Exploration Session Week 7: Computational Biology

Melissa Winstanley: mwinst@cs.washington.edu

(based on slides by Martin Tompa, Luca Cardelli, Emily Fox)

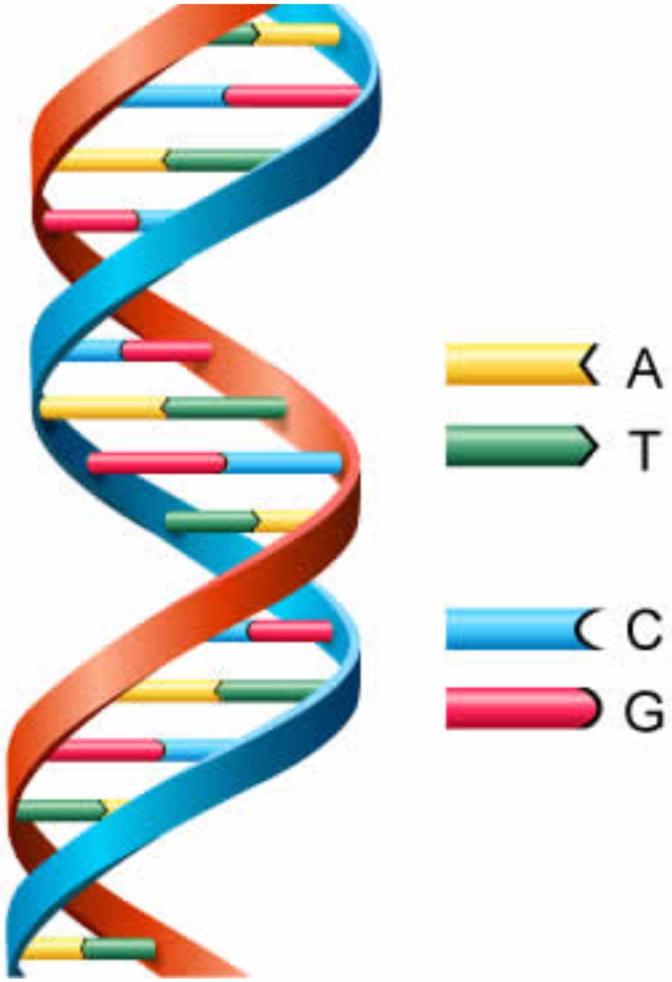
Computational biology

- Machine learning
- Statistics
- Big data
- Algorithmic design

Exploring DNA Sequences

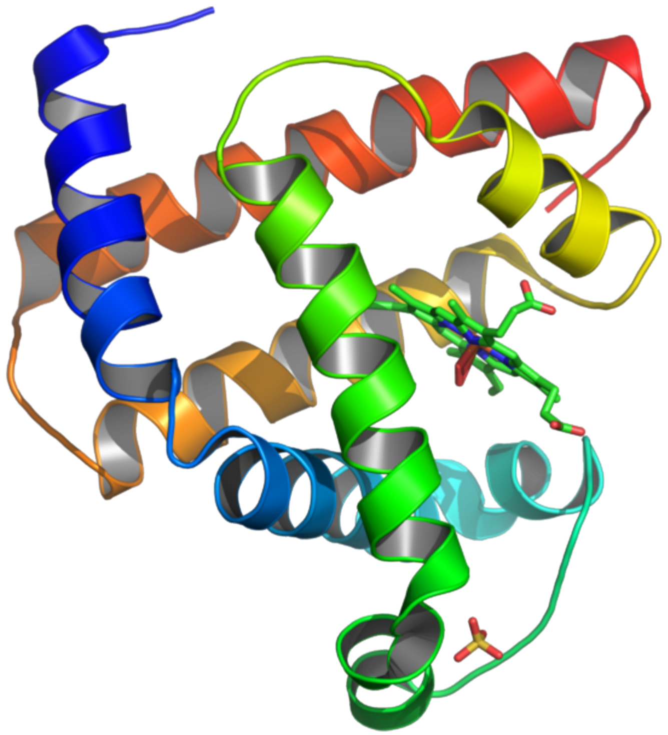
Overview of DNA

- Instructions for cellular function
  - Building proteins
- Composed of *nucleotides*
  - Adenine, thymine, cytosine, guanine
  - A pairs with T, C pairs with G
- Double-stranded: forms a double helix
  - Strands have an *orientation*
  - Pairing of antiparallel strands
- Huge amount of DNA
  - 3 billion base pairs, 2m long in a cell
  - 133 AU long in human
  - 20 million light years long in human population



Overview of Proteins

- Workhorses of cells
- Composed of sequence of *amino acids*
  - 20 to 5000 amino acids in a protein
- 20 possible amino acids
- Proteins fold into complex 3D shapes
  - Fold-It
- Information to make proteins encoded in DNA
  - *Codon*: 3 base pairs
  - Ex. CTA → leucine
  - *Gene*: sequence of DNA for 1 protein



http://upload.wikimedia.org/wikipedia/commons/thumb/6/60/Myoglobin.png/542px-Myoglobin.png

Overall Goals

- Overall
  - Identify key molecules in organisms
  - Identify interactions among molecules
- Computational focus: sequence analysis
  - Identify genes
  - Determine gene function (what protein is produced?)
  - Identify proteins involved in gene expression
  - Identify key functional regions
- Why do we care?
  - Determining function of a new sequence
  - Genetic diseases
  - Evolution

String Alignment

- How to judge how well two strings are aligned?

```
acbcdb          a c - - b c d b
        ───→
cadbd           - c a d b - d -
```

- Each dash represents an inserted space
- Assign +2 to every exact match, -1 to every mismatch

  3 * 2 + 5 * (-1) = 1

- Higher score indicates a greater match between the strings

DNA alignment

- How to approach this?
- Insight: there is a recursive algorithm!
- What are the possible alignments?
  - letter and dash
  - dash and letter
  - letter and letter
- If we knew the optimal alignment of all but the first characters, we could decide which combination was best and return that alignment
- Recursive backtracking ;)

Example

```
acbcdb          a c - - b c d b
cadbd           - c a d b - d -
```

- Option 1: letter and dash

```
a    +    cbcdb        → score = -1 + alignment of rest
-    +    cadbd
```

- Option 2: dash and letter

```
-    +    acbcdb       → score = -1 + alignment of rest
c    +    adbd
```

- Option 3: letter and letter

```
a    +    cbcdb        → score = -1 + alignment of rest
c    +    adbd
```

In reality

- This is the real strategy for computing alignments
- BUT it’s redundant and inefficient
  - Why?
  - Because order matters
- In real life, use a different algorithm
  - Not recursion
  - *Dynamic Programming*

BLAST Algorithm

- "**B**asic **L**ocal **A**lignment **S**earch **T**ool"
- For comparing biological sequence information
  - Amino acid sequences (proteins) or nucleotide sequences (DNA)
- Inputs
  - A query sequence Q
  - A database D of sequences
- Output
  - Sequences from D that match Q above a certain threshold
- Usefulness
  - Unknown gene in a mouse, so query the human gene database to see if a similar gene exists in humans

BLAST ctd

- Make k-letter subsequences from Q

  Ex. k = 3:

  “acbcdb”

  “acb”
  “cbc”
  “bcd”
  “cdb”

- Usually k = 28 for DNA, k = 3 for proteins

BLAST ctd

- For each subsequence w, find matching subsequences
  - Only consider a matching subsequence if its alignment score is greater than some threshold
  - Alignment(seq) >= T

Ex. T = 2, w="TCG"

seq = "TCA" → Alignment = 2 * 2 + 1 * (-1) = 3

Considered

seq = "ACT" → Alignment = 2 * 1 + 2 * (-1) = 0

Not considered

BLAST ctd

- Scan the database for exact matches with the high scoring subsequences
- Take each exact match and extend in either direction (no gaps)
  - Until the score decreases below a "dropoff"
  - Forms a "high-scoring segment pair" (HSP)
- Only save match extensions above a certain score threshold S

Exact match

Query seq:	A	C	T	C	G	G	C
Database:	G	C	T	C	A	G	T
Score	-1	2	2	2	-1	2	-1

HSP: score = 2 + 2 + 2 – 1 + 2 = 7

BLAST ctd

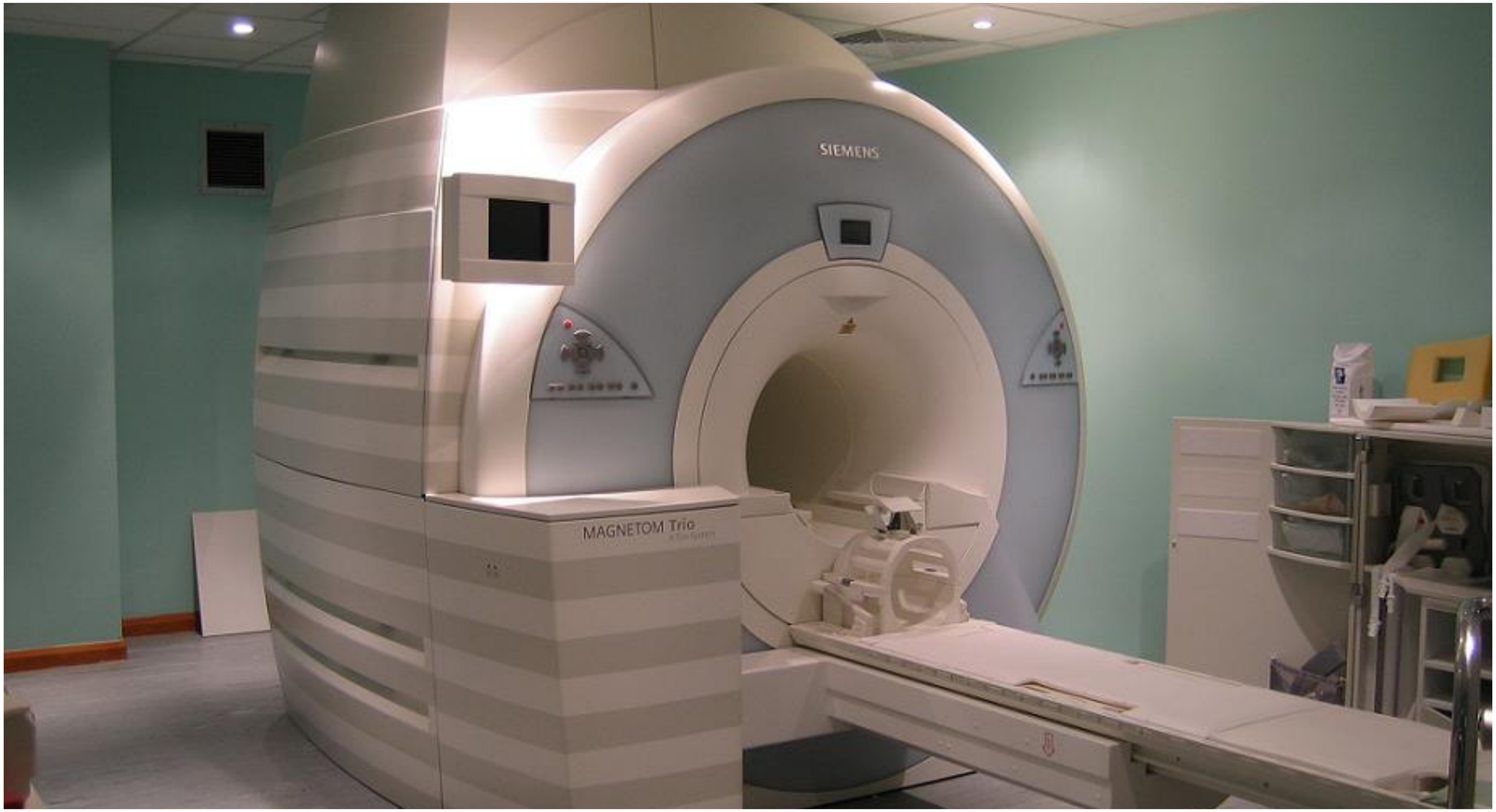
- For each segment pair, do a gapped extension (spaces possible)
- Output each extension that has probability of randomly occurring below a pre-set threshold x

More Complicated Analysis

- Multiple sequence alignment
- Different ways to score subsequences
- Considering context around a sequence
- Predicting 3D structures of proteins

A task from a course project…

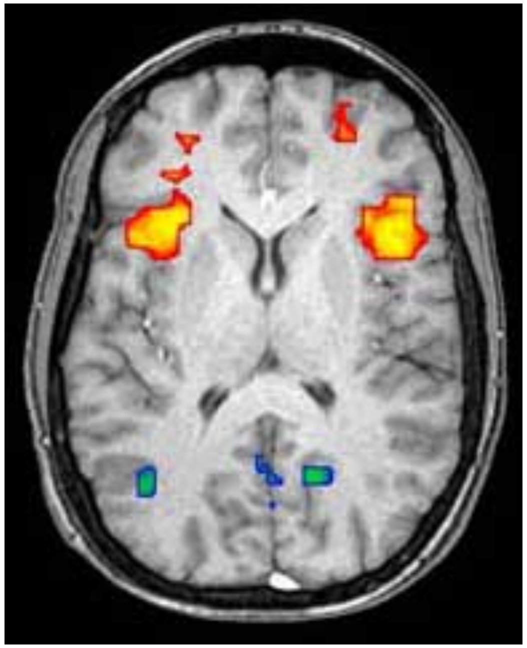
What is this?



http://singularityhub.com/wp-content/uploads/2009/04/fmri_machine_scanner.jpg

fMRI goal

- Goal: predict word stimulus from fMRI image

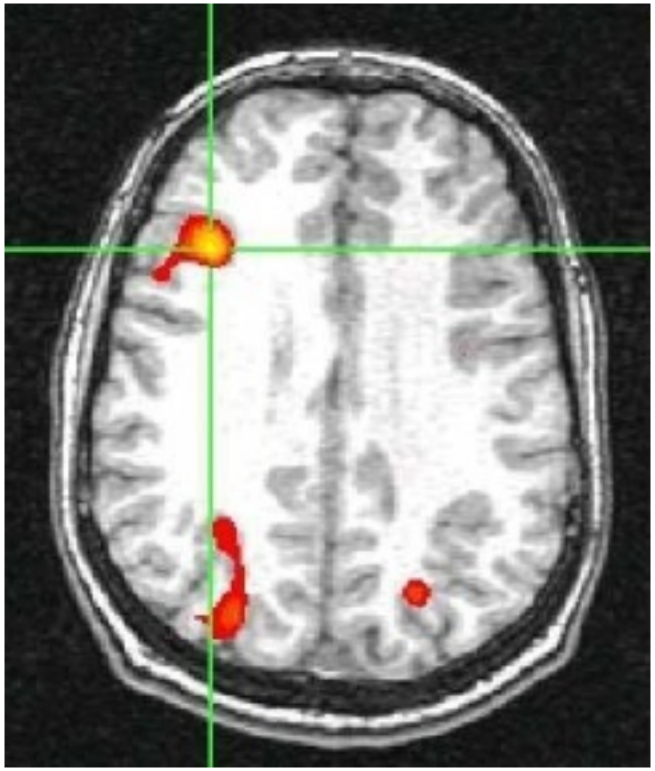


→ classifier → ~~hammer~~ or house

http://upload.wikimedia.org/wikipedia/en/d/d1/FMRIscan.jpg

About fMRI

- ~1 mm resolution
- ~1 image per sec.
- 20,000 voxels/image
- safe, non-invasive
- measures Blood Oxygen Level Dependent (BOLD) response

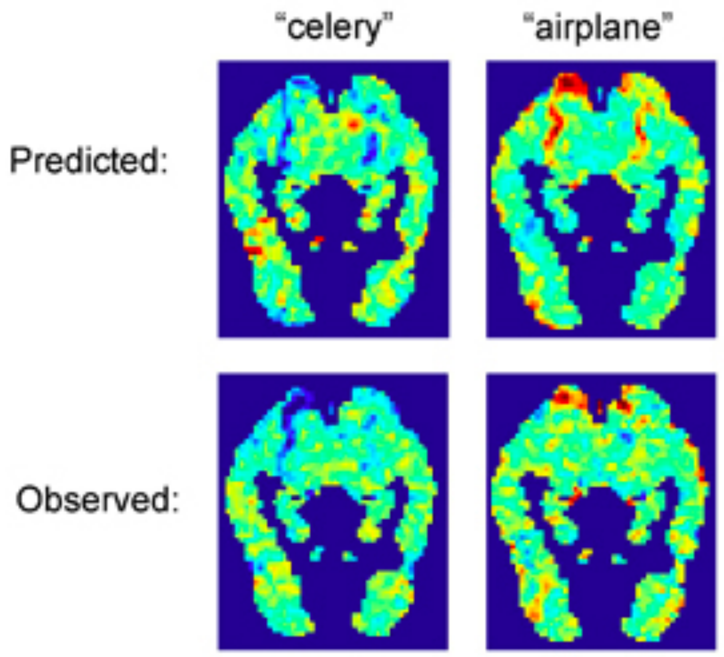


http://www.medphysics.wisc.edu/research/fmri/research/images/fmri1.jpg

Input

- Show a bunch of volunteers a series of images of objects

- See how their brain reacts

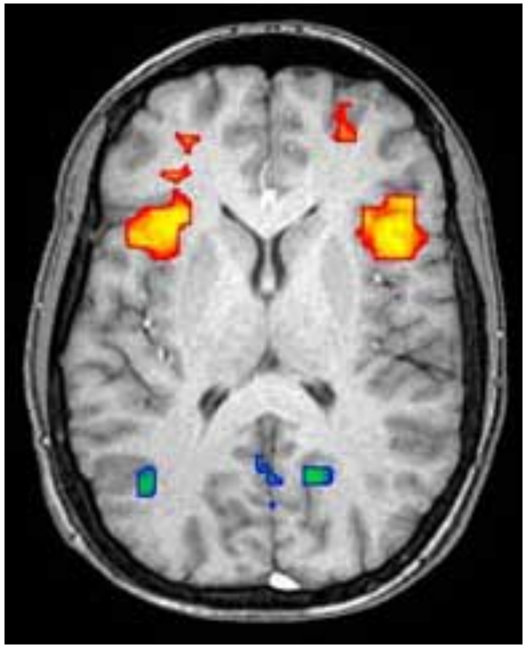


Problems

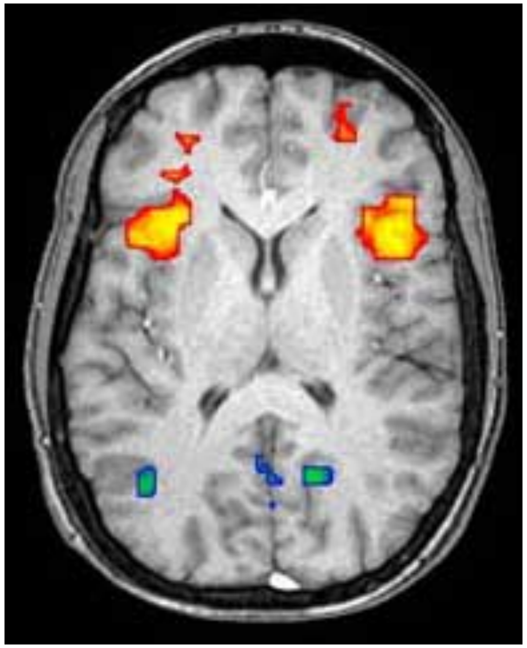
- MANY variables impact the result
  - 20,000 voxels = 20,000 variables (*features*)
- Not many observations
  - fMRI image takes time
  - fMRI image is expensive
  - Only a few examples per word
- Not comprehensive
  - Can’t test every word

Zero-Shot Classification

- Goal: guess word we've never examined before



→ looks like →



?????? house? boat?

Use semantic features

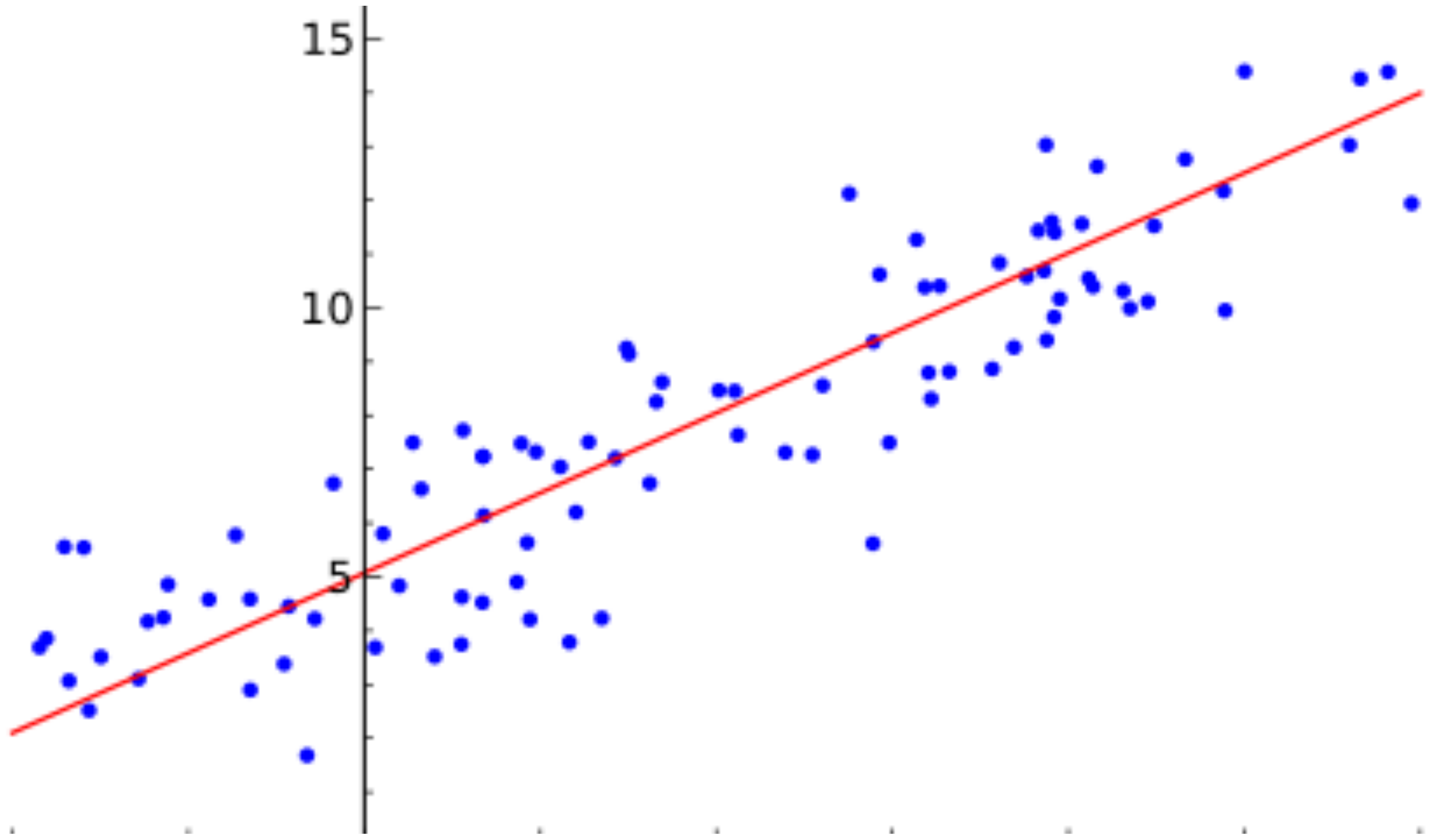
Semantic feature values: “airplane”

0.8673,	ride
0.2891,	see
0.2851,	say
0.1689,	near
0.1228,	open
0.0883,	hear
0.0771,	run
0.0749,	lift
…	
…	
0.0049,	smell
0.0010,	wear
0.0000,	taste
0.0000,	rub
0.0000,	manipulate

Semantic feature values: “celery”

0.8368,	eat
0.3461,	taste
0.3153,	fill
0.2430,	see
0.1145,	clean
0.0600,	open
0.0586,	smell
0.0286,	touch
…	
…	
0.0000,	drive
0.0000,	wear
0.0000,	lift
0.0000,	break
0.0000,	ride

Remember this?



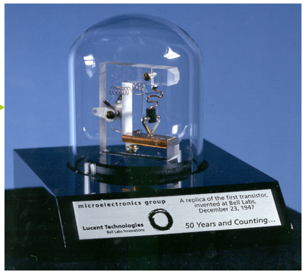
Regression

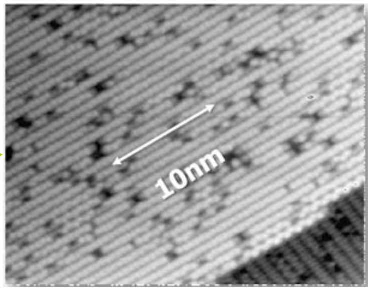
- Do regression in many dimensions
- Two steps
  - Voxels → semantic features (word synonyms)
  - Semantic features → word
- To classify
  - Take image, do regression to get semantic features
  - Then do it again to go from features to word
- Technique: *LASSO*

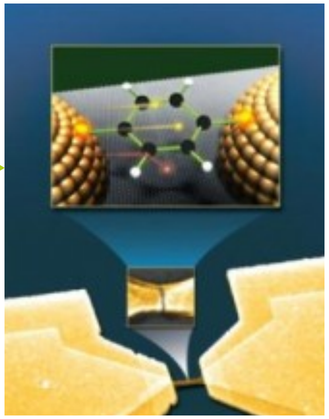
Programming Molecules

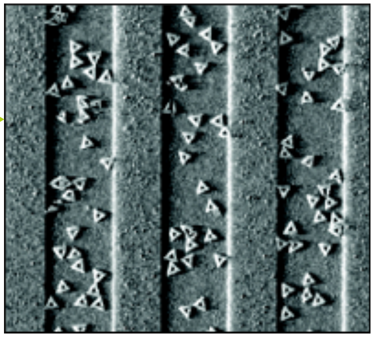
Getting Smaller

- First transistor
- 25nm NAND flash
- Single molecule transistor
- Molecules on a chip
- ~10 Moore's Law cycles left


microelectronics group
A replica of the first transistor, invented at Bell Labs, December 23, 1947
Lucent Technologies
50 Years and Counting...







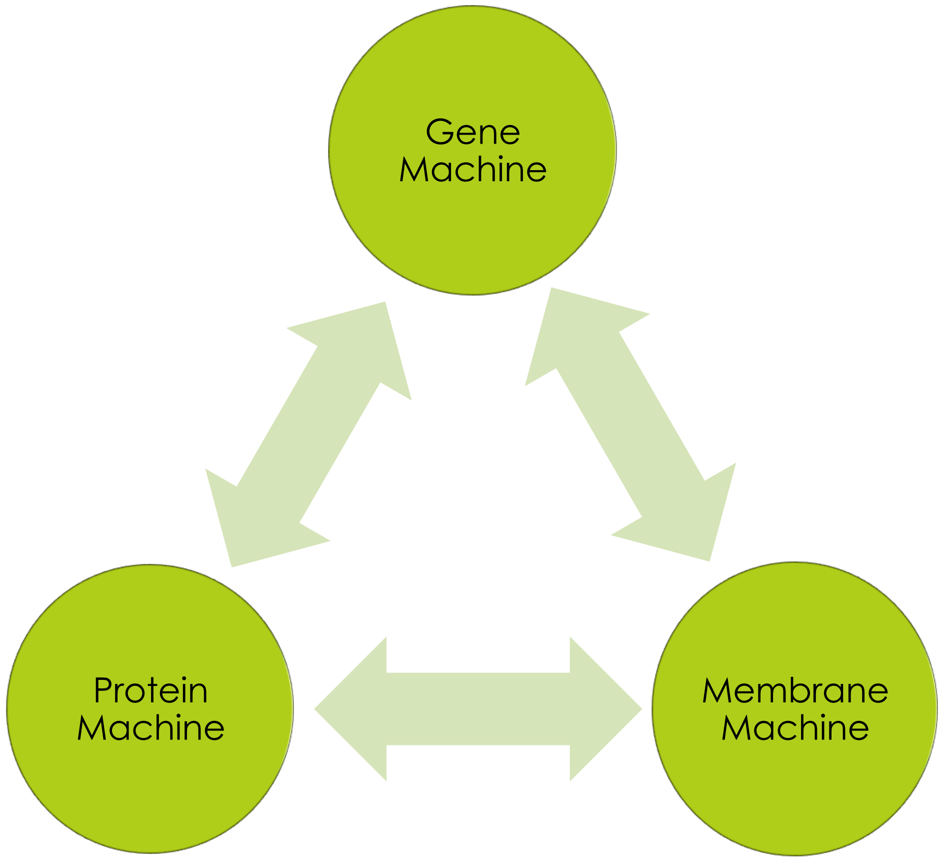
http://upload.wikimedia.org/wikipedia/commons/thumb/b/bf/Replica-of-first-transistor.jpg/200px-Replica-of-first-transistor.jpg
http://www.blogcdn.com/www.engadget.com/media/2010/01/01-30-10intelflash.jpg
http://www.wired.com/images_blogs/gadgetlab/2009/12/molecular-transistor-264x300.jpg
http://www.internetnews.com/img/2009/08/ibm_dna_chips.jpg

Building Smaller

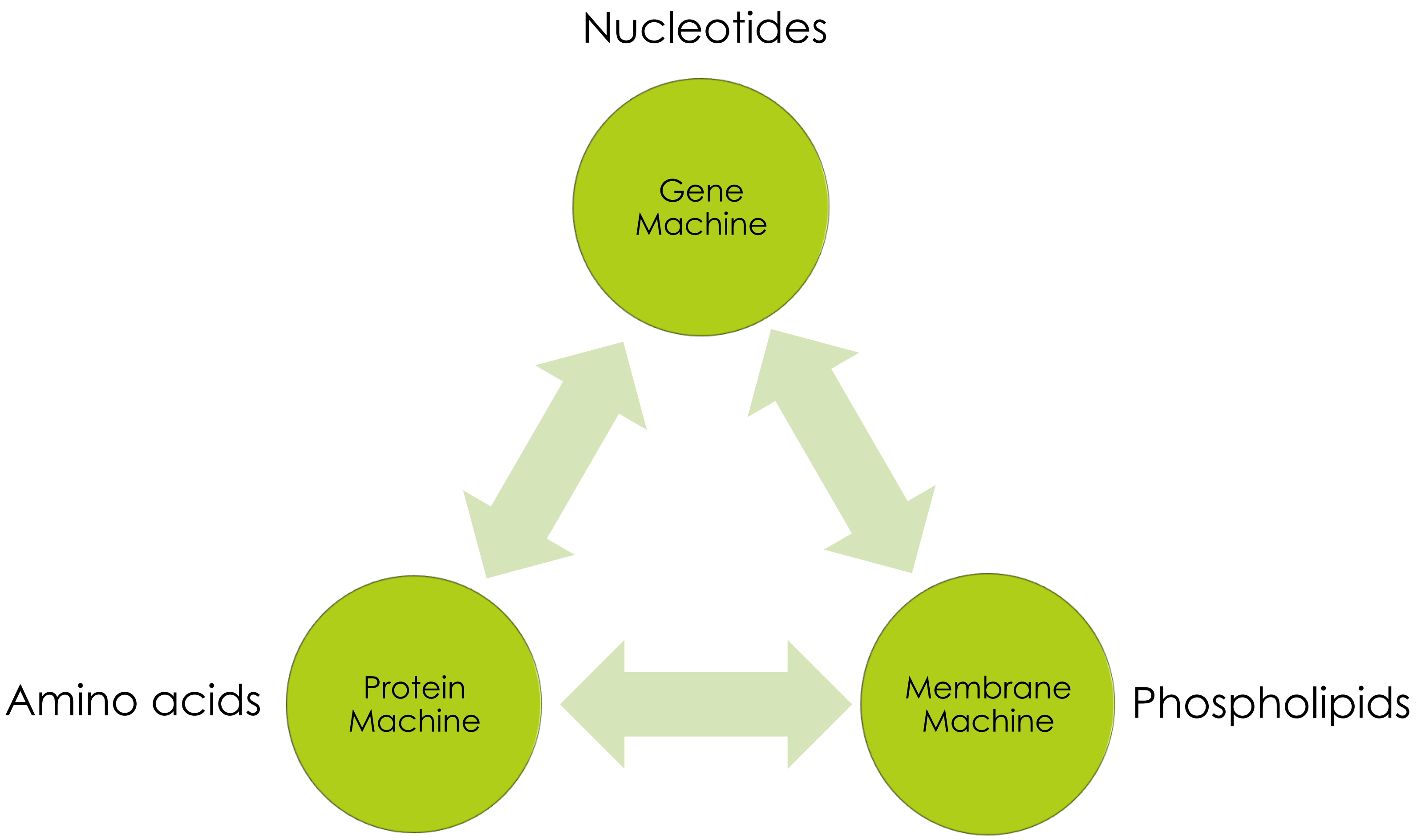
- How to build things smaller than your tools?
- You can't
  - Solution: self-assembly
- Molecular IKEA
  - Dear IKEA, please send me a chest of drawers that assembles itself.
- At a molecular scale, many such materials exist
  - Proteins, DNA/RNA, membranes
  - http://youtu.be/0N09BIEzDII

http://community.crystaltech.com/wp-content/uploads/2009/09/BigHammer.jpg

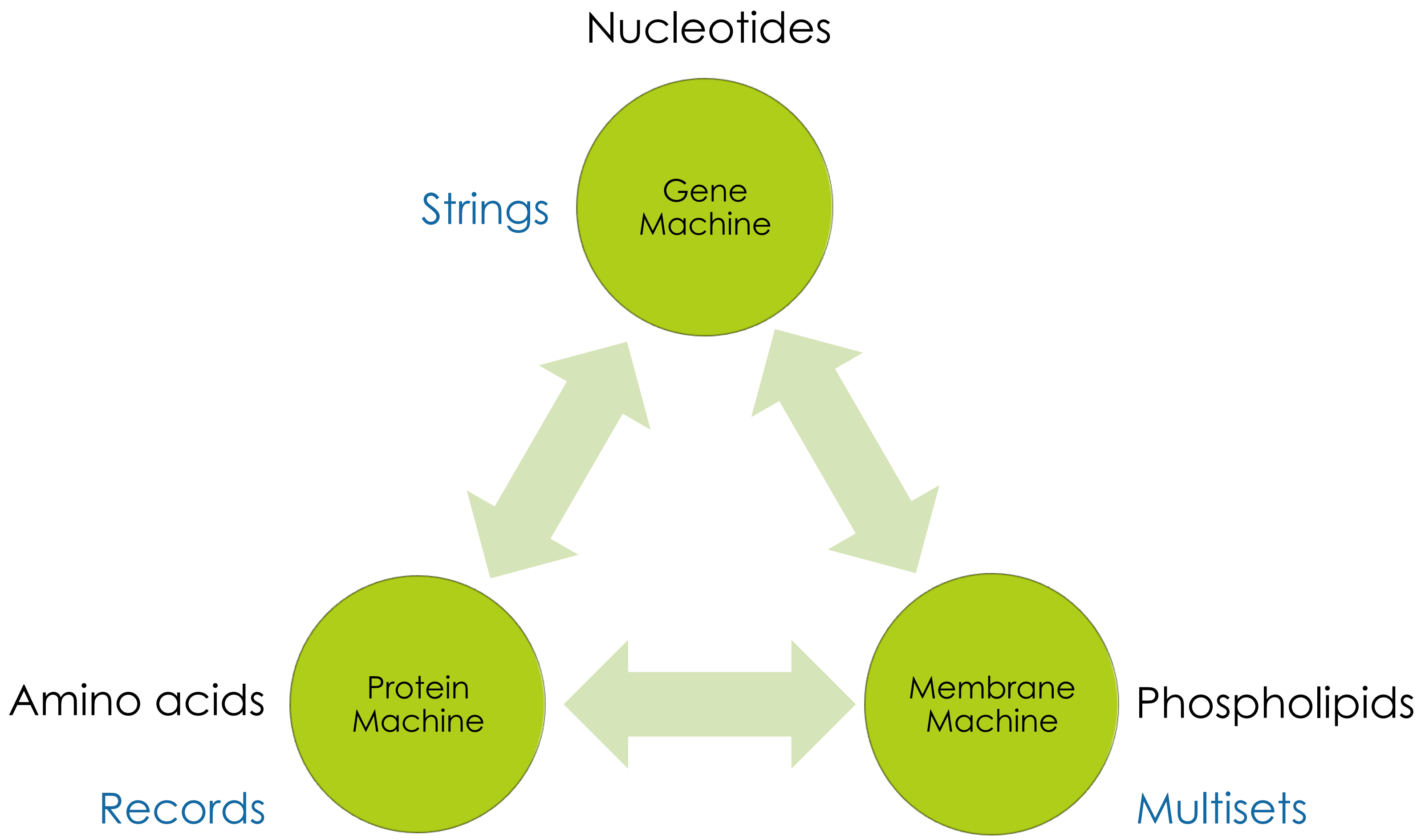
Machines in Biochemistry



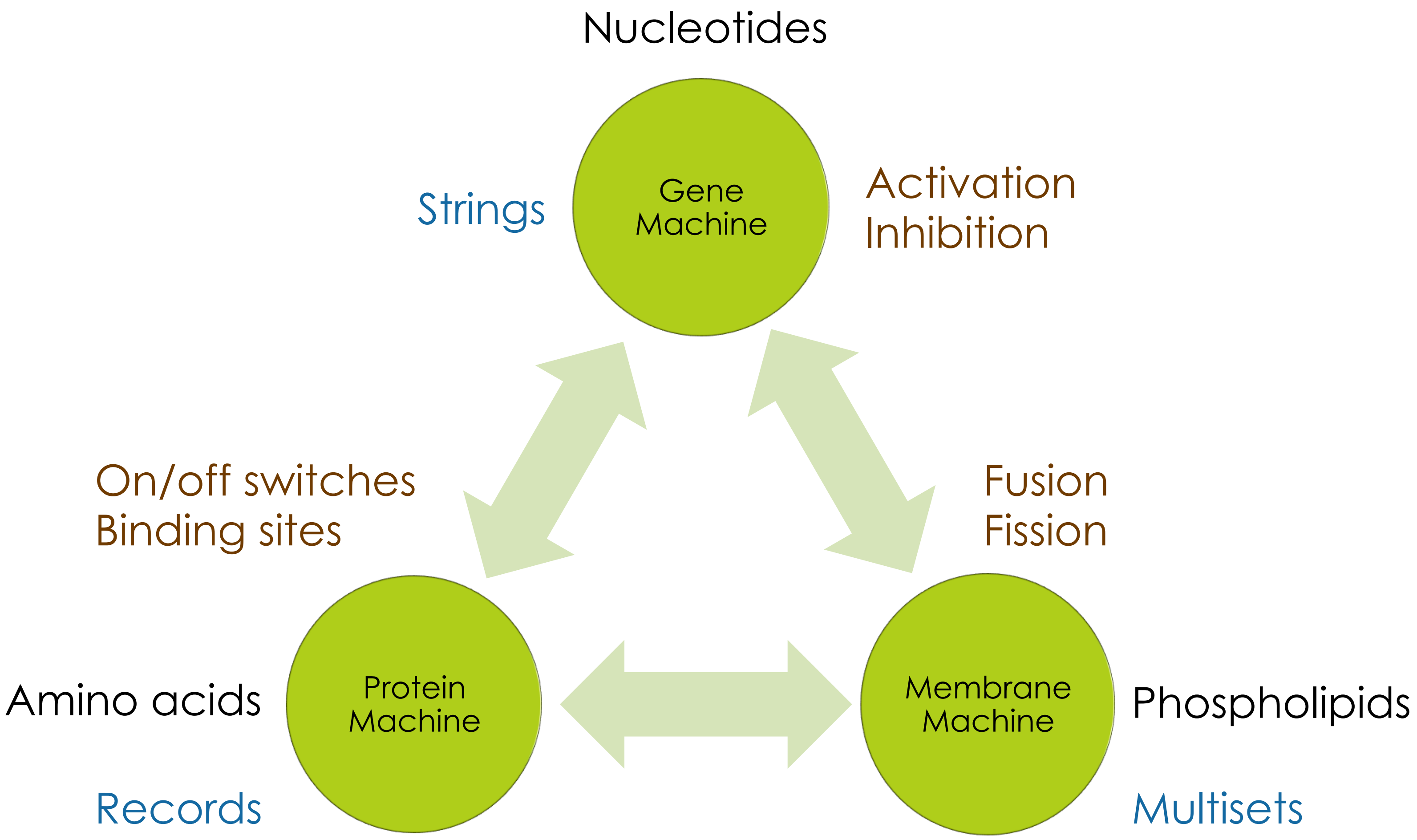
Machines in Biochemistry



Machines in Biochemistry



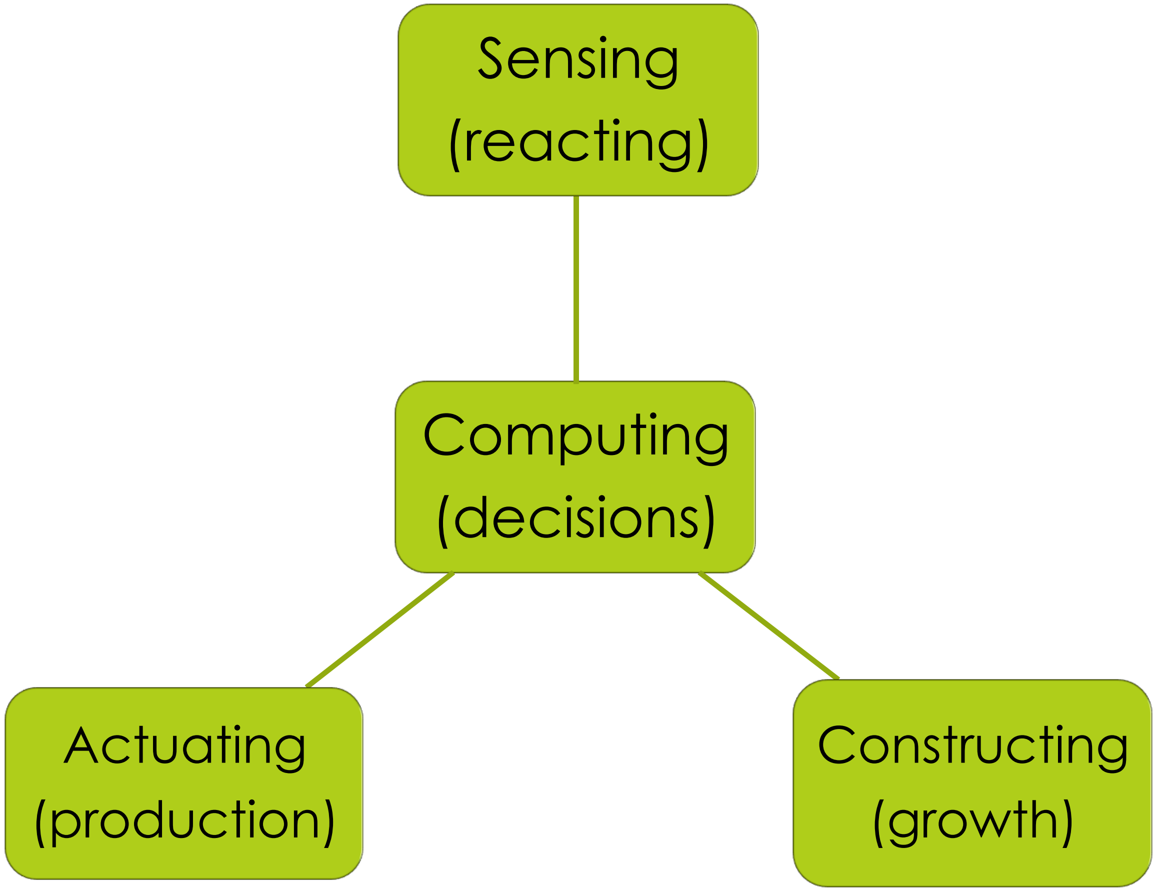
Machines in Biochemistry



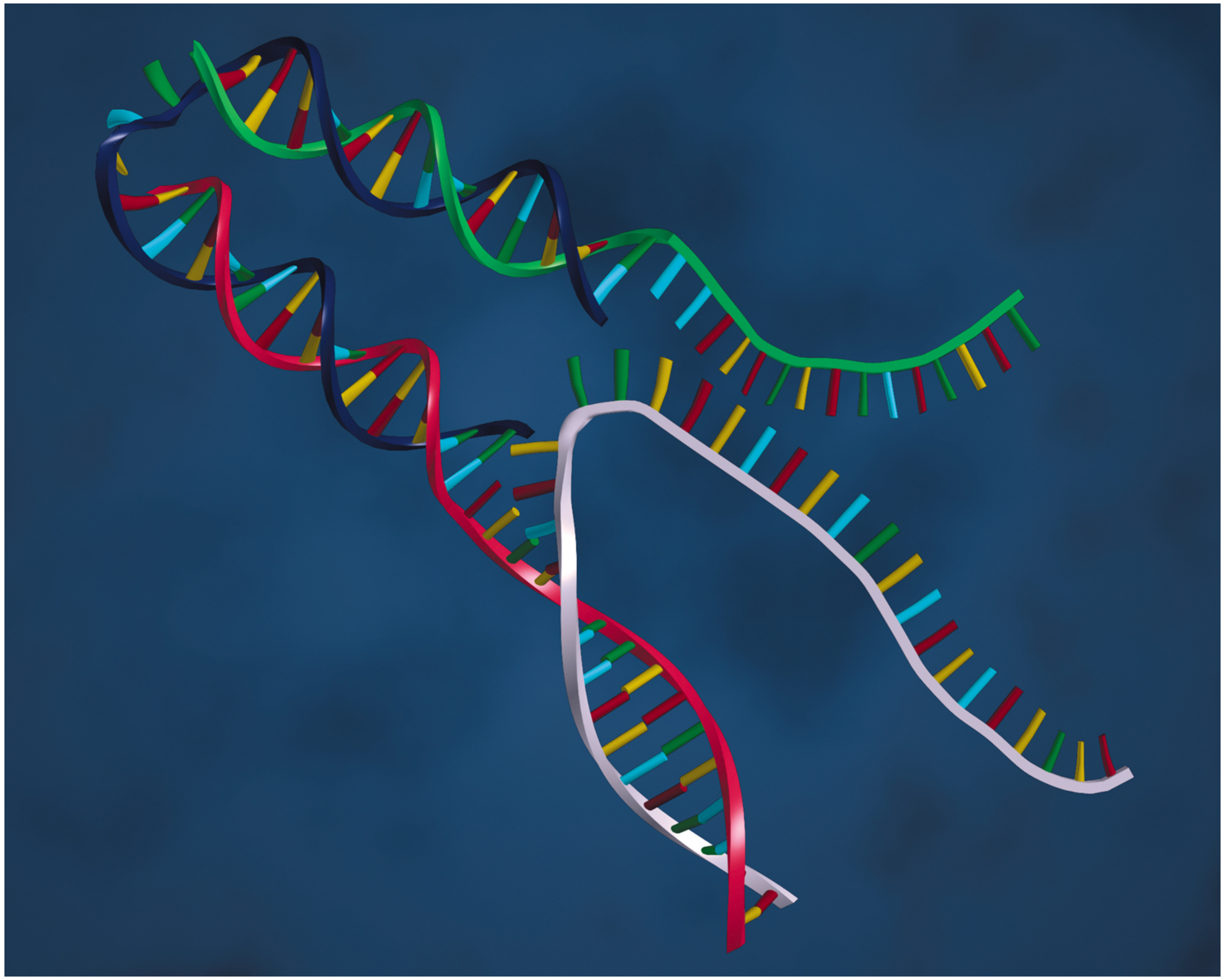
How do we form a “language”?

- Chemical reactions
  - $A + C \rightarrow_r B + D$
  - Instructions in a “program”
- Problem: combinatorial explosion
  - SO MANY chemical reactions in a cell
- Model reactions as automata – machines that perform a task
- Problem: chemistry is not an executable language
  - Dear Chemist, please execute this arbitrary reaction.

Controlling Systems on a Nanoscale



DNA Tweezers



http://www.cs.duke.edu/~reif/courses/complectures/AltModelsComp/MolecularRobotics/NonAutonomousDNAMotors/YurkeTurberfieldDNA.Tweezer/DNAseg2.jpg

Molecular programming workflow

- First figure out what gates you want to use and signals you want to send
- Signals + gates → structures of DNA
- Structures → sequences of DNA (NUPACK)
- Sequences → DNA synthesis (IDT)
- DNA synthesis → mail
- Receipt of DNA $\rightarrow_{\text{water}}$ execution
- Florescence is your "print" statement