Summary of bioinformatics algorithms training

Big O

Time

Memory

Theory vs reality

Deep Learning

Moving away from "understandable" algorithms

Requires lots data

Overfitting (vs overtraining)

Is the data diverse?

Can your model generalize

Genome annotation

Functional vs structural

Repeat masking (hard vs soft - lowercased)

ORF - gene

Gene finding

Just use evidence based (RNAseq)

Simple - start codon - known splice regions

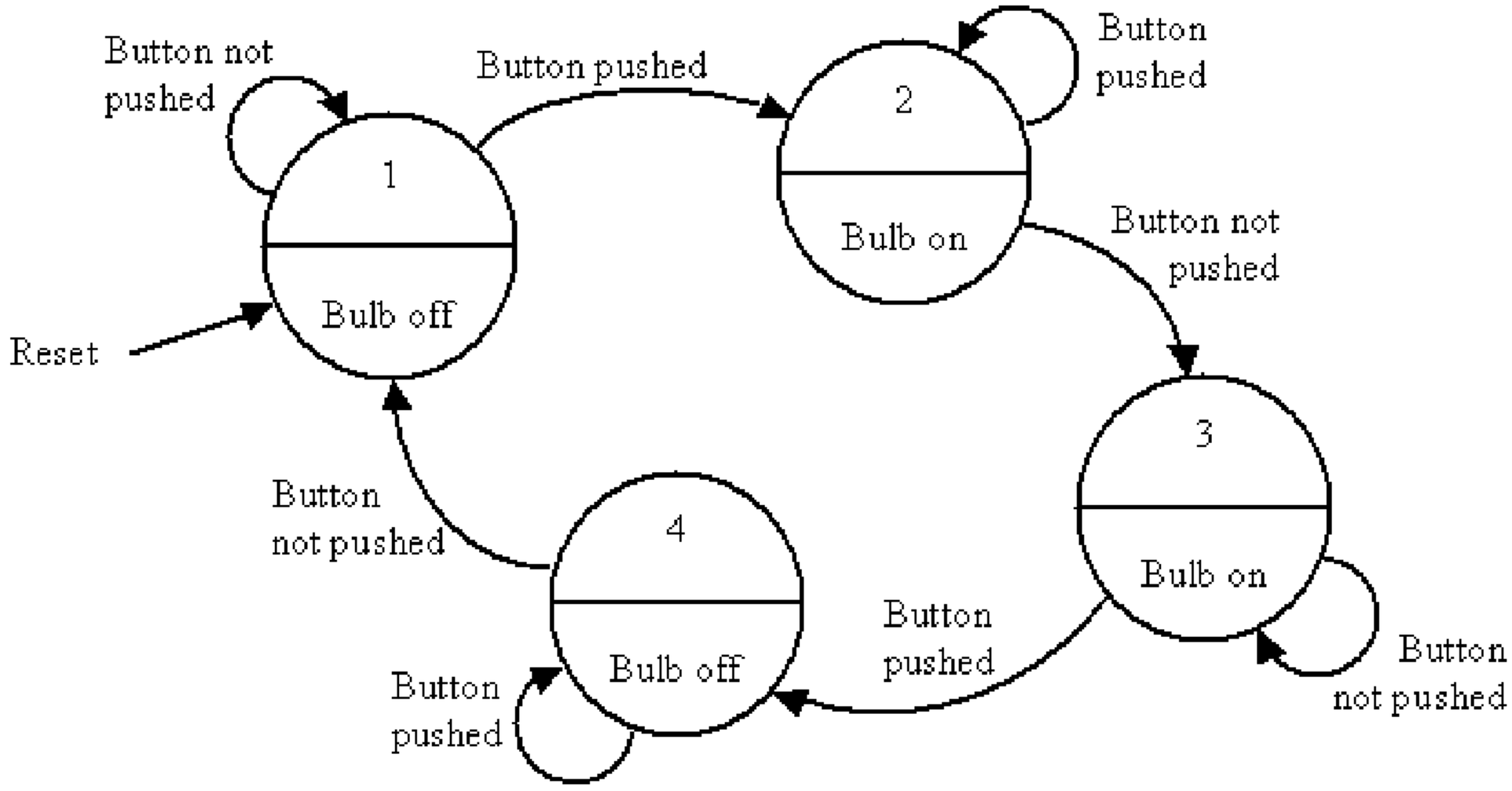
Trained

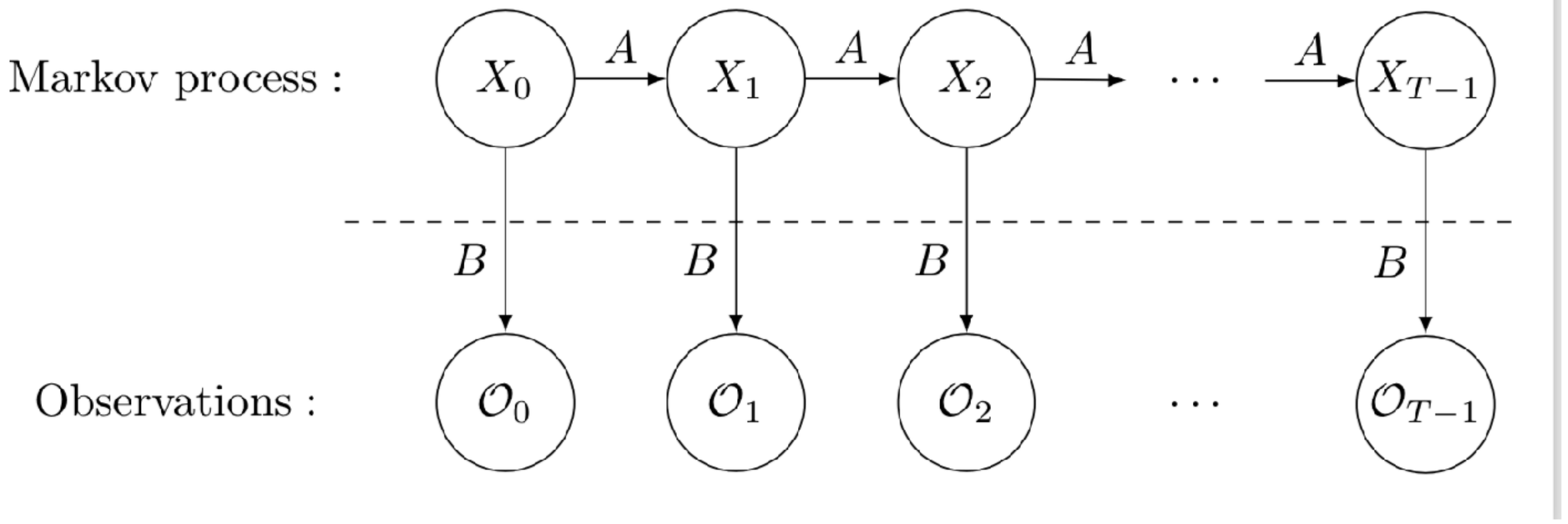
State diagram

From state to state with a probability

You can predict news in news papers (sort of)

First "L"LM - Claude Shannon (1947)


Button not pushed
Button pushed
Button pushed
1
2
Bulb off
Bulb on
Button not pushed
Reset
3
Bulb on
Button not pushed
4
Bulb off
Button pushed
Button pushed
Button pushed


Markov process :
X_0
A
X_1
A
X_2
A
$\cdots$
A
X_{T-1}
B
B
B
B
Observations :
$\mathcal{O}_0$
$\mathcal{O}_1$
$\mathcal{O}_2$
$\cdots$
$\mathcal{O}_{T-1}$

HMM - Markov (1913)

There is a hidden state

There are visible states given emission probabilities

Guess hidden states

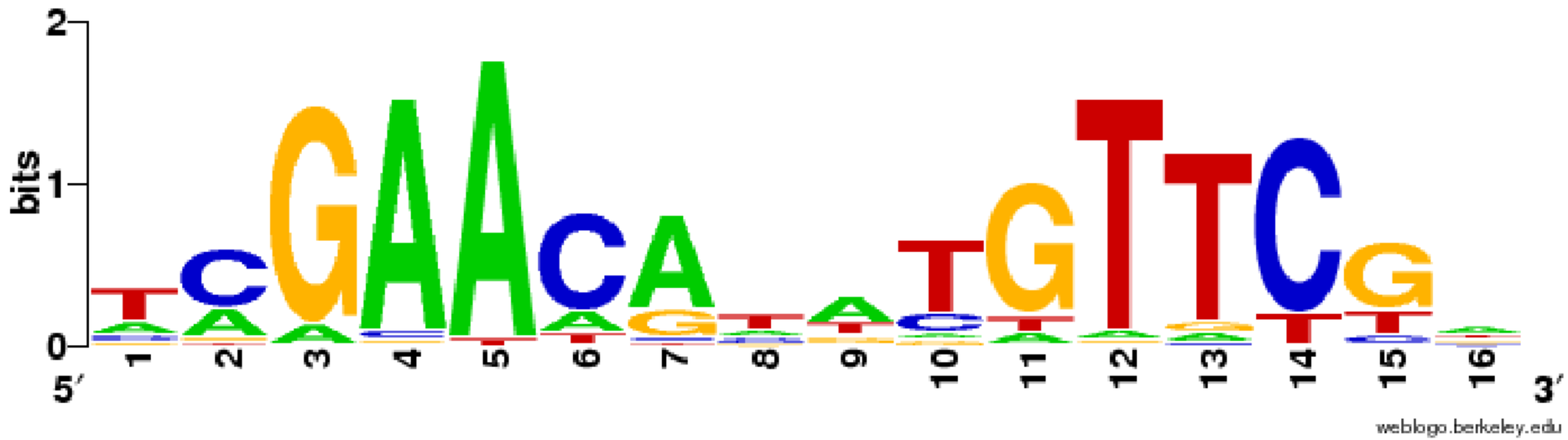
1. Train HMM

2. Solve with Viterbi

This is how gene models work

Exomedepth works similar

HMM can be a layer in a DL architecture


bits
2
1
0
5′
3′
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
weblogo.berkeley.edu

Kmers

Divide a sequence in l - k segments

kmer distribution - uniqueness indicates error in read

kmer graphs

kmer minimizer - select representatives

somewhat smoothing operator to improve computation

metagenomics

look up what k to use

Mapping

read to genome

Seeding:

BLAST - old heuristics based

BWA: FM index for fast seeding (short reads)

minimizers: minimap2 for long reads

More alignment

global vs local - Needleman Wunsch vs Smith Waterman

WFA the newest global alignment algorithm

many low-level optimization possible

Genome Assembly

Very hard problem to do well

Errors

Gaps

Repeats

Copies (CNV)

de novo or with reference (bias)

phasing ploidity

DBG vs OLC

1. Map 2. Graph 3. Trim 4. Profit

DBG: NGS - kmerization (eulerian)

OLC: long reads - Old is new again (hamiltonian)

Bioinformatics love PacBio Hifi reads for this

Additional information

Contact information

HiC

PoreC)

Tools

Canu - big old tool

Spades - and all its variants

Flye - very easy to handle

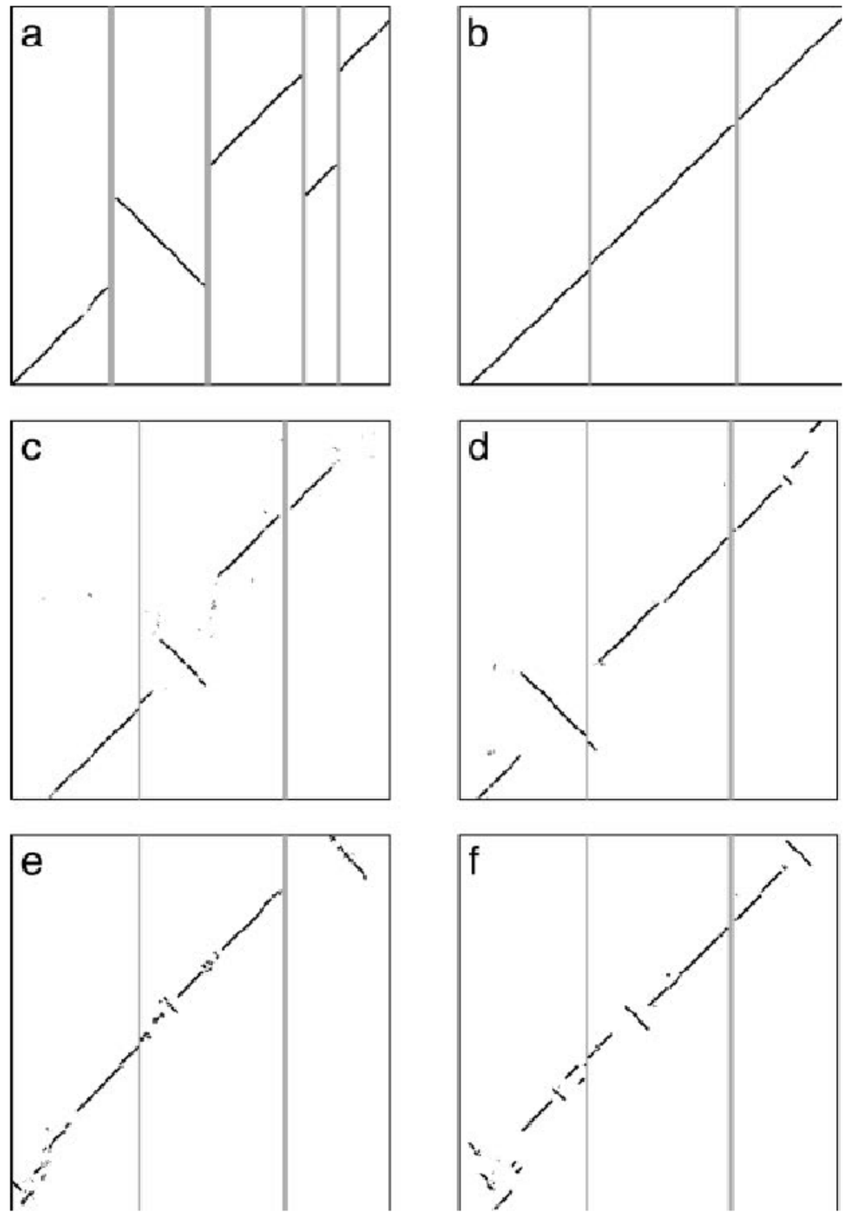
miniasm/hifiasm - if you want something fast

Measuring assembly performance

Contigs

N50/N90

Dot plots


a
b
c
d
e
f

Pangenomes

Reducing reference bias

Adding SV information to the reference

Very new idea

Finally: Heng Li

Samtools, seqtk, gfatools, bwa, bioawk, minimap2, miniasm, minigraph

All basically solo projects in extremely succint C code

Simple and powerfull - no dependencies

Functions can often be copy pasted to your project

This is what other expert bioinformatics actually do

Wrapping up - ...

Bioinformatics is a tool world - not a library world

Lots of tool wrapping as well

Advantages:

Simple APIs (output files)

Tools in a pipeline can be replaced

Disadvantages:

Tools do not provide their full picture

(De-)serialization

Where this is going:

Tools become pipelines (Cellranger)

Integrate end-to-end DL

Experts in niche fields need to write their own tools