Contents

[OVERVIEW OF BUILDING BLOCK ASSEMBLY STEPS 1](#_Toc46915424)

[Troubleshooting PCR 2](#_Toc46915432)

[DNA Sequence Analysis: Error rates in building block synthesis 3](#_Toc46915433)

[Oligo Design 4](#_Toc46915434)

[Step 1: Reverse Translation 5](#_Toc46915435)

[Step 2: Recoding the genome using the preferred codons of S. cerevisiae 5](#_Toc46915436)

[Step 3: Other options for recoding the genome 5](#_Toc46915437)

[Step 4: Inserting Watermarks/PCRTags 6](#_Toc46915438)

[Step 5: Creating oligos 7](#_Toc46915439)

[Step 6: Optimizing oligonucleotide melting temperatures. 7](#_Toc46915440)

# OVERVIEW OF BUILDING BLOCK ASSEMBLY STEPS

## Step 1: Oligo design

The goal of building block synthesis is to assemble a “building block (BB)” of ~750 bp. We use software to divide the BB into single -stranded DNA pieces that are 60-80 bp long (oliogonucleotides or oligos). Individual oligos are then ordered from commercial suppliers.

## Step 2: Templateless PCR

Templateless PCR is performed to form a larger construct (the building block, or BB) from mixtures of oligonucleotides.

## Step 3: Finish PCR and Gel electrophoresis

Finish PCR is performed to amplify the full-length building block that is produced in small quantities in the templateless PCR. The success of the finish PCR is gauged by running a portion of the finish PCR reaction on a gel.

## Step 4: Ligation and Bacterial Transformation

Building blocks that are of the correct size are cloned into a genetic cloning vector and used to transform bacteria. Bacteria are transformed with the cloned construct to create a population of clones for sequencing and future assembly into larger synthetic yeast constructs.

## Step 5: Colony screening PCR

We verify that each bacterial clone has a DNA insert of the correct size by performing PCR and gel electrophoresis.

## Step 6: DNA Sequence Analysis

Bacterial clones containing building blocks that are the correct size are sequenced to make sure that there are no sequence errors. Once the DNA has been sequenced, we compare the DNA sequence of each clone to the desired sequence to identify those without sequence errors (we call these “perfect clones”). This completes BB synthesis.

## Step 7: Troubleshooting PCR

Since many BBs will not assemble correctly the first time, we change the templateless PCR and finish PCR reaction conditions to enable assembly

# Troubleshooting PCR

Building block assigned \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Expected size of BB (bp) \_\_\_\_\_\_\_\_\_\_\_

BB assigned \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Expected size of BB (bp) \_\_\_\_\_\_\_\_\_\_\_

BB assigned \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Expected size of BB (bp) \_\_\_\_\_\_\_\_\_\_\_

What was the problem with your original finish PCR reaction for each building block (no DNA product, multiple distinct DNA bands, smeared DNA band)?

What is your hypothesis for why your PCR did not work?

What did you change to try to get your finish PCR to work (annealing temp changes, extension temp changes, additional rounds of PCR, etc.)? Remember, each of these BBs must be synthesized eventually, so as much information as you can give us about what has already been tried to troubleshoot each specific BB will be very helpful.

What was the result and how would you interpret it (attach gel images of your original PCR and you troubleshooting gel)? Is this PCR product ready to be ligated into the vector? If not, why not?

# DNA Sequence Analysis: Error rates in building block synthesis

For analysis 3, there are 4 different clones of the same building block. Align the sequences for each of the 4 clones to the reference building block and identify all of the mutations in those 4 sequences. **Only record a “mutation” if the forward and reverse reads agree with each other and disagree with the building block sequence. Also, only record a mutation if it falls within the bounds of the building block sequence.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sequencing ID | Name of Building block | Length of BB (nt) | # insertions | # deletions | # transitions | # transversions | Total # mutations |
| Analysis 3-1 | RFP | 714 |  |  |  |  |  |
| Analysis 3-2 | RFP | 714 |  |  |  |  |  |
| Analysis 3-3 | RFP | 714 |  |  |  |  |  |
| Analysis 3-4 | RFP | 714 |  |  |  |  |  |

Total # of mutations in all 4 sequences combined: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Total # nucleotides sequenced (ie size of your BB x 4): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Calculation of error rate

We would like to know the overall error rate for creation of our building blocks (we call this value α) since this information will help us to determine the efficiency of our method and protocols. The error rate can be calculated as follows:

Error rate per nucleotide (α) = (Total # mutations found)/(Total # nucleotides sequenced that are not vector sequence).

For example: if you found 13 mutations in 4 clones of a 750 bp BB, then α=13/(4\*750) = 0.001

From the information above for the building block you analyzed, the error rate per nucleotide α = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The error per building block (μ) is the error rate per nucleotide (a) multiplied by the length of the building block in nucleotides (L). For your building block, m=\_\_\_\_\_\_\_\_\_\_\_\_

## Calculating probability of obtaining a perfect clone

What is the probability of a clone having no errors? The number of errors in a building block follows a Poisson distribution where x represents the number of errors. For a Poission distribution, the probability of outcome x is *P*(*X*)=*e-mmx* . If x=0 (zero errors), then the equation simplifies to p(0) = e-m.

X!

Using the equation p(0) = e-m, what is the probability that any one of your clones is error-free? \_\_\_\_\_\_\_\_\_\_\_

## Calculating probability of a batch of sequences having at least one perfect clone

However, what we really need to know is how many clones we need to send for sequencing. We want to send as few clones in a batch as possible (sequencing is expensive!) while still having a high probability that the batch will contain one clone with a perfect sequence.

If p(0) is the probability that a clone will have no errors, then 1-p(0) is the probability that a clone will have one or more errors.

If B is the number of clones in a batch, then (1-p(0))B is the probability that at least one of the clones in the batch will have an error.

The probability of having at least one clone in the batch without an error (pb) is pb=1-(1-p(0))B .

Based on your calculation above for the probability that a clone will be perfect (p(0)), what is the probability that a batch of 8 sequences will have a perfect clone?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A batch of 5 sequences?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Oligo Design

Our goal is to build a synthetic gene from oligonucleotides. The GeneDesign program allows us to take the sequence of our synthetic gene and break it up into the set of overlapping nucleotides for assembly into a synthetic gene.

The GeneDesign program allows us to make additional modifications to the gene. One of these is **codon optimization**, which is a strategy of using the most common codons to encode each amino acid in the hope of maximizing protein production. In performing codon optimization, we do not want to change the sequence of amino acids, but we want to change which codons encode the amino acids. This is termed **recoding the genome**.

To recode the genome, we must obtain the DNA sequence of the building block (synthetic gene) that we will be synthesizing. We can derive a DNA sequence from the protein sequence by “reverse translating” the protein to obtain a nucleotide sequence (note that for any protein sequence one can obtain multiple different DNA sequences depending upon which codons you choose to use).

For this exercise you can use the sequence of your synthetic gene or you can use this amino acid sequence for practice:

MAKINGSYNTHETICGENESTHISISMYFAVRITEGENEWHATISYRS

(we are using the one letter abbreviation for amino acids as shown in the table below)

|  |  |
| --- | --- |
| **Abbreviation** | **Amino acid** |
| **A** | alanine |
| **C** | cysteine |
| **D** | aspartic acid |
| **E** | glutamic acid |
| **F** | phenylalanine |
| **G** | glycine |
| **H** | histidine |
| **I** | isoleucine |
| **K** | lysine |
| **L** | leucine |
| **M** | methionine |
| **N** | asparagine |
| **P** | proline |
| **Q** | glutamine |
| **R** | arginine |
| **S** | serine |
| **T** | threonine |
| **V** | valine |
| **W** | tryptophan |
| **Y** | tyrosine |
| **X** | stop codon |

## Step 1: Reverse Translation

To design our oligonucleotides, we will use a program called Gene Design, available at [http://54.235.254.95/gd/*.*](http://54.235.254.95/gd/)

Click on “Design a Gene”. Enter the amino acid sequence of your building block (using the single letter amino acid code). For example, a sequence could be: MAKINGSYNTHETICGENESTHISISMYFAVRITEGENEWHATISYRS

Click on “Reverse Translate”. You should get a nucleotide sequence as an output from the program.

## Step 2: Recoding the genome using the preferred codons of S. cerevisiae

Synonymous codons are codons that encode the same amino acid. We can recode the genome by substituting synonymous codons for the original codons. Organisms tend to favor certain synonymous codons over others (these are called preferred and non-preferred codons), a phenomenon we term **codon bias**. Preferred codons are associated with more efficient translation while non-preferred codons are associated with less efficient translation. If we use all preferred codons when recoding the genome, we are “optimizing the genome”.

Click “Back” on your browser. Notice that at the bottom you can select the organism that you are working on. This table will then show you the most preferred codons for each of the 6 organisms (H. sapiens, S. cerevisiae, E. coli, B. subtilis, D. melanogaster and C. elegans).

1. For which amino acids is the same codon preferentially used in all 6 species? How many differences are there in preferred codon usage between S. cerevisiae and C. elegans?

Select *S. cerevisiae* as your organism and click on “reverse translate” to obtain a nucleotide sequence. The nucleotide sequence that you are given now uses the preferred codons of *S. cerevisiae* to encode your building block. This potentially alters the nucleotide sequence without altering the protein sequence.

1. When thinking about gene expression, what is one advantage to using the preferred codons to encode your synthetic gene(s)? By changing the DNA sequence, what are some potential problems that we may encounter at the level of the RNA?

## Step 3: Other options for recoding the genome

At the bottom of the web page, click on “Codon Juggling”. This will give you several different versions of the sequence: an optimized version, a most different sequence, etc. Scroll to the bottom of the page and notice that all of the sequences encode the same amino acids.

The overall GC content of yeast is 38%. Notice that for each of the nucleotide sequences that you are given the %GC is very similar. The program is designed to maintain the GC content of the DNA when selecting which codon to use for each amino acid (in order to obtain the proper GC content, the most preferred codon is not always the one that is used at each position).

1. Why might we want to control the %GC in each building block? What properties of the DNA sequences would be affected by a change in the %GC?

## Step 4: Inserting Watermarks/PCRTags

When we create synthetic DNA, we need to mark that DNA as synthetic by inserting watermark sequences. GeneDesign will not do this step, instead we need another program, Watermarker. The Watermarker program recodes a portion of the genomic sequence that you enter so that the amino acids are preserved but the nucleotides are changed. Our goal is to change the nucleotide sequence by at least 33%.

Go to the Watermarker program (http://watermarker.loyola.edu/). Copy and paste the sequence of your gene into the genomic sequence box. You can use the Standard Genetic Code or switch to another organism. [Note that you may need to copy and paste the sequence 4-5 times to make it large enough for this program]. Click “Generate Watermarks”.

Once you have successfully generated watermarks, record:

Sequence of Left Watermark: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Start position of left watermark: \_\_\_\_\_\_\_\_\_ End position: \_\_\_\_\_\_\_\_\_\_

Sequence of Right Watermark: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Start position of right watermark: \_\_\_\_\_\_\_\_\_ End position: \_\_\_\_\_\_\_\_\_\_

Expected size of PCR product: \_\_\_\_\_\_\_\_\_\_\_\_\_

Melting temp of left watermark: \_\_\_\_\_\_\_\_ Melting temp of right watermark: \_\_\_\_\_\_\_\_

Percent changed left watermark: \_\_\_\_\_\_\_ Percent changed right watermark: \_\_\_\_\_\_\_

## *Step 5: Creating oligos*

Now that we have our nucleotide sequence, we need to have DNA chemically synthesized. However, only single stranded DNA can be chemically synthesized. Rather than ordering one piece of DNA that is the length of our building block (750 bp), we order our DNA as oligonucleotides of approximately 60 bp.

We order our oligonucleotides as single-stranded DNA and then use enzymes to synthesize the complementary strand, thereby producing double-stranded DNA. We design our oligos to partially overlap one another (we term these “gapped oligos”). This allows the end of on oligonucleotide to serve as a primer for DNA synthesis.

1. Each oligonucleotide can only be extended in one direction. Why? On the figure below depicting overlapping oligos, fill in the missing DNA nucleotides and use an arrow to indicate the direction in which each section of DNA is synthesized.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5’ | 3’ | 5’ | 3’ |  |
| CT | AGA CCC GCT | CAT | GCA TGT CCA ATG CAT |  |
|  | GGG CGA TGA | CAC GTA | CGT TAC GTA CAC GTC | TGA |

## Step 6: Optimizing oligonucleotide melting temperatures.

One of the reasons that we need the Gene Design program is that the program will adjust the length of the oligos to design oligos that have the same Tm in their overlap regions.

1. Why might it be important to have a uniform Tm when assembling oligos into a building block

Go back to Gene Design at <http://54.235.254.95/gd/> and click “Building Block design (constant length overlap)”. Make sure the settings are:

“Target assembly oligo length” =80 bp

“Return Assembly oligos with an overlap of Tm” = 56 degrees.

Copy and paste your sequence from the Watermarker program and click “Design oligos”. Notice that in the white box at the top you see four DNA strands: the top and bottom strands are the two complete strands of the building block and in between them you see the overlapping oligos that can be used to synthesize the building block.

Hit the “Back” button on your browser. Change the parameters to:

“Target assembly oligo length” =80 bp

“Overlap melting temperature” = 64 degrees

“Maximum Assembly Oligo Length” = 100 bp.

1. What about the oligos has changed? How would this affect the stability (Tm) of the double stranded DNA?
2. What is the advantage to using longer oligos for gene synthesis?

Go back and change the parameters again to:

“Target assembly oligo length” =60 bp

“Overlap melting temperature” = 56 degrees

“Maximum Assembly Oligo Length” = 80 bp.

1. What about the oligos has changed? How would this affect the stability (Tm) of the double stranded DNA?
2. What would be the advantage to using smaller oligos for gene synthesis? (Think about your answers about error rates above).

Click “Assembly oligos (tabbed format)” and submit the list of oligos for one building block along

with your assignment.