Notes on Synthetic Biology

Engineering Principles in Synthetic Biology - Prof. Richard Kitney

DNA Assembly – Dr. Geoff Baldwin

Control and Dynamics of Genetic Networks – Dr. Geoff Baldwin

Synthetic Biology for Biosensors – Prof. Paul Freemont

Modelling Biosensors – Prof. Paul Freemont

Parts – Dr. Tom Ellis

Logic Networks – Dr. Tom Ellis

Genome Synthesis – Dr. Tom Ellis

Optogenetics – Prof. Mark Isalan

Pattern Formation – Prof. Mark Isalan

Engineering Principles in Synthetic Biology - Prof. Richard Kitney

Fast Fourier Transform

Each of samples in frequency domains are frequency components.
If you have a sampled data waveform with N samples, you end up with N frequency components in the spectrum.
By looking at real spectrum, you see there is a mirror image at a midpoint fs/2, where fs is sampling frequency. This relates to sampling interval T, fs= 1/T. fm <= fs/2.
Imaginary spectrum: start from 0, N components. Key difference from real spectrum, is that the reflection is an inverted mirror image. If you make a power of 2, then the process of doing calculation speeds up (e.g. N= 128, 1024, 4096 etc...).
If we take two random individual components at the same frequency, these have real values (e.g. a and b). So, each frequency components there is a real part and an imaginary part of the component.

Two recording of thermoregulatory experiment

Patient has a glove where hot or cold water can be controlled (periodic thermostimulus) and record blood flow in the other hand. The point is that if you stimulate right hand, what happens on the left hand is direct measurement of nervous control (it's not like hot passing from right to hand).
A square wave thermostimulus was applied. Bottom graph is blood flow on the left hand without any thermostimulus. It appears quite random.
On top graph, there is some structure. However, the interpretation of these waves in time domain is difficult. This is why we look at them in frequency domain.
If we look at spontaneous spectrum (the top one in the second slide corresponding to the bottom one in the slide before). You get a spread of frequency components throughout the spectrum.
In the spectrum of the stimulus bottom graph, you see a spike at 0.05 Hz, that is the same as stimulus frequency. It is entrained to the stimulus.

Analogue to Digital Conversion

How do we determine what T needs to be for a particular experiment?
In practice, there is unit associated with the computer you store the data in, the pulse generator (second graph), produces a series of vertical pulses each of the same amplitude and separated by interval T. x(t) time p(t) produces z(t). So it is a multiplication in the time domain. In the frequency domain (b) it is easier to see.

Aliasing:

Nyquist sampling criterion: in order to avoid corruption of data (when two spectra overlap), make sure that sampling frequency has to be grater or equal than 2fm.

Promoter Measurement Workflow

Example of protocol for constitutive promoters. The key point is that they do the assay over 6 hours but sampling is every 15 minutes.
Reference J23101 over three days. Pretty clean set of data. If we compare this with over run over three days with j23101. There is a particular set of run, where there is

“contamination”, lots of variance. The question is, is this data produced by aliasing or problem with the biology?
Look at the sampling time of T=15 mins. There is a lot of variation in this time window. Actually what was going here is a biological problem. Here T (=15min) fulfils the criterion of 2 fm.
We need to reduce sapling interval. Fluctuations could appear between the two points. If time interal is too big than fluctuations are more important and when you interpolate the points you might lose info.

How do we carry out filtering in terms of frequency domain?
The continuous waveform goes from ADC process to Fast Fourier Transform. You find the frequency domain component. This then has to undergo frequency domain filtering. Apply a high-pass filter. In frequency domain, this high-pass filter actually comprises sets the frequency components from 0 to 0.2 Hz to 0 amplitude. You are left with the frequency component of the respiratory component. This has to be done twice in the real part and twice in the imaginary part.
Then do inverse Fourier transform and get back the filtered time spectrum.

The Stages of Engineering Biology

- **Characterisation** (lab robots)

- **Automation** (foundries)
  Imperial founded International Consortium on Foundries. What we are looking for is reliability and reproducibility. London DNA foundry was funded by EPSRC, InnovateUK, BBSRC and Imperial College itself. All equipment is linked together by an information infrastructure. This allows to use “best-of-breed” approach, pretty straightforward to exchange pieces of equipment. It is moving to WhiteCity Campus. It comprises three platforms: liquid handling (load well-plates), then characterization platform, then assembly platform.
- **Addressing Complexity**
- **Industrialization**
  Usually now R&D is done by university. Even pharma companies now tend to do less basic research. Once something interested is found than they take over. Process:
  R&D → Systematic Reproducibility → Manufacturing

Information Systems and Information Integration
SynBIS is information system developed by Imperial College. Underlying architecture: 4 layer. Top layer is interface layer (HTML), below is the communication layer (transfer images, DNA data – using the DICOM standard and XML), then application layer (specialist softwares), final layer is database layer (the registry, a SQL structure database, commercial).
In one way we have done characterization experiment ourselves and so data are in right format, these go straight into SQL database. However, different people characterise different bioparts. Here we need to go through 2 steps. First thing is to check If the format of the data and see if it corresponds to format into SynBIS and SQL. If not, convert it using a standards converter (intimately related to DICOM SB model). Second step is to do the compliance check (around 70 different fields and

simply run though the protocols, data, metadata etc; checks those against a template for that particular kind a part). If it does not pass the test, then ask the collaborator to re-characterise it or characterise it yourself using the foundry. They are now incorporating a catalogue of models. The future is to put a comparator between model and part.
How do we deal with all the different DATA formats, of different companies (e.g. Agilent). You need to introduce a Converter (e.g. Web-Based DICOM SB converter). This converts the data into this common format. Kitney et al have done it for a lot of different equipments.

Five Hard Truths About Synthetic Biology

1. Many of the parts are undefined (poorly characterised)
2. The circuitry is unpredictable
3. The complexity is unwieldy
4. Many parts are incompatible
5. Variability crashes the system

Baldwin 1 - DNA Assembly

"The lack of standardization in assembly techniques for DNA sequences forces each DNA assembly reaction to be both an experimental tool for addressing the current research topic, and an experiment in and of itself." Tom Knight, MIT

The challenge for synthetic biology is to develop standardised assembly methods allowing work at all levels of abstraction – genes, pathways and genomes – and to clearly understand the context dependencies when parts are physically placed next to other parts

Idempotency: assembled parts retain the prefix and suffix of the original, allowing successive rounds of hierarchical cloning.

BioBrick Assembly

This is a standardised restriction enzyme assembly protocols, developed by Tom Knight. A BioBrick is a DNA unit with standardised flanking sequences.

→ Pros:

Idempotency: standardised prefixes and suffixes.

BioBrick = DNA unit with standardised flanking sequences that enabled assembly to be achieved by a cheap, simple and standardised restriction/ligation method.

→ Cons:

The major downside of the BioBrick approach is that the same 8 bp scar sequence is found at every junction. The presence of this scar sequence is unacceptable at

certain positions, notably the RBS, meaning that alternative assembly methods must be used in cases where context-dependency is a problem. The scar is also problematic when assembling fusion proteins as it encodes an in-frame stop codon.

More recently, a standard called BglBricks has been described[19] that uses different sequences for assembly and leaves a smaller 6 bp scar. This encodes a simple glycine-serine motif in frame, making the method more amenable to protein fusions. BglBrick assembly also has the advantage of using highly efficient and commonly-used restriction enzymes whose recognition sequences are not blocked by the most common DNA methylases, Dam and Dcm.

Despite revisions and new standards, neither BioBrick™ nor BglBrick methods can assemble a scarless gene from parts and crucially cannot assemble every sequence of DNA as the use of restriction enzymes means that the sequences they use as recognition sites are forbidden within a part

Gibson Assembly

Sequence-independent overlap technique. It is an isothermal assembly method, using a high-fidelity DNA polymerase, T5 exonuclease and *Taq* DNA ligase.

In demonstrating these methods, Gibson *et al.* successfully assembled a complete synthetic 583 kb *M. genitalium* genome *in vitro* from four 100 kb+ fragments. They also showed their protocols to be efficient with fragments at the 2 kb scale and have recently adapted the enzyme ratios in their preparation to allow the method to be used to assemble genes directly from single-stranded 60-mer oligos that overlap by 20 bases, bypassing gene synthesis.

This protocol has allowed Gibson *et al.* to successfully assemble the entire 16.3-kilobase mouse mitochondrial genome from 600 overlapping 60-mers using only the Gibson isothermal assembly method at all stages.

→ Pros:

It allows to assemble five or more parts together without forbidden sequences.

→ Cons:

However, it is not standardised, but customised parts are usually produced via PCR amplification. So essentially every reaction is a bespoke assembly, each requiring its own verification, optimization. Also, reliance on PCR limits fidelity and possibility for automation.

BASIC Assembly
Method that uses restriction-ligation reactions to ligate orthogonal oligonucleotide linkers with ss overhands that define the assembly order.

BASIC brings together six key concepts: standard reusable parts, single-tier format (all parts in the same format), idempotent cloning, parallel (multipart) DNA assembly, size-independence, automatability.
As in BioBricks, the use of prefix and suffix confers idempotency: assembled parts retain the prefix and suffix of the original, enabling successive rounds of hierarchical cloning.
The method is based on MODAL, which introduced the concept of computationally derived orthogonal linkers. BASIC is based on restriction/ligation reactions to ligate orthogonal oligonucleotide linkers with single-stranded overhangs that define the assembly order. This is achieved by using a standard format that facilitates the reuse of both linkers and parts.
The standard involves using BsaI to release a DNA part from a storage vector, leaving a 4 bp scar on prefix and 6 bp on suffix. Digestion then yields different 4bp overhangs at prefix and suffix, enabling end-specific ligation. Linkers are thus attached by simultaneous restriction and ligation. Unligated excess linkers are then removed via magnetic bead purification.
To generate the final construct, linker-adapted parts are mixed and annealed in an ionic buffer at high temperature. No ligase is required for final step and nicked plasmid is repaired in vivo followed transformation

However, sometimes it is useful to assemble a set of parts ina module and then combine different modules to create more complex systems or reuse modules in different assemblies. Therefore, BASIC requires an idempotent method by which iP and iS can be reused. To do so, DNA methylation is used to protect the BsaI sites.
The cognate DNA methyltransferase of the BsaI restriction modification system is a C-5 methyltransferase, but its target within the BsaI recognition sequence is not known.
Restriction digests using the cognate BsaI methylase clearly reveal that methylation of the bottom strand only partially protects the DNA from digestion, while methylation of either cytosine in the top strand effectively protects the DNA from digestion by BsaI.
They demonstrated that methylation of a single cytosine in the BsaI recognition sequence provides sufficient protection against BsaI digestion to enable an idempotent strategy without modification of the protocol. Maintaining the same protocol for all stages of assembly and for all parts ensures an easy workflow for both bench-scale work and automation.

Physical DNA standards are the integrated Prefixes and Suffixes (iP, iS).

- → Pros:
  Linkers can be used as composable parts, encoding RBSs or peptide linkers for fusion proteins.
  Idempotent
  Standardised
  Parallel
  No PCR required
  Orthogonal linker sequences provide positional watermarks in the final assembly, thus they may be used to validate assemblies since they provide ideal PCR primer sites.
  Long overlap efficiency
  Size independence

Automatability
No ligase
→ Cons:
Not scarless

Baldwin 2 – Genetic Networks: Control and Dynamics

Programmable biological control is done via manipulation of genetic networks. Schematic of an enzyme that activates its own production. Differential equation with hyperbolic function synthesis term and degradation.

$$\frac{d[E]}{dt} = \frac{k1 * [E]}{Km + [E]} - d2[E]$$

Hyperbolic drops out from binding equilibria, Michaelis-Menten kinetics. Degradation is dependent on concentration of protein species, times a degradation term. When synthesis is balanced by degradation, the change in concentration is zero. This is where the two rates (nullclines) intersect, the steady state.

Robustness is an engineering principles but very common in biology (imagine homeostasis). Robustness is the opposite of sensitivity. **Sensitivity** is a measured term in applied sciences. ***Sensitivity coefficient***. The amount by which an output of the system (e.g. enzyme concentration) changes with respect to variation in an input quantity (e.g. degradation rate).

Sensitivity coefficient $\frac{dln[P]}{dln\,(d2)}$

For the dynamics of the simple biological system above, a small change in degradation rate leads to a relatively large change in enzyme concentration. This is system is thus not robust (i.e. it is very sensitive to changes in d2).

In biology many systems contain feedback control (like cruise control). In abstraction, this requires a certain process, which gives some outputs, a sensor senses the output and the information is passed to a controller (e.g. if speed is too high or too low) and an actuator changes the input.
This feedback control is very observed in genetic networks.
A constitutive system is where there is no negative feedback. Feedback reduces the overall amount of protein
To produce the same amount of protein without protein requires a weaker promoter
Feedback reaches the equilibrium position faster than a non-repressed system
Because it is working from a stronger promoter.

The speed of controlled response is not the only advantage of negative feedback. Negative Feedback can also be used to reduce noise. Becskei and Serrano (2000) built a tet repressor protein fused to GFP. TetR being produced represses its own production. Now use flow cytometry to measures GFP intensity (distribution of fluorescence across a population of cells). There is a tight distribution of GFP fluorescence. Then make a mutation in tetR gene (Y42A), which is involved in operator site. Now there is a much broader distribution than negative feedback.

Then changed tet operator site into a lac operator site. Now transcription factor cannot bind there. Here also wide distribution of cells.
Negative feedback is limiting the dynamic range.

Opposite is positive feedback. For example, when a transcription factor enhances its own rate of production. Positive regulation introduces a delay and leads to more instability (can lead to bistability). In principles this can have two steady states. In positive feedback it takes longer to get to max. This is counterintuitive. Positive regulation is slower than negative feedback! The reason for this lag is that in the absence of protein, there is no protein driving the production. At t=0, there is no A present and so nothing to activate expression. At some dt there is going to be a stochastic expression burst in a cell that produces a mRNA, which then will produce few copies of protein, this then kick starts the positive feedback. This is observed as a bimodal distribution of cells in Flow cytometer histogram.

Positive and negative feedbacks are usually embedded in larger motifs. Feedforward loops are motifs that exist in larger cellular networks. FFLs consist of 3 genes:
Regulators X and Y that control gene Z
8 possible permutations of activation and repression of the two control genes
This is independent of the genetic logic at Z, it just defines the input status from X and Y
Coherent: signs down both branches are the same
Incoherent: signs down both branches are different
Uri Alon (2007) demonstrated that coherent and incoherent type I occur much more commonly in nature than the other 6 motifs.
The coherent type I FFL can act like an AND gate (sustained input detector or persistence filter). If you turn X on, there is no immediate Z. There is a delay. You need build-up of Y as well. Once x goes to zero z also declines to zero. There is no delay in the disappearance after input is removed (a sign sensitive delay, it will delay the on phase but not off).
In bacteria, ara operon follows FFL type I. x = CRP, Y = AraC, and Z= araBAD. cAMP is input to activate CRP. The OFF state does not have delay. Indeed you see the delay of the output.

With an incoherent FFL, in the presence of external activator X, the X branch is turned on. If we turn on Y, though this will turn off Z. So Y branch is on when Y is absent. The prediction form modelling is that there is pulse dynamic in Z. Steady state level of Z depends on strength of Y as a repressor
Tight repressor; Z=0
Moderate repressor; Z reaches steady state balance.
Induction of galactose operon ad type I iFFL. This acts like a pulse generator and a response accelator (the activation branch on the left). X= CRP, Y= GalS, Z= galETK.

If you have repeated pulses this can lead to oscillations.
Stricker et al., (2008) constructed a robust oscillator with iFFL. They made promoters that can activated by araC and repressed by LacI (in the absence of IPTG). They achieved this by engineering dual promoters. They added degradation taks into proteins. With tags they can have cycles of 13 min, which is pretty fast.

Optogenetics – Mark Isalan

Ways of using light to trigger gene expression. Bacteria can be engineered to see light via expression of receptions. Light is more reversible than chemical inducers. Light is already used in biological systems (e.g. photosystems). *Halobium* uses a proton pump and this is driven by light (used as an energy transduction system). Two components kinase systems can trigger signalling using light.

Image processing: from bacterial 'photography' to the edge detector.
Two component systems usually exists with one component being a membrane bound sensor, this autophosphorylates itself and the P is transferred to another protein. The phosphorylated form is in on state and this can act as a transcription factor. To achieve that you can make a hybrid protein, one domain from phytocrome (PCB) and one autophosphorylation domain (EnvZ- OmpR). This kind of system is on in the presence of dark. In the presence of red light, the system is off. Link this system to lacZ as output (they used S-gal and not X-gal, is converted into black). So in absence of light, the system is on, produces lacZ and produces black pigments (so black output). Then you can project an image with strong contrast (in dark areas of picture lacZ is produced and in bright areas you are white).

Can we design bacteria to "see" and process "images"?
In biology, complex functions can be achieved by collective of "simple" agents.
The goal was edge detection. Take projected images and detect the borders of that using bacteria (signal processing). What logic do we need to apply?
There are gonna be bacteria on the loght, some in the dark and some at the border.
One logic is to say if you are in light produces some red (actually is an AHL molecule), if in dark, produce blue.
If red and blue, produce color.
An alternative way is to say:
If in dark, produce a diffusing chemical compound.
If in light and sense diffing chemical produce colour.
Then you need to go from AND logic to DNA sequence. Now essentially we have a dark sensor (same PCB fusion system using phicobilin chromophore). In dark, then

you express an AHL. Then AHL can freely diffuse across the cell membrane. If you are in the light and sense AHL then you produce color.
Cells in light would not produce any cph8 and you express luxR.
Cells in dark, would produce cph8, then synthesise quorum sensing molecule. Also NOT logic here by expressing a repressor (cl), which makes sure that the lacZ gets turned off. So that cells in the dark do not turn black.
Those are the edge they are in the light so the cph8 machine stays off but can sense the AHL. AHL binds LuxR, lucR activates expression from pLux promoter and this triggers expression of lacZ.
Then this design should output color only at the edge.

It is important to maintain the contrast. If AHL diffusion is too high then edges would blur away. Test sensing, communication and inverter components. Put them altogether and end up with the edge detector. Indeed, that is what you see. If you project a square there is a corner artefact so more diffusion!

Parts Library for Complex Synthetic Biology – Tom Ellis

iGEM parts registry is a library of DNA parts, in BioBricks format. Mostly they *are E. coli* parts. There is by far more parts than any other SynBio database. The problem is that characterization is quite poor and not standardised. More professional part registries include synberc (sponsored by a particular sponsor so not easy to access all the information). Another place is addgene, non-profit company that shares plasmids from publications.
We are going to talk about the efforts to make more standardised parts.

Inverter Network:
NOT gate: constitutive promoter, RB, transcriptional regulator, terminators, the transcriptional regulator regulates a promoter upstream of a reporter. It is hard to make cells that all have this device and work at the same time. Predictability is problem. Problems include burden and lack of orthogonality. Orthogonality allows to have in the soup of the cells to have many wires and no short circuits. We can achieve this by having lots of sequence-specific transcription factors. So orthogonal promoters allow to have same effect as insulated wires.

Constitutive Promoter:
Simplest part is the constitutive promoter. In bacteria this is particularly straightforward to make many copies of (we call this a library of part). These have -35 and -10 consensus. To be able to diversify these DNA sequences you can do conservative mutations of the consensus sequences (slightly better or worse promoters) or keep consensus sequences the same and change bases in between of the two consensus, you can change conformation of DNA and you can get different effects (call this synthetic library).
Constitutive E. coli promoters are short enough to be encoded on a primer. With these primers do PCR around plasmid, ligate and transform in E. coli. Characterise them with plate reader. And see different outputs.

The most famous library is the Anderson promoter collection, he used the conservative mutation in -10 and -35 region to be able to have a catalog of promoters of different strengths.

Ribosome Binding Site:
Can be as short as 5-7 bases (usually 10-15). Short enough to make them de novo. This was worked by Chris Voigt. They reasoned that the role of an RBS is that when in E.coli the promoter start producing the mRNA, this is a signal that recruits the small subunit of the ribosome and begins translation from the ATG codon. The rate of translation is determined by rate of initiation (rate determining step). The initiation rate itself is largely dictated by interaction between RBS sequence and the part of the 16S rRNA within the ribosome 8so just interaction between RNA and RNA). Shine-Dalgarno sequence (RBS) bunds the anti-SD-sequence in the rRNA. Also RNA folding determines the rate of translation. You can calculate how sequences are good at these interactions by using Gibbs free energies (as we know the energy of base pairing). So RBS calculator looks at all the total change in free energy, when the mRNA binds ribosomal rRNA + start site connection + change in free energy in folding the spacing region minus the standby energy of folding minus also the free energy of the mRNA that is folded on itself. So calculating this gives you a prediction for any given sequence. RBS calculator can do reverse engineering (produce DeltaG of my sequence) or forward engineering (give me a RBS sequence with the specified delta G).
Standby and start site are all taken into consideration when how good this RBS is. So you need to know everything around (the parts cannot be described in its isolation, so there is context dependency).

Context is so a problem for synthetic biology. For example set is the world in the English language with the most definitions. To understand the meaning of set we need to know in which sentence is placed. Same as context-dependent parts. So generally combining many different promoters, RBS and CDS parts doesn't lead to predictable gene expression output.
How can you alleviate problem of context dependency?

1. You understand enough so you can predict its effect (RBS calculator does this by modelling the effect of upstream and downstream sequence of the RBS calculator). However, you need to understand a lot…
2. Use parts that themselves are designed to remove context dependence (insulators).

An example pf the second strategy is the use of a ribozyme (folds up into something that is like an enzyme but is RNA and cuts itself of). So if you put RiboJ between promoter and RBS, it leaves a clean 5'UTR.
Another method is the Csy4 method. Is an enzyme that cut a short specific RNA sequence that makes a hairpin fold. So similarly, any RNA will have the same end.

Terminators
RNA based parts. Very boring. Tell RNAP to stop making gene. They are encoded that maked the mRNA forms a stem-loop structure using a palindromic sequence. It is not a good idea to use the same one repeatedly, because of their repeated bases. Biofab designed hundreds of terminators and used model to design new ones. Essentially they placed terminators in between an upstream and an downstream

genes. An ideal one would produce lots of upstream gene and none of the downstream.
Then Chris Voigt characterised 500+ terminators and used data to make a "terminator calculator".

Regulators and Regulated Promoters
Key parts that enable to build logic systems. In E.coli you see the same transcription factors and promoter pair a lot of time (there is always LacI, pLAC, araC and araBAD etc). So it's basically a set of six wires. This is a problem because if you want to do fanciers stuff with more than one NOT gates, we need hundreds of them!
Without a large orthogonal sets of predictable regulatros the complexity of synthetic biology cannot increase.
One way is to look at TF that happen in nature and are already modulars. A good one is Zinc-finger proteins (little alpha helices that recognise thre bases at a time based on 4 amino acids). It is very difficult to deisng them.
Khalil et al., 2012 made a great set of synthetic TFs paired to synthetic promoters. However they only work in yeast, not in E. coli.

Next thing is a new type of DNA binding protein, TAL effectors. These are individual fingers that end up with 2 amino acids that specifically recognise one base. Much easier to design (one finger module per base). So you can target all the sequences you want. But again this does not work in *E.coli.*

Until Chris Voigt did some part mining to find sets of orthogonal regulators. Let's find things that look like the tet repressor and synthesised those sequences and mutated them and characterised them. They were able to get a 15 different promoters that are all specifically repressed by things like tetR from different bacteria. They are all different based on threshold switches and different sensitivities.
So this is the best so far transcription factor library.

CRISPR
Bacterial immune system where RNA sequences are repurposed by bacteria and guide Cas) to cut DNA specifically. So you have a an enzyme that is directed to a precise DNA location by a guide RNA.
If you make a version of Cas9 that I deactivated but interaction stills hold. So if you target it to the promoter you block the binding of transcription factors and so it can act like a repressor. So Stanley Qi showed that you can use CRISPRi to knock down transcription from the targeted promoters. The only things that displaces dCas9 is DNA polymerase.
Also a good thing is that you can express different gRNAs at the same time, to combinatorially target many loci. This is a nice example of system multiplexing.

Remember that also expressing CRISPR guides you don't need an RBS (as usually do if you want to make inverters with repressors). Nielsen and Voigt also put dCas9 gene downstream of TetR promoter (so you have inducible expression of the inverter logic).

Bacterial Logic Gates
Martin Buck and Kitney made AND gates (by using sigma factors).

Then you can tie this together with a transcription factor that acts a NOT gate, giving a NAND gate (AND+NOT).
You can get to a NAND gate also by ting together NOR gates (universal gates). You can get all the gates by putting together a lot of NOR gates. So NOR gates can be used to do all the logic that you need. This was used very early on in electronics, where loads of NOR gates are wired up to perform some sort of computation. The Apollo Moon Mission Circuits was made of NOR gates. In total it was just 5600 gates. So you can get to the moon and back with that.

Chris Voigt looked at way in which bacteria can be used as loads of different NOR gates. In this system, each cell is a gate. A system to distribute the computation around different cells. It showed that in *E. coli* you can do implement NOR logic. Each cell is a NOR gate with different input/outputs. Wires are diffusible signals. You can design a NOR gate from a NO19:15 gate by placing a tandem promoter upstream of a repressor gene. For example, pBAD-ptet –repressor. So only if you have two inducers you have the repressor transcribed.
In 2016, Voigt group published a paper showing automated design of complex logic functions… all inside an *E. coli* using a CAD software (Cello). For this they used that 16 NOT gates with orthogonality.
Only with the RiboJ you get predictable behaviour. So insulation is very important. Another important factor is the lack of repeated parts in the same plasmid, otherwise you end up with homologous recombination between them and deletion of some DNA.
Some of those gates are not as good as the others. The range of outputs of one gate needs to match with the other gate's input range. Output1 is going to be the input for gate 2 and the switch-like change of the inverter switch needs to match the switch-like range for gate 2.
Testing many combinations reveals ones with the best ranges and those not to use. After optimization of the gates that best worked together, add all the info into a software for rational and automated design (CelloCAD), which uses Verilog (it is a hardware description language, a text description of digital circuits). So CELLO is like Verilog but for *E. coli* logic circuit design. Inputs for CELLO are sensors (like inducers). Outputs are usually fluorescent proteins or other TFs.
You can make priority detectors and all sorts of staff.
Now you can do the logic with CRISPRi. The caveat is that it is not very digital. As gRNA follows a linear function, whereas TFs have Hill functions.

Why don't we do 100+ logic gates. The problem is that bacteria cannot do all that extra work. Extra DNA programs are a burden that slows growth and impairs performance. This goes against the E. coli number one goas: takes sugars and grow better than its neighbours.
Two solutions to reduce overloading a cell:

1. Use programs that are less costly to the cell
2. Distribute program among a population (i.e. multicellular genetic networks).

Less costly programs can be RNA devices. You can do logic with RNA, you can get an order of magnitude less expensive. You can use riboregulators and riboswithces. Even less costly with RNA is just to do it with DNA. However DNA is mostly inert so you need still some proteins. You can do logic via DNA recombination using

integrases. You can flip DNA into ON and OFF states (inverted) by inducing expression of integrase. If you nest integrase genes between recombination sites (if you have 11 for example you can encode 1 byte of digital information.
If you sequence then the DNA, you can see which event happened and in which order. So very clever way to encode digital information in DNA.

Genome Synthesis – Tom Ellis

A description of the recent application ns of synthetic biology. Circuits and logic gates is in the rounds of app designing at the end, but instead that in smartophojnes you install them in cells.

1. First Synthetic Genome and Craig Venter

Synthetic genomes came out of initial works investigating minimal cells. The minimal cell is an interesting question. Which machinery is required to Make a living things. A self-sustaining central dogma. Lot of effort into understanding the major machines involved in the central dogma. Proteins and RNA component involved have been mostly all crystallised. So there is a minimal recipe. Is not actually a cell, is essentially a self-replicating broth.
Estimates is 151 genes required. 2 genes for gene replication. Transcription only need a1 gene. This cell however would need all the metabolites, as metabolic genes have been taken out. It would be very fast evolving sequence. A primordial life system. To go beyond that add more genes. For example, metabolism for amino acids. As long system is compartmentalised it would self-sustain. Another approach is to find the cells that can self-sustain with the smallest genome. Compare them and find homology, to find genes that are absolutely essential as they are in all the minimal cells. About 50 to 380 are conserved throughout life. If you delete genes of small genomes, you get that 400 hundred are absolutely required. However, we still do not know what these genes are actually doing.
Craig Venter set up the Craig Venter hiring nobel laureates and invested a lot of money to build a minimal cell. The aim is to build a minimal cell by synthesis a reduced genome. In 2010 they announced success.

Step 1: Genome chemical synthesis (2008)

This project was very helpful for synthetic biology in terms of assembly techniques. They had to worked out how to assemble the genome at the chemical level. The taking genetic info and transforming it into DNA
To do this they had to synthesise ca. 10,000 DNA 50-mers and assembled them into a complete 583 kbp *M.genitalium* genomes
DNA synthesis companies synthesised 101 pieces of 5 to 7 kbp from overlapping oligos.
These 101 pieces are then recombined using enzymes in vitro to make 24 big pieces. These 24 pieces were maintained in BACs in E. coli and recombined to make bigger pieces. Then these pieces were transferred in yields to make even bigger pieces and homologous recombination to knit them together.
This was done by Dan Gibson, who created the stepwise gibson method. This works by mimicking what happen when a piece of DNA breaks. The cell then thinks this is a problem. So DNA machineries trim back the ends of the DNA and they do this by looking for homology. When homology is detected and are then knitted back together. Gibson worked pout that if in vitro you recreate this environment you can get this sort of repair of DNA ends that have the same sequences. The Gibson method requires T5 exonuclease, Taq ligase and Phusion Polymerase. At 50 degrees the fragments are chewed back and the polymerase fills the gaps.
You do this in vitro and then you put in vitro. First in e. coli then put it into yeast. Yeast is cool in that you don't even have to add enzymes as yeast is an expert in homologous recombination. All you need to do is add linearised fragments of DNA with homology ends and yeast will do the assembly work. This method is called Yeast assembly or TAR cloning.
A cool way to change bacterial genomes on demand is to take the genome linearise it and put into yeast, do the genome editing changes in yeast and then take it out and insert it back into the bacterium.

Step 2: Clean DNA in cell (Lartigue et al., 2007)
DNA need to be booted-up in a cell. Cell starts to use a new synthetic genome and not the old one
The AIM is essentially to put genome A into cell B, so to turn cell B into cell A>
To get DNA into cell B it requires cell fusion (mycoplasma has no cell wall)
Genome B does not have antibiotic resistance so it gets lost. Using sequencing, proteomics and phenotyping they confirmed that *M. capricolum* turned into M. mycodies.

Step 3: Combine step 1 and step 2 to boot-up a cell from a synthesised genome (2010)
They have assembled the entire genomes from oligos using Gibson assemblies. What was surprising is that it was not *M.genitalium* and was not *M.laboratorium* the reduced one., It was *M. mycoides,* the one that was booted already.

But is this a minimal cell ? *M. mycoides* is >1 million bp. *M. genitalium* is 550,000 bp.

2. DNA Assembly for Synthetic Genomes

3. Whole Cell Modelling

In 2010 they announced the first whole cell computational simulation of a bacterial cell. Taking info from data mainly from JCVI, this makes predictions and models. 900 publications are mined for these data. Key important is connections. Polymerastion of the wall is almost always modeled in stochastic modelling. Metabolism is rarely modeled with ODEs. Lots of different ways to do modelling. Here instead of one model with ODEs, they are modeling single processes individually and using dynamics. So it is tying together lots of models into one. They are all interconnected models.

You can quantify energy usage of the cell, to do replication and other processes. Making protein is by far the largest cost. Membrane transport is also a big cost. Then they used this model to see what happens to cell when you delete genes.

In 2016 a paper came out. They took the genome they took the mycoides genome and reduced it down. They reduced to be smaller just smaller to genitalium. They used the same process. Deletion of genes leads to worse growth. They kept 473 genes. They even changed the order of the genes and put them in categories depending on functions like an engineering project and amazingly it worked.

4. Sc2.0

It would be nice to the same as synthetic bacteria but in eukaryotes… let's do this in yeast! The project started from John Hopkins University 2009 iGEM team. Undergrads working on building synthetic parts of an 11 million bp genome with 16 chromosomes. In 2011 they first announced the first two parts of chromosome are synthetic. When they are synthesising they are changing the sequences of the genes. Not like JCV putting texts and emails just to show off.

They designed the sequences that they wanted, from companies they bought chunks and linked together using restriction enzymes.

Small oligos are assembles using Polymerase Chain Assembly (like building a wall, parts are make to overlap and then PCR is run).

Getting this into yeast is quite simple. If homologous region is detected the yeast knits them together as he thinks there is an error in replication. Do reiterative recombination to select for synthetic sequences in yeast chromosome. This is a serial process.

In 2014 they finished the first chromosome. This was done mainly by undergrads. Then different chromosomes are g9iven to different labs. Ellis lab has chromosome 14.

Different designs are added to the synthetic genomes. All the transposons are deleted as they make genomes unstable. Remove all the introns. What are they for? Spoiler alert is that some are needed. All the tRNAs are moved from chromosomes into a separate chromosome.

LoxP sites are also added to be recognised by Cre to do Scramble. If you turn system on, all the cells die excepts a few. Some survive and sometimes they canb express betters. This is a way to minimise the genome for you but get the biology to do that.

Also swat all the TAG codons (there are three different stop codons) into TAA. You can free up soime codons to insert unnatural amino acids.

5. RE.coli
George Church is swapping codons and deleting some to free up codon spaces. Is this orthogonal xenobiology? The code is rewritten and new codons are incorporated. This e. coli is more resistant to phage attack due to orthogonality. It can be made to be dependent on non-standard amino acids for growth for biocontainment.

6. Applications of synthetic genomes
One reason is understanding by building. Design cells for specific tasks.
For custom synthesis of products.

Pattern Formation – Tom Ellis

Cells differentiate into specialised cells. This task differentiation generates a pattern, morphogenesis.
Patterning theories:

- Growth, clock and rule-based (e.g. dendrite growth)
- Gradient models (e.g. French flag)
- Reaction-diffusion systems (Turing patterns).

Maternal cell un Drosophila blastoderm there is already a morphogen gradient (bicoid).

A synthetic multicellular system for programmed pattern formation (Basu et al., 2005).
They went into *E. coli* and built from modular parts a system for French flag pattern formation. Developed a synthetic gene network that acts as a band detector. It is like a band-pass filter (but is not filtering the signal but only detecting a band). This is an incoherent feed-forward loop. When AHL is high, makes lacI dominating the system and tunring GFP off. If there is low AHL, GFP on. If no AHL, no GFP. So GFP expressed only at medium AHL levels.
You can have different mutated plasmids, enabling to tune the band.
The half-life of LacI is a key parameter that determines the kind of observed parameters.

Essays on Synthetic Biology

Genome Synthesis:

- **How JCVI made the first cell with a synthetic genome**

Creating the first cell with a synthetic genome at the JCVI took several years, lots of money and several scientific challenges. The task involves generating a self-sustaining cell using DNA from chemical synthesis. Interestingly, this is the first time that the operating information encoded in the genome is not passed by a mother to a daughter cell but is written by a human on a computer.
Work on minimal cells helped to find the minimal genetic requirement to synthesise a genome. Using this, the goal of Venter was also to remove one by one non-essential genes from this smallest genome. This is a reductionist approach, in contrast to a comparative approach, where genes common to all organisms indicate the minimal genetic requirements for life.
To achieve this dream of synthetic life three main technical obstacles needed to be overcome: synthesise a genome chemically (it is a pretty huge molecule), prove that bacterial cells can be transplanted with other genomes and finally combine the two previous steps.

1. Chemical Genome Synthesis (Gibson et al., 2008)

The first question is perhaps which genome to synthesise. Of course, the smaller the genome the easier it is. *Mycoplasma genitalium* is a good candidate as it has the smallest known genome of a bacterium capable of independent growth.
The genome was assembled from shorter pieces of DNA that were chemically synthesised. The short molecules are then knitted together to form larger chunks. To do this a new DNA assembly method was developed based on overlap. It is an in vitro recombination method.
With this method, the JCVI constructed the 582 kbp genome of *Mycoplasma genitalium*. Synthesis of a genome is a hierarchical process. First oligodeoxiribonucleotides were ordered by DNA synthesis companies and then assembled them together by overlap annealing. They developed a method to assemble sequences of DNA together in a sequence-independent manner, and scar-less, now called Gibson Assembly.

Assemblies of up to quarters of genomes were cloned in vitro in E. coli bacterial artificial chromosomes

2. Genome Transplantation (Lartigue et al., 2007)

A whole bacterial from one species is inserted into the cell of another bacterial species. Following this “genome invasion”, the native host genome has to be overpowered by the intruder genome. This is achieved simply by cell replication, where the two genomes of the original cell end up in the two daughter cell. Isolation of the cell containing the alien genome (by marker selection) yields a transplanted cell.
Here they used *M. mycodies* (Mmc) as the donor and the related species *M. capricolum* (Mcc) as recipient cells.
Mmc and Mcc, as with all mycoplasma species, do not possess cell walls, which may be significant for the entry of large DNA into recipient cells.
Successful selection of Mcc cells harbouring Mmc genomes yielded the transplanted cell “Synthia”.
It was hard to achieve as methylation interfered. Mmc and Mcc have similar methylation protection systems so when donor Mmc genome is replicated in yeast it gets unmethylated and so inside Mcc it gets destroyed. They solved the problem by methylating the DNA in vitro and then knocked out also the restriction enzyme in Mcc.

3. “Boot-up” (Gibson et al., 2010 and Hutchinson et al., 2016)

Once the designed DNA molecule, whether it is a gene, pathway or genome, has been obtained, the challenge is to boot it up and then to propagate it. Boot-up and propagation involve transcribing, translating, replicating and regulating the information contained within the DNA to produce the desired effect or phenotype. They reported successful transplantation of a synthetic Mmc genome (JCVI-syn1.0) into Mcc cells. The only DNA in the cells is the designed synthetic DNA sequence, including watermarks (names and Feynman quotes). However, they did not introduce the M. genitalium genome synthesised by Gibson et al. (2008) but the one from M. mycodies, as they could only transform Mcc cells with Mmc genome (as reported by Lartigue et al. 2007).

Notwithstanding the importance of this project, it should not be overinterpreted as synthesis of a cell, or life, as standard usage of “synthetic” would imply synthesis of the cell material as well as just a synthetic genome. A cell built *ab initio* should at least contain the following components: an information storing molecule (probably DNA), transcription/translation extracts to carry out gene expression in vitro, a synthetic vesicle to hold it together and avoid dilution of protein synthesis rate. To allow self-reproduction, the membrane of the cell should also contain pores to allow the continuous feeding of the internal reactions with external nutrients.

- **Compare and contrast JCVI project and Sc2.0**

From JCVI project we learned that there is a trade-off between genome size and growth rate
Now there is Sc2.0 and Human Genome Project-Write. Also JCVI showed how dynamic life is, that entire genomes can be swapped to retain still functional cells.

- **DNA Assembly method used for synthetic genome**

The 583,000 M. genitalium genome was divided into 101 oligomers (each of about 6 kbp) and each with sequences overlapping neighbouring oligos. These pieces were ordered from DNA synthesis companies (who built them from ca. 10,000 DNA 50-mers). Using in vitro enzymes (DNA Pol without dNTPs, DNA pol as fillase and DNA ligase) they assembled they recombined these oligos into 24 larger sequences. These 24 molecules were propagated in E. coli in bacterial artificial chromosomes (BACs) to yield 4 quarters of genome. These 4 linear molecules of DNA were transformed into yeast using YACs. Homologous recombination between the 4 chunks yielded a complete circular *M. genitalium* genome.

In (Gibson et al., 2008) they performed this recombination method in a thermocycler and using T4 Polymerase without dNTPs as 5'-exonuclease, and so individual reactions were carried out in only two steps. In (Gibson et al., 2009) they improved this two-step thermocycled method by using exonuclease III and antibody-bound Taq DNA polymerase, which allow for one-step thermocycled in vitro recombination.

- **How to model a whole cell and why**

- **How to make synthetic yeast chromosomes**

Important is the CEN element in the YAC vector, which corresponds to the yeast centromere. This allows the vector to be stably maintained as a chromosome for homologous recombination of the synthetic DNA sequences.

- **Rational vs. automatic re-factoring of genomes**

Rational is for example expanding the genetic code by removing synonymous codons and incorporate non-canonical amino acids. Automatic is like MAGE and SCrAMBle.

- **Uses of recoded *E. coli* genome**

"The origin of life cannot be discovered, it can only be reinvented" Prof. Eschenmoser

The degeneracy of the canonical genetic code allows the same amino acid to be encoded by multiple synonymous codons

By recoding bacterial genomes, it is possible to create organisms that can potentially synthesise products not commonly found in nature.

Genomically-recoded organism are interesting as they have an orthogonal genetic code, making them isolated from viral attacks and other forms of genomic invasion. Genetic isolation is achieved because DNA acquired from viruses, plasmids, and other cells would be improperly translated, which would render GROs insensitive to infection and horizontal gene transfer

Recoding the genomes also allows to free up some codon space for the incorporation of non-canonical amino acids (Jason Chin).
Huguenin-Dezot et al., (2019) developed a strategy for incorporating 2,3-diaminopropionic acid (DAP) into recombinant proteins, via expansion of the genetic code. They show that replacing catalytic cysteine or serine residues of enzymes with DAP permits their first-step reaction with native substrates, allowing the efficient capture of acyl-enzyme complexes that are linked through a stable amide bond. Good for click and protective chemistry (as you can block and unblock cysteines).

- **Applications for modelled cells and engineered genomes**

Synthetic genomics has the potential to design, synthesize and boot-up novel pathways or even organisms leveraging the immense sequence generated from genomic and meta- genomic studies allowing for the production of medically or industrially important products.

Metabolic engineering

Synthetic genomics embodies the ambition that whole genomes might be ‘booted up’ to allow the studying of organisms without developing specialized genetic tool-sets for them

On the importance of DNA Assembly in Synthetic Biology

DNA sequences have made marked changes in our ability to understand and engineer biological systems. Advances in assembling DNA into longer and longer pieces have led to methods to construct large enzyme complexes, entire metabolic pathways and even complete genomes. DNA assembly methods are in effect a form of hierarchical polymer synthesis

- Polymerase Chain Assembly

Of these one of the first PCR-dependent method used to assemble DNA synthons from oligonucleotides is PCA. Using PCA, a dsDNA sequence is divided into oligonucleotide sequences (typically 60 – 80 nt), which encode both strands of the DNA duplex with overlaps between adjacent oligonucleotides that range from 15 to 25 nt in length. Typically, adjacent oligonucleotides are designed with gaps between

the forward and the reverse overlapping regions of the assembly oligonucleotides to reduce the amount of oligonucleotide synthesis required to synthesize a given sequence. Once designed and synthesized, the substituent oligonucleotides for an assembly are pooled together in equimolar concentrations and cycled in a one-pot "assembly" reaction in which adjacent oligonucleotides are randomly extended in a nonexponential manner by a DNA polymerase to produce a mixture of oligonucleotide extension products of various lengths. This mixture is then used as a template to seed a second PCR reaction, in which the desired "full-length" product is amplified from the assembly mixture in the presence of an excess of the outermost assembly primers.

- BioBrick Assembly

Although restriction enzyme-based cloning techniques have been the main choice for manipulating DNA constructs for a couple of decades and were the basis of early BioBrick and similar assembly methods, the need to simplify the cloning/assembly process while reducing the limitations on sequence design has led to the development of scar-less restriction-enzyme-free cloning and assembly techniques

- Gibson

Of these methods, Gibson assembly is probably the most commonly used to assemble multiple pieces of DNA together into larger constructs. This meth- od uses a one-pot isothermal technique, which uses an enzyme mixture containing a thermostable DNA polymerase, DNA ligase, and exonuclease to chew-back, anneal and repair adjacent overlapping DNA sequences to assemble the desired construct. Recent innovations in the design of unique overlapping sequences to direct the assembly process has further expanded the usage of the Gibson assembly method for combinatorial assembly of large DNA sequences.

- BASIC

The development of automatable robust chemistries for chemical DNA synthesis over the last 40 years has contributed to the advancement of our understanding of biology and has laid the groundwork for the predictable engineering of biological systems. Synthetic DNA is central to the development of methods to engineer bi- ology and when combined with the massive amounts of sequence data being generated by NGS efforts will contribute to the advancement of synthetic biology toward applications hereto- fore unimaginable. To date, there have been a handful of moonshot demonstrations such as the complete synthesis of an entire yeast chromosome (Annaluru et al. 2014), an entire bacterial genome (Gibson et al. 2008), and the sub- sequent synthesis of a minimal bacterial genome (Hutchison et al. 2016), which illustrate the use of synthetic DNA and the capabilities of existing gene synthesis methods to accomplish large-scale synthetic biology efforts.

Optogenetic Control for Synthetic Biology