-bp sequence) need to be eliminated from plasmids and there could be many of these sites. However, plasmid construction only needs to be done once.

- If the source plasmids are not completely cut, then they may be able to transform cells. It is unknown whether the proposed anti-selection against the parent plasmids is sufficient.
- The long overhangs used may make it more difficult to separate the fragments on a gel.
- Only existing components can be manipulated. It is not possible to generate something that isn't directly related to modules that already exist.

2 Operational Details

All the figures and sequences for the operations are given at the end.

Three recognition sequences are reserved for parts and thus cannot be used in any parts. The enzymes and their recognition sequences are given in Figure 1. AarI has a 7-bp sequence, BbsI has a 6-bp sequence and BbvCI (and its derivative nicking enzymes) have a 7-bp sequence.

Additional enzymes and sequences are reserved for the plasmid backbone and these sequences must be eliminated from all plasmids. These are given in Figure 2. The parts can have the sequences reserved for the plasmid with no problems.

All parts are required to be of the form given in Figure 3. The prefix and suffix regions for each part are fixed sequences common to all parts, and could be made arbitrarily long to make the ligation reaction use longer overhangs. The part in the base plasmid is shown in Figure 4.

2.1 Assembly

Assume that we have two parts, $part_1$ and $part_2$, that we wish to assemble together into the new part, $part_1part_2$. We cut the prefix part, $part_1$ with BbsI and N.BbvCIA and the suffix part, $part_2$ with AarI and N.BbvCIB. We cut the assembly plasmid with N.BstNBI and BsaI (Figure 5). The cut parts and the assembly plasmid are mixed together and will form the desired part construct, $part_1part_2$, in the exact same form as the original parts.

To avoid the 4-bp site between the parts, we first cut the parts with AarI or BbsI. Then we use a nuclease, such as mung bean nuclease or nuclease S1, to remove the overhang creating a blunt end. Next, the appropriate nicking enzyme is used to create the long overhang. The ligation would then involve long overhangs on the two ends and a blunt end ligation between the parts.

2.2 Entry Cloning

To make parts requires initially putting parts into the standard form. PCR primers can be designed to contain the sequence up to the closest BbvCI upstream and up to the BbsI sequence downstream. After PCR, cut the part with BbsI and BbvCI. Then cut the entry plasmid with SapI and Eco3II, ligate, and clone to get a part of the correct form (Figure 6).

Another method for cloning that may be simpler is to use TA cloning. PCR primers do not need to have any additional bases as it so happens that the base immediately 3' to the part is A on either side. During a PCR using Taq polymerase, an extra A will be added to the 3' end of the PCR product. The TA Entry plasmid is cut with enzymes that leaves a single T that can be ligated directly with the PCR product. Both non-directional (Figure 7) and directional (Figure 8) TA cloning is possible.

2.3 Reversing

To seamlessly reverse the orientation of a part, cut the part with BbsI and BbvCI. The reversing plasmid is cut with SapI and Eco31I and ligated with the part to obtain the part in the correct BioBricks++ form but flipped in the other direction (Figure 9).

2.4 Base Removal

The removal of bases from the ends of parts can be useful especially for operations such as protein fusions or to add tags to proteins. For example, to add a *ssrA* degradation tag to a protein, it is necessary to remove the last stop codon (presumably the last 3 bases) and attach a *ssrA* tag with a stop codon after it. Due to the properties of the offset cutters, up to 6 bases at the end and up to 10 bases at the front of a part can be removed in one operation. More bases can be removed by repeated removal operations. An example of the removal method is shown in Figure 10.

2.5 Fixed Prepend and Append Plasmids

Assembly of short parts may be difficult to deal with due to short fragments being lost. Plasmids can be constructed that have one part pre-designed into it for prepending or appending. Thus, assembly only requires a two-way ligation rather than a three-way ligation. This can be useful for frequently used parts that are always used in a similar fashion. For example, a *ssrA* tag or a *his* tag are short sequences that may be attached often to coding sequences. By designing a plasmid with the tag built in, assembly can be made easier. Figure 11 and Figure 12 show the details of the operations.

Unfortunately, it is necessary to introduce mutations from the standard sequence to knock out extra N.BstNBI sites. For prepending, a single mutation is needed, but for appending, two mutations are needed. These sequences do not impact any of the standard part operations, so reversal and assembly would still work. For the sake of consistency, it is possible to cut out the final part with BbsI/BbvCI and put it into pEntry, putting it back into the standard form.

2.6 Part Transfer

A simple way to transfer a part to a completely different plasmid is to cut the part with BbsI and BbvCI and ligate into another plasmid but this introduces some extra bases on each side of the part.

If we want to transfer a part cleanly and seamlessly to put on completely new ends, we can do the following procedure.

1. Cut the part with BbsI and blunt the end with a single-strand nuclease.
2. Cut the part with N.BbvCIA.
3. We need an intermediate plasmid that can be made to have the matching long prefix sequence and the desired final suffix sequence.
4. After ligating the part with the plasmid, we will have a part with the correct suffix. We then cut with AarI and blunt the end. It is then necessary to cut with another enzyme in the new plasmid to transfer the part to the final plasmid.

This doesn't work to put into a plasmid that uses both of AarI and BbsI.

2.7 BioBricks Conversion

To convert existing parts in the BioBricks form to this BioBricks++ form, we propose some transition plans that do not involve redesigning specific primers for parts. We assume that the parts to be converted do not contain any of the new restricted restriction sites (AarI, BbsI, BbvCI). If they do, these sites would have to be mutated separately.

Two methods for moving a BioBricks part into the new form are presented below. Both add a certain number of bases on either side to the part. For a seamless upgrade, we can use an appropriate pRemove plasmid to remove some number of bases from either end.

2.7.1 Method 1: Simple

We cut both the upgrade plasmid and the original BioBricks part with XbaI and SpeI (Figure 13). We ligate the plasmid with the part in the presence of XbaI and SpeI. As XbaI and SpeI have identical overhangs, the part can ligate into the plasmid in either orientation. To force the correct orientation, by adding XbaI and SpeI during the ligation, we cut any XbaI or SpeI site that reforms. If the part ligates correctly, two mixed sites will be formed that cannot be cut. We end up with a part that has 7 extra bases in the front and 6 extra bases in the back, but is in the correct BioBricks++ form.

This method for upgrading parts is relatively simple, but may be problematic due to the identical XbaI and SpeI overhangs. In particular, the upgrade plasmid ends can ligate to themselves forming a site that will not be cut by XbaI or SpeI. The presence of the insert probably needs to be confirmed by verifying colonies.

A variation would be to cut an appropriate plasmid and the BioBricks part with NotI and SpeI. This would solve the problem with the overhangs and would force directional cloning. However the final part would have more extra bases on the front than could be removed in one iteration with pRemove.

2.7.2 Method 2: PCR

Another BioBricks upgrade method is given in Figure 14. Again, the original part is cut with XbaI and SpeI. Instead of selecting for the insertion in the correct orientation by cutting the ligations with XbaI and SpeI, a PCR is done to amplify only the correctly ligated part. The amplified ligation is then cut and cloned into the entry plasmid. The new part will contain 4 extra bases on either end of the original part.

3 Design Notes

3.1 Restriction Enzymes

Where possible, the restriction enzymes were chosen to have desirable properties.

- All enzymes except N.BstNBI and BsaI operate at 37°C. N.BstNBI works at 55°C and BsaI works at 50°C.
- Eco31I and BsaI have the same recognition sequence. As Eco31I works at 37°C, it can be used for double digesting with SapI.
- All enzymes are commercially available. AarI and Eco31I are available from Fermentas and all other enzymes are available from NEB.
- BbvCI, according to NEB, has less than 5% ligation after cutting. The reason for this is unknown and whether the nicking enzymes may have a similar problem is not known either. However, experiments with the cloning methods here have not come across any problems, perhaps due to the fact that the plasmid and inserts are not being both cut with BbvCI. It may be that BbvCI de-phosphorylates when it cuts. Another enzyme that could be tried is Bpu10I from Fermentas that has a recognition sequence that is less specific than BbvCI.
- Methylation issues. All potentially overlapping dam, dcm, EcoBI, and EcoKI sites were checked for and ensured to not occur with any enzymes sensitive to methylation to avoid potential problems with restriction digests.
- The positions of the restriction enzymes were chosen to have an easy to remember mnemonic. The ‘A’ enzymes, AarI and N.BbvCIA, are used to cut in front of the part and the ‘B’ enzymes, BbsI and N.BbvCIB, are used to cut behind the part. The offset cutters, AarI and BbsI, are used to make short ends and the nicking enzymes make long ends.
- Different nicking enzymes are used for the prefix and suffix so a single insert’s ends won’t be compatible with the destination plasmid if the offset cutter doesn’t cut.
- After choosing the enzyme SgrAI, it was discovered that it is not an ideal enzyme. Even though it has a long recognition sequence, it is able to cleave several defined secondary sites when it cuts the primary sequence (see NEB enzyme description of star activity). If

one eliminates all such secondary sites from the plasmid backbone, this should not be a problem.

- The restriction enzymes BspMI/BfuAI (NEB) and BveI (Fermentas) have the recognition sequence (ACCTGC 4/8). These enzymes are less expensive than AarI and may be better cutters. The recognition sequence is a superset of AarI so it can directly replace AarI in this scheme. However, it is a 6-bp recognition sequence vs. the 7-bp sequence for AarI. It is conceivable to use these enzymes for operations for all parts that don't need the longer sequence of AarI.
- The extra reserved restriction enzymes for plasmids can be used to engineer the various plasmids needed. The plasmids are designed to have many AarI and BbsI on them. Thus, when the parts are cut during the assembly process, the parent plasmids are cut into many pieces, all of which can have different overhangs.

3.2 Miscellaneous

- There is one caveat with respect to part design. Other than not having any of the recognition sequences for AarI, BbsI, and BbvCI, caution should be used for assembly with parts ending with the sequence CCTC. If a part that ends in CCTC is used as a prefix part, *and the seamless assembly method is not used*, then the extra AGCT inserted between the parts would insert an extra BbvCI site. If the seamless assembly method is used, then any of these three recognition sequences could be potentially formed across the boundary of the two parts. Choosing whether to assemble seamlessly or not should be able to handle most cases without extra restrictions on parts.
- There is large flexibility in the prefix and suffix regions. The currently chosen sequences are given in Figure 3 and were selected to introduce restriction sites to engineer the plasmids and to make both overhangs 25bp in length. It is not known if making the overhangs *too long*, the insert would become difficult to separate from the parent plasmid after the restriction digest. Also, if the prefix and suffix are too long, it may become difficult to use PCR to generate the long overhangs for the assembly plasmid, requiring extremely long primers. The prefix and suffix sequences were designed to not be complementary both with respect to themselves and with each other.
- The system was designed to use 5' overhangs so that the plasmids with long overhangs could be constructed by PCR with non-natural bases such as 2'-O-methyl bases instead of using the nicking enzyme N.BstNBI.
- If the parent plasmid anti-selection method by chopping does not work, then the 3-antibiotic selection mechanism can still be used.
- Potentially unwanted ATG start codons were checked for to prevent extra start sites being formed regardless of the part to be inserted.
- One additional feature that this system lacks would be to use a non-self-complementary 4-bp overhang between parts to lower the chance

of any self-ligations. Unfortunately given the other constraints of being able to reverse a part and for seamless assembly, this could not be designed into the system.

3.3 Implementation Status

The operations and status of plasmids that have been constructed and tested in the lab include:

3.3.1 Plasmids

- The base plasmid is derived from pSCANS with an inducible copy number. Many point mutations were introduced to remove restriction sites from the plasmid.
- The plasmids pAssembly, pEntry, pReverse, pBBUpgrade have been constructed based on the base plasmid.

3.3.2 Operations

- Cloning of parts into pEntry has been tested.
- Reversal of parts into pReverse works as expected. Double reversals returns the part to its original form with no extra bases.
- Assembling two parts together has been tested.
- Upgrading of BioBricks parts into BioBricks++ has been tested.

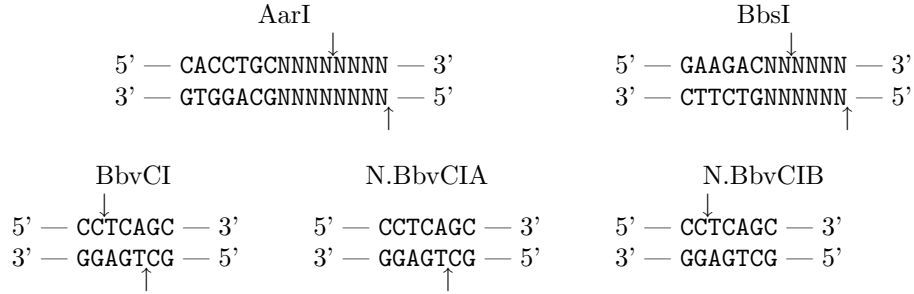


Figure 1: These restriction enzyme sites are reserved for parts. Conforming parts must not have these sites within its sequence. AarI and BbsI are offset cutters. The recognition sequence for BbvCI and the nicking enzymes, N.BbvCIA and N.BbvCIB, are identical.

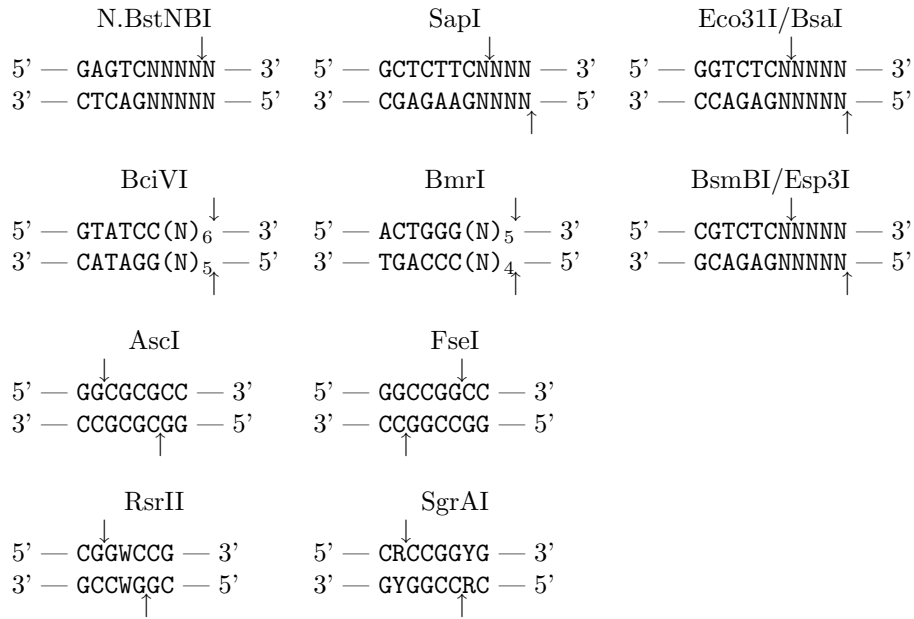


Figure 2: These restriction sites are reserved for plasmid operations and should not appear in the plasmid backbone.

Standard Part Sequence:

```

AarI  BbvCI  FseI  BbvCI
--GCTGAGG<--Prefix-->CACCTGCCCTCAGCT<--Part-->AGCTGAGTCTTCGGCCGGCCCTCAGC<--Suffix-->GCTGAGG--
--CGACTCC<--Prefix-->GTGACGGGAGTCGA<--Part-->TCGACTCAGAAGCCGGCCGGGAGTCG<--Suffix-->CGACTCC--
BbvCI                                     BbsI                                     BbvCI

```

Part cut with AarI and N.BbvCIB:

```

AGCT<--Part-->AGCTGAGTCTTCGGCCGGCCCTCAGC<--Suffix-->CGACT
<--Part-->TCGACTCAGAAGCCGGCCGGGAGTCG<--Suffix-->CGACT

```

Part cut with BbsI and N.BbvCIA:

```

TGAGG<--Prefix-->CACCTGCCCTCAGCT<--Part-->
CGA<--Part-->TCGA

```

Part cut with BbsI and BbvCI:

```

TCAGCT<--Part-->
CGA<--Part-->TCGA

```

Prefix Sequence:

```

AscI
GGCGCGCC
CCGCGCGG

```

Suffix Sequence:

```

ggtagatagatagcc
ccatctatctatcgg

```

Figure 3: The standard sequence for parts is shown cut with various enzymes. *FseI* was inserted to eliminate any potential problems with the simultaneous digest of *BbsI* and *BbvCI*. The chosen prefix and suffix sequences is given here. The length of the overhangs during assembly will be $17+|\text{prefix}|$ and $10+|\text{suffix}|$. Thus, with these sequences, the overhangs will be 25bp on both ends.

Base (Part) Plasmid:

```

RsrII      N.BstNBI      AarI      BbvCI
...Plasmid-->CGGACCGCaGTTCGAGTctagCTGAGG<--Prefix-->CACCTGCCCTCAGCT<--Part...
...Plasmid-->GCCTGGCGtCAGAAGCTCAGatCGACTCC<--Prefix-->GTGGACGGGAGTCGA<--Part...
          BbsI      BbvCI

(N.BstNBI)      FseI      BbvCI      AarI      SgrAI
...Part-->AGCTGAGTCTTCGGCCGGCCCTCAGC<--Suffix-->GCTGAGGtaGACTCCACCTGgCACCGGTG<--Plasmid...
...Part-->TCGACTCAGAAGCGCGGGGGAGTCG<--Suffix-->CGACTCCatCTGAGGTGGAGCgGTGGCCAC<--Plasmid...
          BbsI      BbvCI      N.BstNBI

```

Figure 4: Part of the sequence of the plasmid containing a part is shown. This plasmid can be used as the base to construct the other plasmids. The lowercase letters represent bases not part of a restriction enzyme recognition sequence. The extra *N.BstNBI* does not interfere with assembly. The extra *AarI* and *BbsI* sites allow for higher enzyme cutting efficiency (for *AarI*) and for ease of purification.

Assembly Plasmid:

```

N.BstNBI
--GACTCtaGCTGAGG<--Prefix--->CACCTGCCCTCAGGAGACCg<--ccdB-->cGGTCTCCTCAGC<--Suffix--->GCTGAGGtaGACTC--
--CTCAGatCGACTCC<--Prefix--->GTGGACGGGAGTCCCTCTGGc<--ccdB-->gCCAGAGGAGTCC<--Suffix--->CGACTCCatCTGAG--
BbvCI
AarI
BsaI BbvCI

```

Assembly Plasmid cut with N.BstNBI and BsaI:

```

N.BstNBI
--GACTCtaGC
--CTCAGatCGACTCC<--Prefix--->GTGGACGGGAGT
BbvCI
TCAGC<--Suffix--->GCTGAGGtaGACTC--
CCatCTGAG--
N.BstNBI

```

Figure 5: The assembly plasmid is used to join together parts using long overhangs, eliminating the need for ligase. The sequence for the part of the plasmid not shown is the same as the base plasmid.

```

Entry Plasmid:
AarI          Eco31I          FseI          BbvCI
--G<--Prefix-->CACCTGCCCTCAGGAAGAGC<--ccdB-->cGGTCTtAGCTAGTCTTCGGCCGGCCCTCAGC<--Suffix-->G--
--C<--Prefix-->GTGGACGGGAGTCCTTCTCG<--ccdB-->gCCAGAGaTCGACTCAGAAGCCGGCCGGGAGTCG<--Suffix-->C--
SapI          BbsI
AarI          AarI          BbvCI
--G<--Prefix-->CACCTGCCC          AGCTGAGTCTTCGGCCGGCCCTCAGC<--Suffix-->G--
--C<--Prefix-->GTGGACGGGAGT          CTCAGAAGCCGGCCGGGAGTCG<--Suffix-->C--
BbsI

```

Entry Plasmid cut with SapI and Eco31I:

Desired Part construct:

```

AarI  BbvCI          FseI
CACCTGCCCTCAGCT<--Part-->AGCTGAGTCTTCGGCCGGCC
GTGGACGGGAGTCGA<--Part-->TCGACTCAGAAGCCGGCCGG
BbsI

```

PCR primers:

5' -CACCTGCCCTCAGCT--Part--3' and 3' --Part--TCGACTCAGAAGCCGGCCGG-5'

Part PCR cut with BbsI and BbvCI:

```

TCAGCT<--Part-->
CGA<--Part-->TCGA

```

Figure 6: The entry plasmid is used to easily make parts from a PCR reaction. The sequence for the part of the plasmid not shown is the same as the base plasmid. To obtain the desired part construct, the 5' ends for the PCR primers are given. The 3' ends need to be designed to anneal with the desired part sequence. After a PCR to amplify the desired part, a digestion with BbsI and BbvCI allows for insertion into the entry plasmid.

Non-directional TA Entry Plasmid

```

AarI  BbvCI  NdeI      BciVI      FseI      BbvCI
--G<--Prefix-->CACCTGCCCTCAGCTCATATGGATAC<--ccdB-->GTATCCATATGAGTCTTTCGGCCGGCCCTCAGC<--Suffix-->G--
--C<--Prefix-->GTGGACGGGAGTCGAGTATACCTATG<--ccdB-->CATAGGTATACTCGACTCAGAACCGCCGGGGGAGTCG<--Suffix-->C--
      BciVI      NdeI      BbsI

```

Non-directional TA Entry Plasmid cut with BciVI

```

AarI  BbvCI
--G<--Prefix-->CACCTGCCCTCAGCT
--C<--Prefix-->GTGGACGGGAGTCG

      FseI      BbvCI
      GCTGAGTCTTCGGCCGGCCCTCAGC<--Suffix-->G--
      TCGACTCAGAACCGCCGGGGGAGTCG<--Suffix-->C--
      BbsI

```

Part PCR with Taq polymerase:

```

<--Part-->A
A<--Part-->

```

Figure 7: Another method of entry into the system is via TA cloning. PCR primers are designed with no additional bases on the 5' end. Taq polymerase is used during the PCR or afterwards to add a single non-templated A on the 3' end. The TA Entry Plasmid is cut with BciVI which leaves single base overhangs that can then be ligated directly with the PCR product. This is non-directional so the part can go in either direction. As we can easily reverse a part, this may not be a problem depending on the application. It is also easy to screen colonies for the correct insert direction via PCR.

Directional TA Entry Plasmid

```

AarI  BbvCI  NdeI          BmrI          FseI  BbvCI
--G<--Prefix-->CACCTGCCCTCAGCTCATATGGATAC<--ccdB-->ACTGGTAGGAGCTGAGTCTTCGGCCGGCCCTCAGC<--Suffix-->G--
--C<--Prefix-->GTGGACGGGAGTCGAGTATACTATG<--ccdB-->TGACCCATCCTCGACTCAGAAGCCGGCCGGGGAGTCG<--Suffix-->C--
                                     BbsI

```

Directional TA Entry Plasmid cut with BmrI, dephosphorylated, and then BciVI

```

AarI  BbvCI
--G<--Prefix-->CACCTGCCCTCAGCT
--C<--Prefix-->GTGGACGGGAGTCG*

FseI  BbvCI
GCTGAGTCTTCGGCCGGCCCTCAGC<--Suffix-->G--
TCGACTCAGAACCCGGCCGGGGAGTCG<--Suffix-->C--
                                     BbsI

```

Part PCR with Taq polymerase:

```

<--Part-->A
A<--Part-->*

```

Figure 8: For directional TA cloning, we can force phosphorylation of one end of both the insert and the plasmid to allow ligation in only one direction. One example is given here although other directions are possible. The asterisk represents the presence of a phosphate group. The plasmid is cut with BmrI and then dephosphorylated with a phosphatase. Then it is cut with BciVI to leave single T overhangs except that one end is missing a phosphate. The part is PCR'd with primers that have a phosphate on the 5' end added to only one of the primers (e.g. with T4 PNK). The ligation will then favor the part going in so that the ends with no phosphate are matched with those that have a phosphate.

Reversing Plasmid:

```

AarI   BbvCI   SapI   FseI   BbvCI
--G<--Prefix-->CACCTGCCCTCAGCTaGAGACCg<--ccdB-->GCTCTTCCCTGAGTCTTGGCCGGCCCTCAGC<--Suffix-->G--
--C<--Prefix-->GTGGACGGGAGTCGAtCTCTGGc<--ccdB-->CGAGAAGGACTCAGAAGCCGGCCGGGAGTCG<--Suffix-->C--
Eco31I

```

Reversing Plasmid cut with SapI and Eco31I:

```

AarI   FseI   BbvCI
--G<--Prefix-->CACCTGCCCTC
--C<--Prefix-->GTGGACGGGAGTCGA
TGAGTCTTGGCCGGCCCTCAGC<--Suffix-->G--
CAGAAGCCGGCCGGGAGTCG<--Suffix-->C--
BbsI

```

Part cut with BbsI and BbvCI:

```

TCAGCT<--Part-->
CGA<--Part-->TCGA

```

Figure 9: The reversing plasmid is used to reverse parts easily. The sequence for the part of the plasmid not shown is the same as the base plasmid.

Base Removal (-6) Plasmid:

```

N.BstNBI          AarI  BbvCI          BciVI          SgrAI
--GAGTctaGCTGAGG<--Prefix-->CACCTGCCCTCAGCtaGACTC<--ccdB-->GTATCCACATCGCTTCCACCCGGTG<--Plasmid--
--CTCAGatCGACTCC<--Prefix-->GTGGACGGGAGTcGatCTGAG<--ccdB-->CATAGGTGTACGCAGAAGGTGGCCAC<--Plasmid--
BbvCI              N.BstNBI

```

Plasmid cut with BciVI:

```

N.BstNBI          AarI  BbvCI          BciVI          SgrAI
--GAGTctaGCTGAGG<--Prefix-->CACCTGCCCTCAGCtaGACTC<--ccdB-->GTATCCACATGC  GTCTTCCACCCGGTG<--Plasmid--
--CTCAGatCGACTCC<--Prefix-->GTGGACGGGAGTcGatCTGAG<--ccdB-->CATAGGTGTAC  GCAGAAGGTGGCCAC<--Plasmid--
BbvCI              N.BstNBI

```

Plasmid cut with BciVI, blunted, and then cut with N.BstNBI:

```

N.BstNBI          SgrAI
--GAGTctaGC
--CTCAGatCGACTCC<--Prefix-->GTGGACGGGAGT
BbvCI              BbsI
                  GTCTTCCACCCGGTG<--Plasmid--
                  CAGAAGGTGGCCAC<--Plasmid--
                  BbsI

```

Prefix Part (cut with BbsI, blunted, and then cut with N.BbvCIA):

```

TGAGG<--Prefix-->CACCTGCCCTCAGCT<--Part-->
CGA<--Part-->

```

After ligation:

```

N.BstNBI          AarI  BbvCI          SgrAI
--GAGTctaGCTGAGG<--Prefix-->CACCTGCCCTCAGCT<--Part-->GTCTTCCACCCGGTG<--Plasmid--
--CTCAGatCGACTCC<--Prefix-->GTGGACGGGAGTcGAK<--Part-->CAGAAGGTGGCCAC<--Plasmid--
BbvCI              BbsI

```

Ligation cut with BbsI, blunted, and cut with N.BbvCIA makes a normal prefix part with -6 bases:

```

TGAGG<--Prefix-->CACCTGCCCTCAGCT<--Part(-6)-->
CGA<--Part(-6)-->

```

Figure 10: A plasmid that can be used to remove 6 bases from the end of a part is shown here. The notation $-n$ is used to indicate a plasmid that removes n bases from the end. A $+n$ plasmid removes n bases from the front. Plasmids to remove different amounts from the front or back can be constructed analogously starting from this one. For instance, to create a -5 plasmid, this plasmid can be cut with BciVI, filled in with Klenow DNA polymerase, and re-ligated to itself. In fact, this -6 plasmid is sufficient for both removing 6 bases and 5 bases from the end depending on whether one blunts the end with a nuclease or by fill-in with a polymerase. Thus, in theory, one only needs to create -6 , -4 , and -2 plasmids to be able to remove 1–6 bases from the end of a part.

Part Append Plasmid:

```

N.BstNBI
AarI
--GACTctaGCTGAGG<--Prefix-->CACCTGCCCTCAGGAGACGGg<--ccdB-->...
--CTCAGatCGACTCC<--Prefix-->GTGGACGGGAGTCTCTTGcc<--ccdB-->...
BbvCI
BsmBI
Eco31I
FseI BbvCI
... cGGTCTgAGCT<--AppendPart-->ACGTGcGTCTTCGGCCGGCCCTCAGC<--Suffix-->GCTGAGGtaGACgC---
... gCCAGAGcTGA<--AppendPart-->TCGACgCAGAAGCCGGGGGAGTGG<--Suffix-->CGACTCCatCTGcG---
BbsI
BbvCI

```

Plasmid cut with Eco31I, blunted (optional), and then cut with BsmBI/N.BstNBI

```

N.BstNBI
--GAGTCTaGC
--CTCAGatCGACTCC<--Prefix-->GTGGACGGGAGT
FseI BbvCI
<--AppendPart-->ACGTGcGTCTTCGGCCGGCCCTCAGC<--Suffix-->CCTGAGGtaGACgC--
<--AppendPart-->TCGACgCAGAAGCCGGGGGAGTGG<--Suffix-->CGACTCCatCTGcG--
BbsI
BbvCI

```

Plasmid ligated with a Part prepared for Prefixing:

```

N.BstNBI
AarI BbvCI
--GACTctaGCTGAGG<--Prefix-->CACCTGCCCTCAGCT<--Part...
--CTCAGatCGACTCC<--Prefix-->GTGGACGGGAGTCCA<--Part...
BbvCI
FseI BbvCI
...Part--><--AppendPart-->ACGTGcGTCTTCGGCCGGCCCTCAGC<--Suffix-->GCTGAGGtaGACgC--
...Part--><--AppendPart-->TCGACgCAGAAGCCGGGGGAGTGG<--Suffix-->CGACTCCatCTGcG--
BbsI
BbvCI

```

Figure 11: This type of plasmid can be used to append short and commonly used sequences (e.g., a *ssA* tag). This is not intended for general use in assembly as the append sequence has additional restrictions on disallowed sequences. One particular use is when the *AppendPart* is empty, the plasmid can be useful for directly transferring a part into a standard form directly from the base removal plasmid. This plasmid has two mutations from the base plasmid to eliminate *N.BstNBI* sites. The final constructed part will not have the standard part sequence, but all operations can still be done normally.

Part Prepend Plasmid:

```
AarI BbvCI
--GcGCTtaGCTGAGG<--Prefix-->CACCTGCCCTCAGCT<--PrependPart-->AGCTcGAGACCG<--ccdB...
--CgCAGatCGACTCC<--Prefix-->GTGGACGGGAGTCCA<--PrependPart-->TCGAGCTCTGGc<--ccdB...
BbvCI
Eco31I
BsmBI BbvCI
...ccdB-->cCGTCTCTCAGC<--Suffix-->GCTGAGGtaGACTC--
...ccdB-->gGCAGAGGAGTGG<--Suffix-->CGACTCCatCTGAG--
BbvCI N.BstNBI
```

Plasmid cut with Eco31I, blunted (optional), and then cut with BsmBI/N.BstNBI

```
AarI BbvCI
--GcGCTtaGCTGAGG<--Prefix-->CACCTGCCCTCAGCT<--PrependPart-->
--CgCAGatCGACTCC<--Prefix-->GTGGACGGGAGTCCA<--PrependPart-->
BbvCI
```

19

```
TCACG<--Suffix-->GCTGAGGtaGACTC--
CCatCTGAG--
N.BstNBI
```

Plasmid ligated with a Part prepared for Suffixing:

```
AarI BbvCI
--GcGCTtaGCTGAGG<--Prefix-->CACCTGCCCTCAGCT<--PrependPart--><--Part...
--CgCAGatCGACTCC<--Prefix-->GTGGACGGGAGTCCA<--PrependPart--><--Part...
BbvCI
FseI BbvCI
...Part-->AGCTGAGTCTCGGGCCGCCCTCAGC<--Suffix-->GCTGAGGtaGACTC--
...Part-->TCGACTCAGAAGCCGGGGGGGAGTGG<--Suffix-->CGACTCCatCTGAG--
BbsI BbvCI N.BstNBI
```

Figure 12: This family of prepend plasmids is similar to the append plasmids, providing the ability to easily prepend a sequence to a part.

BioBricks Upgrade Plasmid:

```

AarI  BbvCI  BsaI  FseI  BbvCI
--G<--Prefix-->CACCTGCCCTCAGCTGGCGGAGACC<--ccdB-->GGTCTCaTAGTACTCTTCGGCCGGCCCTCAGC<--Suffix-->G--
--C<--Prefix-->GTGGACGGGAGTCGACCGGCCTCTGG<--ccdB-->CCAGAGtGATCGACTCAGAAGCCGGCCGGGGAGTCG<--Suffix-->C--
BsaI

```

Upgrade Plasmid cut with BsaI:

```

AarI  BbvCI  FseI  BbvCI
--G<--Prefix-->CACCTGCCCTCAGCT
--C<--Prefix-->GTGGACGGGAGTCGACCGG
          CTAGTACTCTTCGGCCGGCCCTCAGC<--Suffix-->G--
          GACTCAGAAGCCGGCCGGGGAGTCG<--Suffix-->C--
BbsI

```

BioBrick Part Form

```

EcoRI NotI  XbaI  SpeI  NotI  PstI
GAATTCGGGCGGCTTCTAGAG<--Part-->TACTAGTAGCGGGCCGTGCAG
CTTAAGCGGGCGGAAGATCTC<--Part-->ATGATCATCGCCGGCGACGTC

```

BioBrick Part cut with NotI and SpeI:

```

XbaI
GGCCCGTCTTAGAG<--Part-->TA
CGAAGATCTC<--Part-->ATGATC

```

BioBrick Part ligated into Upgrade Plasmid:

```

AarI  BbvCI  XbaI  FseI  BbvCI
--G<--Prefix-->CACCTGCCCTCAGCTGGCGGCTTCTAGAG<--Part-->TACTAGTACTCTTCGGCCGGCCCTCAGC<--Suffix-->G--
--C<--Prefix-->GTGGACGGGAGTCGACCGGGCGAAGATCTC<--Part-->ATGATCGACTCAGAAGCCGGCCGGGGAGTCG<--Suffix-->C--
BbsI

```

End BioBricks++ Part:

```

GGCCCGTCTTAGAG<--Part-->TACT
CCGGCGAAGATCTC<--Part-->ATGA

```

Figure 13: One method for moving parts from the BioBricks form to the BioBricks++ form is shown here. It is assumed that the part is a valid BioBricks++ part by not containing the restricted restriction enzyme sites. The BioBricks part is cut with SpeI and NotI and ligated into the cut plasmid. The final part is in the valid BioBricks++ form but with an additional 14 bases on the front and 4 bases on the end. These extra bases can be later removed by using the base removal operations.

BioBricks Upgrade Oligos:

```
AarI BbvCI FseI
CGCCACCTGCCCTCAG CTAGTACTCTTCGGCCGGCC
GGGGTGGACGGGAGTCGATC GACTCAGAAGCCGGCCGG
BbsI
```

BioBrick Part cut with XbaI and SpeI:

```
XbaI SpeI
CTAGAG<--Part-->TA
TC<--Part-->ATGATC
```

PCR primers:

5'-CGCCCACCTGCCCTCAGCTAGAG-3' and 3'-ATGATCGACTCAGAAGCCGGCCGG-5'

BioBrick Part ligation/PCR amplified:

```
AarI BbvCI FseI
CGCCACCTGCCCTCAGCTAGAG<--Part-->TACTAGTACTCTTCGGCCGGCC
GGGGTGGACGGGAGTCGATCTC<--Part-->ATGATCGACTCAGAAGCCGGCCGG
BbsI
```

Ligation cut with BbsI and BbvCI:

```
TCAGCTAGAG<--Part-->TACT
CGATCTC<--Part-->ATGATCGA
```

Clone into entry plasmid. New part:

```
AGAG<--Part-->TACT
TCTC<--Part-->ATGA
```

Figure 14: Another method for upgrading BioBricks parts. Upgrade oligos are synthesized with compatible XbaI/SpeI ends or they are created by digesting with the enzymes. After ligation with the cut BioBricks part, a PCR is done with the given primers. The PCR should not amplify any part that was ligated in the incorrect direction. After PCR, the product can be cut and cloned as usual. The final part will have 4 extra bases on either end.