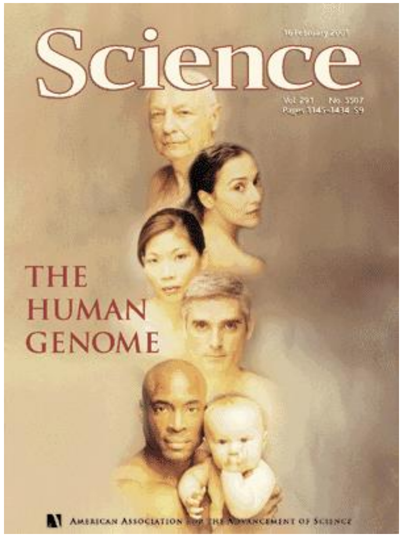
Genomes and Assembly

- Ask your elderly aunt what is bioinformatics ?
- Read the Hamburg Abendblatt
    - genomes
- June 2000 – human genome sort of finished
- Feb 2001 – publication of human genome



The Plan

- what does one really know ? where are the problems ?
- assembly
- problems
  - technical
  - repeats
  - coverage
- high and low quality

What does one really know ?

Interpretation – diseases with hereditary component

- what is the gene for multiple sclerosis ? type 2 diabetes ? blood pressure ?
- there is no answer (many years after first genome)

What do genes do ?

- dead genes, pseudo genes, regulatory sites, ..

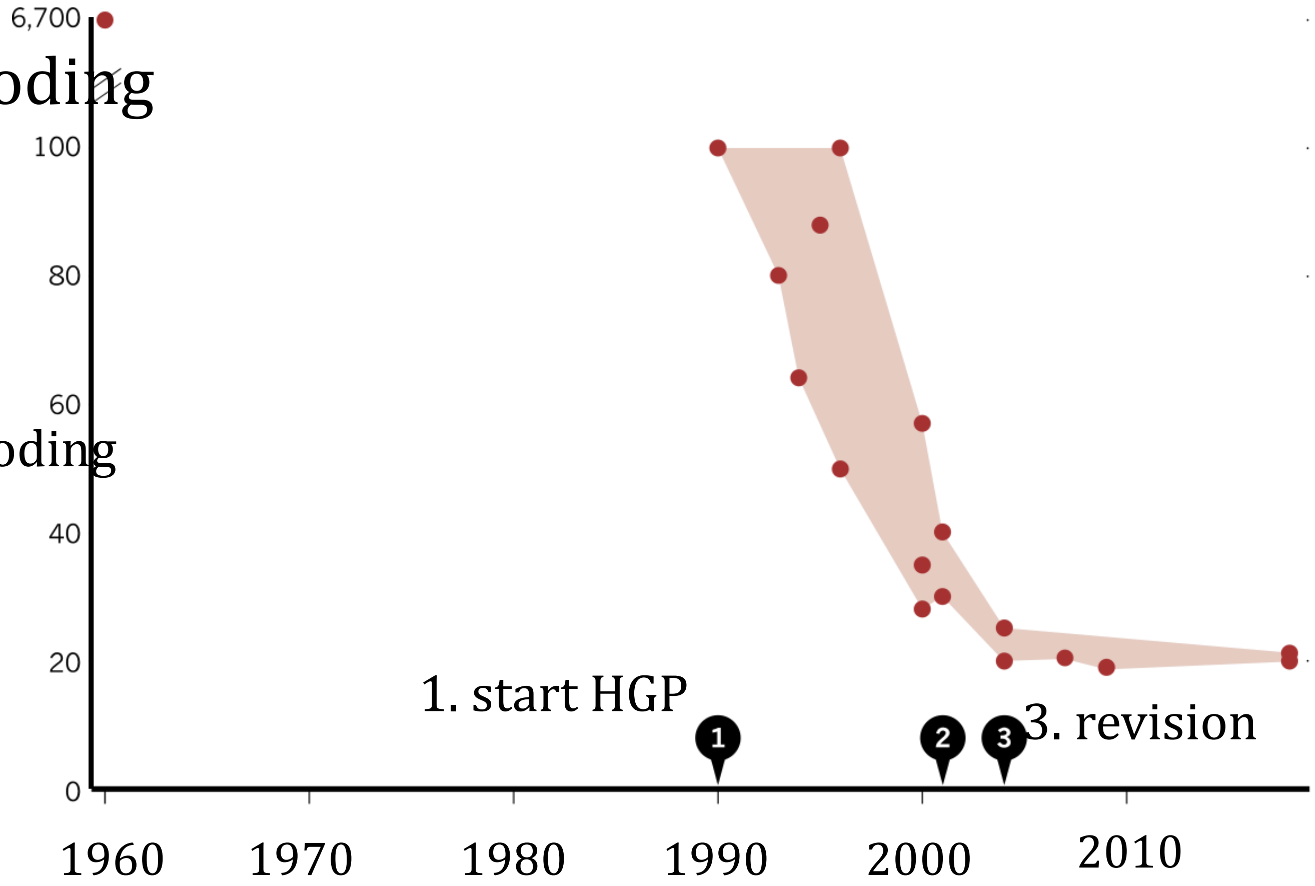
nature
THE HUMAN GENOME AT TEN
Growing pains of the genomics age
THE FAINT YOUNG SUN
A climate paradox revisited
QUANTUM MECHANICS
Controlling objects you can see
SLEEPING SICKNESS
Drug target for a neglected disease

- Easy questions ?

How many genes do I have ?

How many genes ?

recent estimate
21 306 protein coding genes



n protein-coding genes / 10^3

1. start HGP

3. revision

2. genome published

Willyard, C. Nature, 558, 354-355 (2018)

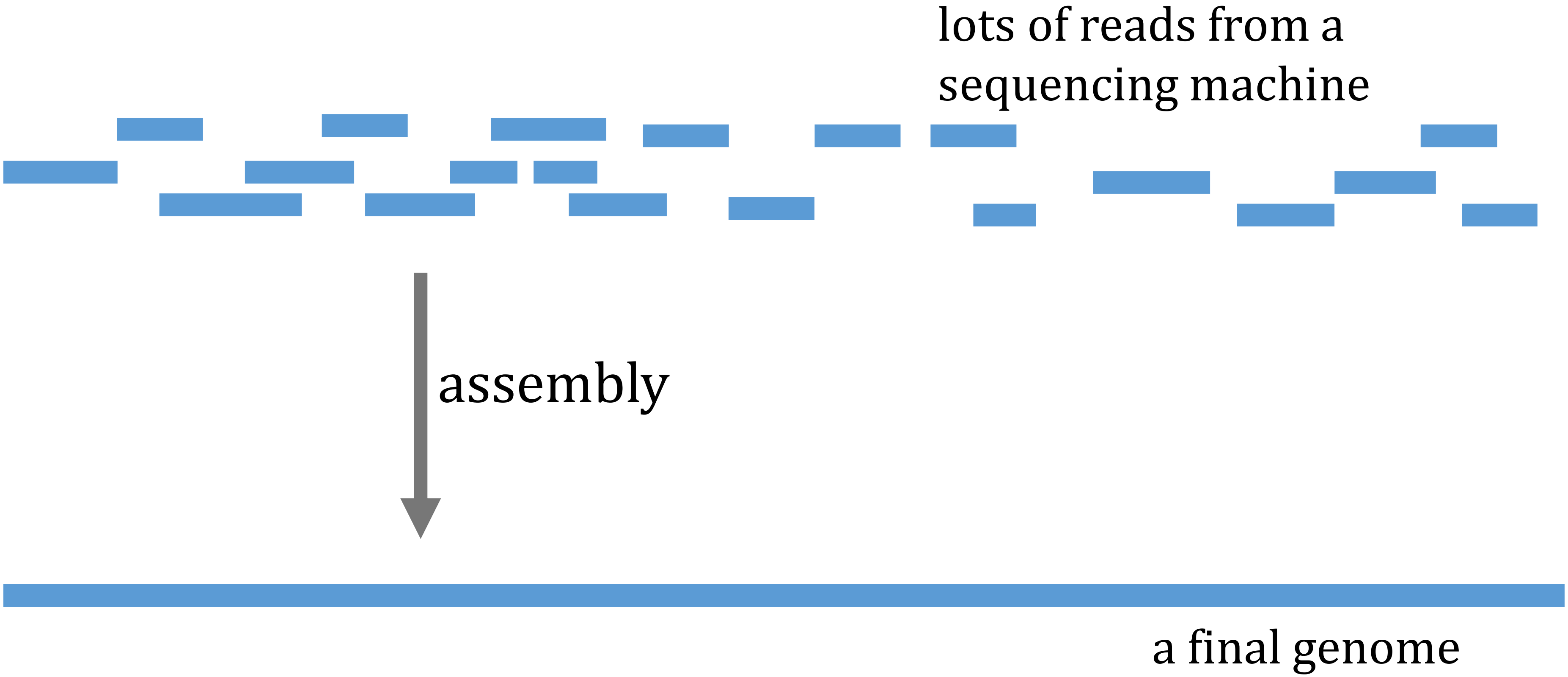
Genome assembly

- Genome is big
- split into pieces (enzymes, mechanical, …)
- read and sequence pieces
- put these unordered fragments together

genome assembly problem

What will happen ? Most of what you imagine

- some parts – no overlap, some much overlap
- overlap with reads from different parts of genome…



typical numbers

Computational problem ?

- genome size
- read size (10^2 – 10^5)

Just use a method that gives us 10^5 bases at a time ?

	bases
viroids	200 – 300
virus	10^4 – 10^6
prokaryotes	10^5 – 10^7
eukaryotes	10^5 – 10^{10}

Read lengths / tradeoffs

Why do we not just use long read methods ?

- errors – error rate much higher on long reads
- cost to start (cost of a machine / investment)
- cost per base
- speed / bases per day

Now consider the assembly problem

- how big is the computational problem ?

how big ?

Original human genome (10 years)

- 3×10^7 reads of <800 base pairs

Yeast

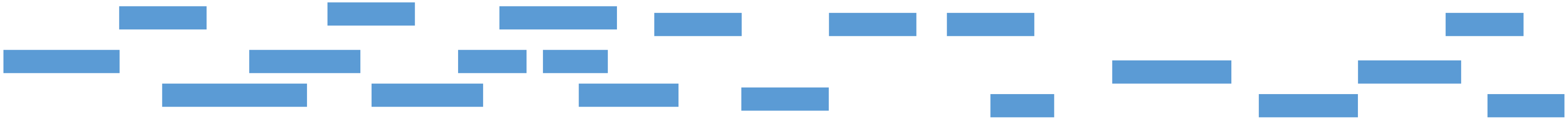
- 10^7 reads

Newer human genome

- 10^8 reads (shorter)

- What if I have a step that needs to compare all fragments with all ? $10^8 \times 10^8 = 10^{16}$
- where would you start ? is the problem like a multiple sequence alignment ?

Intuitive approach to assembly (bad)



Treat like a multiple sequence alignment

- compare all against all
- find closest sequences
  - align first
- align groups of sequences with each other
- look at each column and just read the consensus

```
          ACG
                CGATC
reads
                  ATCGCTT
            GATTCGA
consensus ACGATTCGATCGCTT
```

Not the right approach

- not practical – not error tolerant, gaps not wanted

Multiple Sequence alignment – why not to

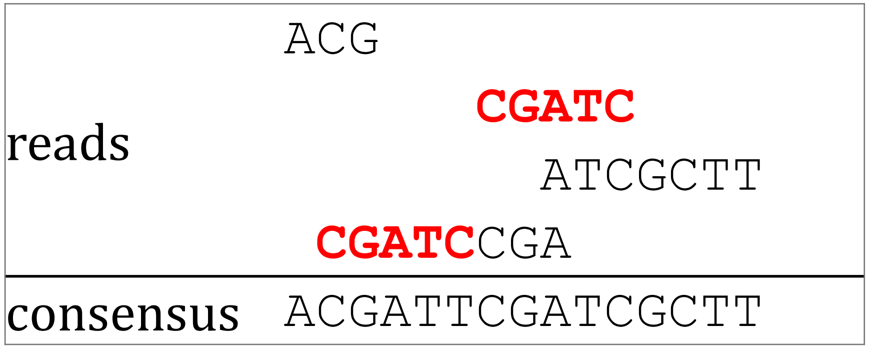
multiple sequence alignment – basic idea

- all sequences are a bit different
  - all sequences are variations on same region
  - mostly similar matches

genome assembly

- fragments of one long sequence
- difference are errors / polymorphisms
- want exact matches

Not multiple sequence alignment



```
          ACG
                CGATC
reads             ATCGCTT
           CGATCCGA
consensus ACGATTCGATCGCTT
```

What might happen ?

- **CGATC** might align with **CGATC**CGA

Multiple sequence alignments allow for gaps

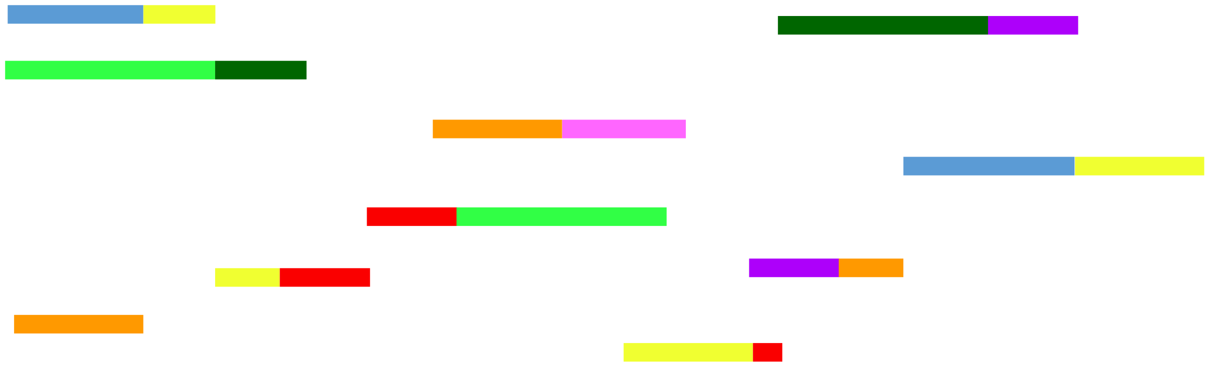
- not wanted here

Speed: 100 000 reads would need $\frac{10^{10}}{2}$ alignments

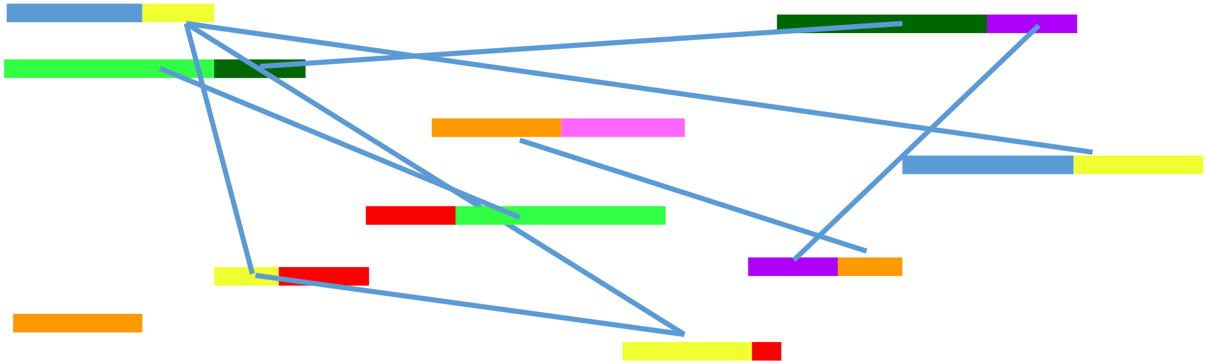
Consider a different philosophy

overlap layout consensus

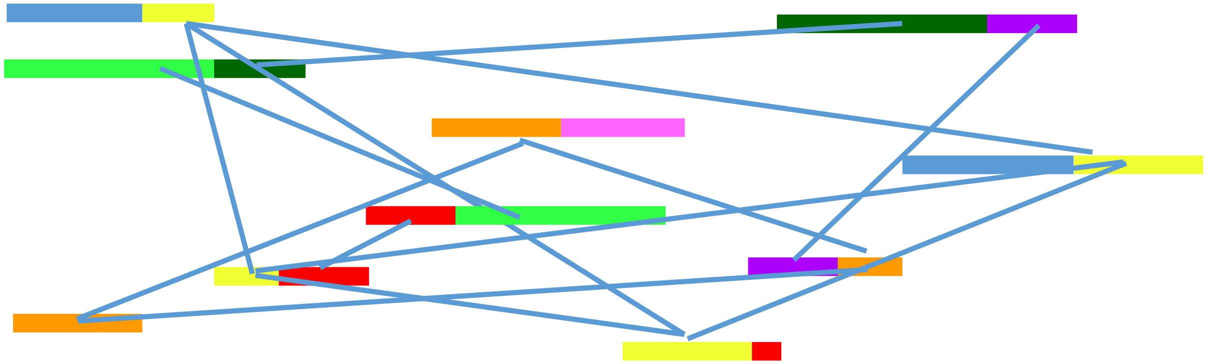
Two methods coming – this is the easier



find overlap – first few edges



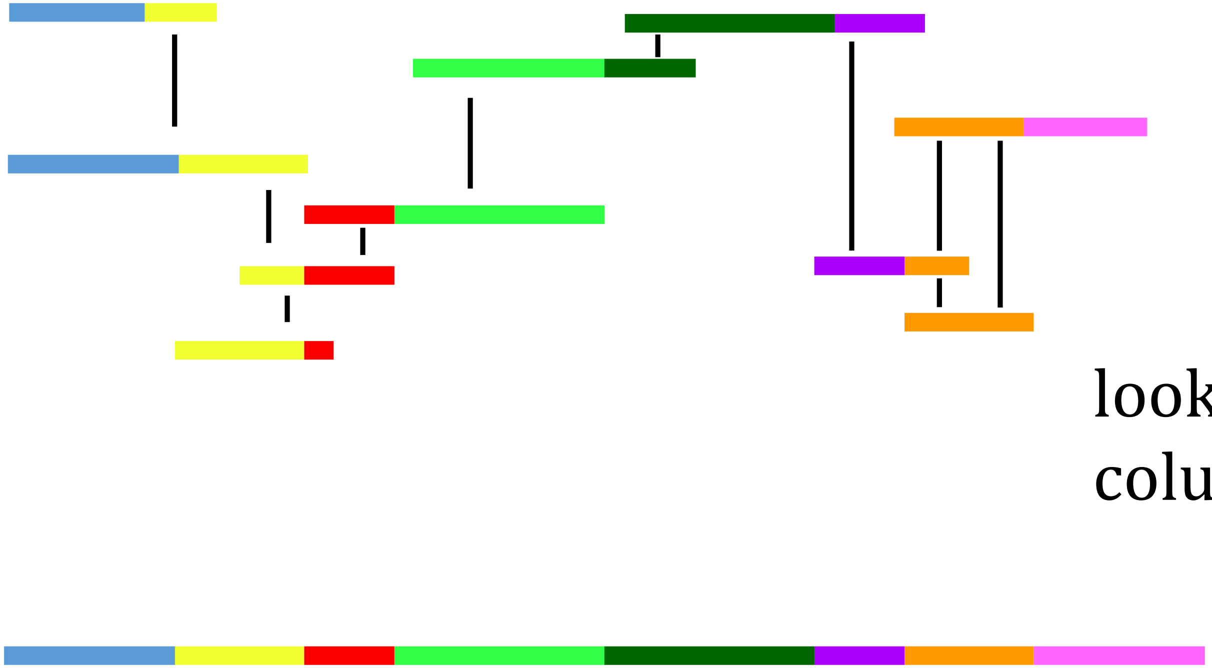
find overlap – most edges



push aligned fragments on top of each other

hopefully all fragments are connected

layout and consensus



look down each column for consensus

Does it always work ? more later

de Bruijn graphs and *k*-mers

2nd method

Remember blast ?

- fast because it looks for identical matches (seeds)
- use fast lookups

- use this idea of quick searches for identical pieces

break into *k*-mers

ich mag fisch

```
ich_m
 ch_ma
  h_mag
   _mag_
    mag_f
     ag_fi
      g_fis
       _fisc
        fisch
```

put them together…

the *k*-mers

```
fisch
ch_ma
ag_fi
_mag_
h_mag
g_fis
ich_m
_fisc
mag_f
```

prefixes length $k - 1$

fisch

ch ma

ag fi

 mag

h mag

g fis

ich m

 fisc

mag f

find suffixes

fisch

ag_fi

ch_ma

mag

h_mag

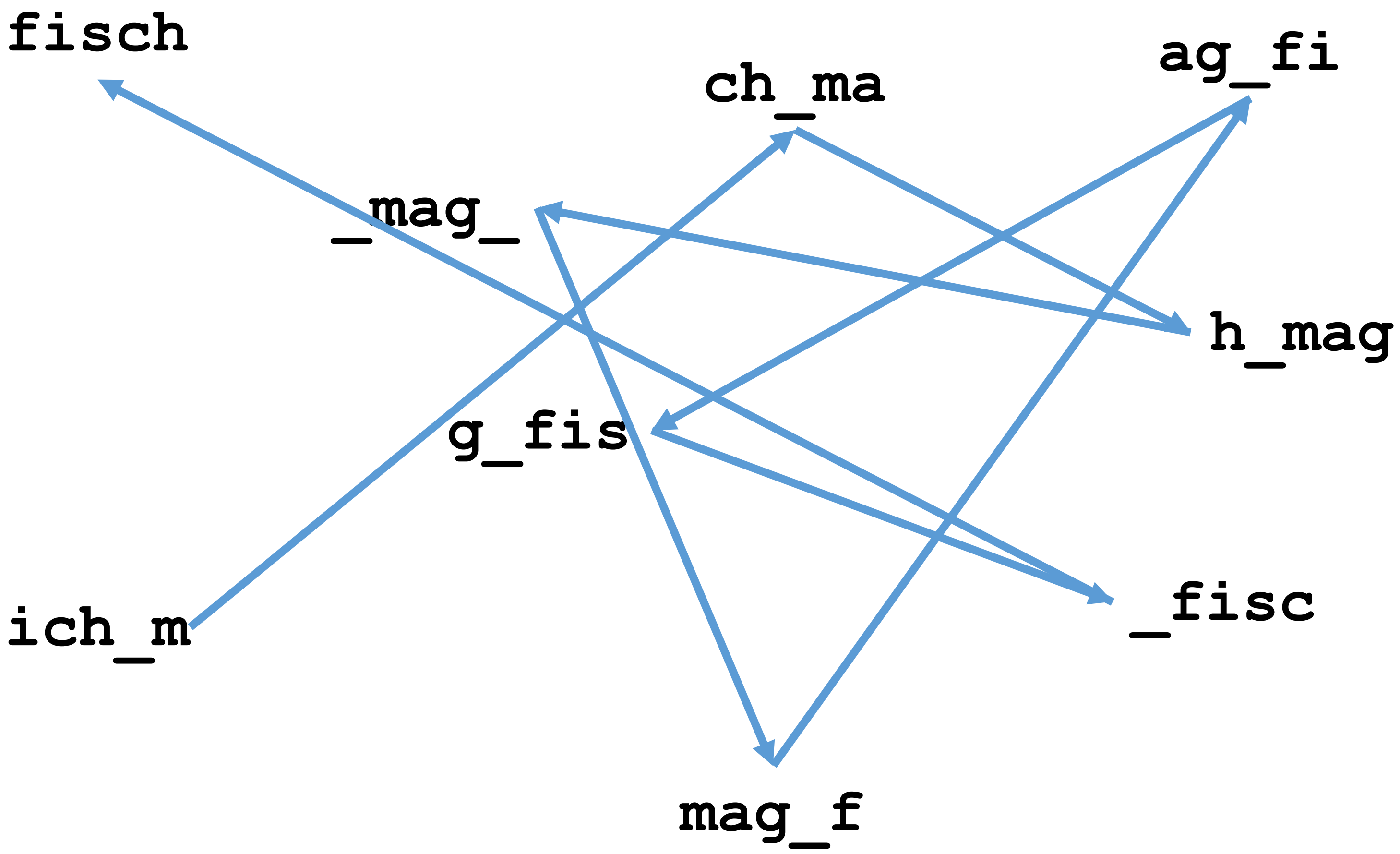
g_fis

_fisc

ich_m

mag_f

join suffixes and prefixes

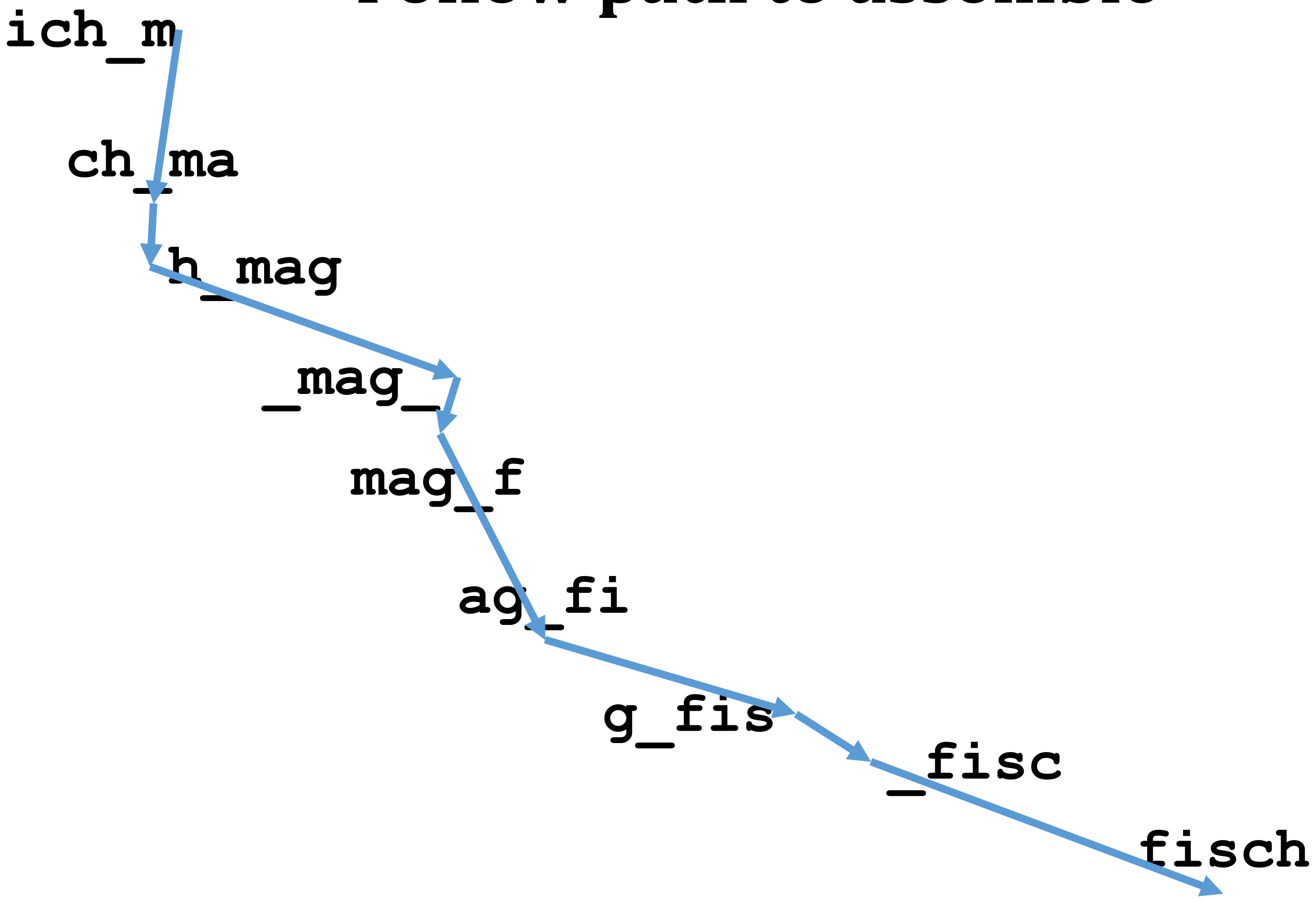


find a node with zero inputs **ich_m**

- start

find a node with no outputs – must be end **fisch**

Follow path to assemble



ich_m

ch_ma

h_mag

mag

mag_f

ag_fi

g_fis

_fisc

fisch

Is this boring ? Why is it faster ?

- original (consensus overlap) had O(n^2) comparisons
- Here we can use some tricks
- Build a table with all the k-mers you know

- run over reads once and mark k-mers with fragments

- more complicated example

$$\left[\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ \ldots \\ 100 \\ 101 \end{array}\right] \begin{array}{c} \texttt{ich\_} \\ \texttt{ch\_m} \\ \texttt{other} \\ \texttt{...} \\ \texttt{schi} \\ \texttt{fisc} \\ \texttt{isch} \end{array}$$

```
                      aal_und_brot_und_ei
aal_u
 al_un
  l_und
   _und_
    und_b
     nd_br
      d_bro
       _brot
        brot_
         rot_u
          ot_un
           t_und
            _und_
             und_e
              nd_ei
```

the fragments

rot_u

brot_

nd_br

und

l_und

d_bro

t_und

ot_un

und_e

und_b

_brot

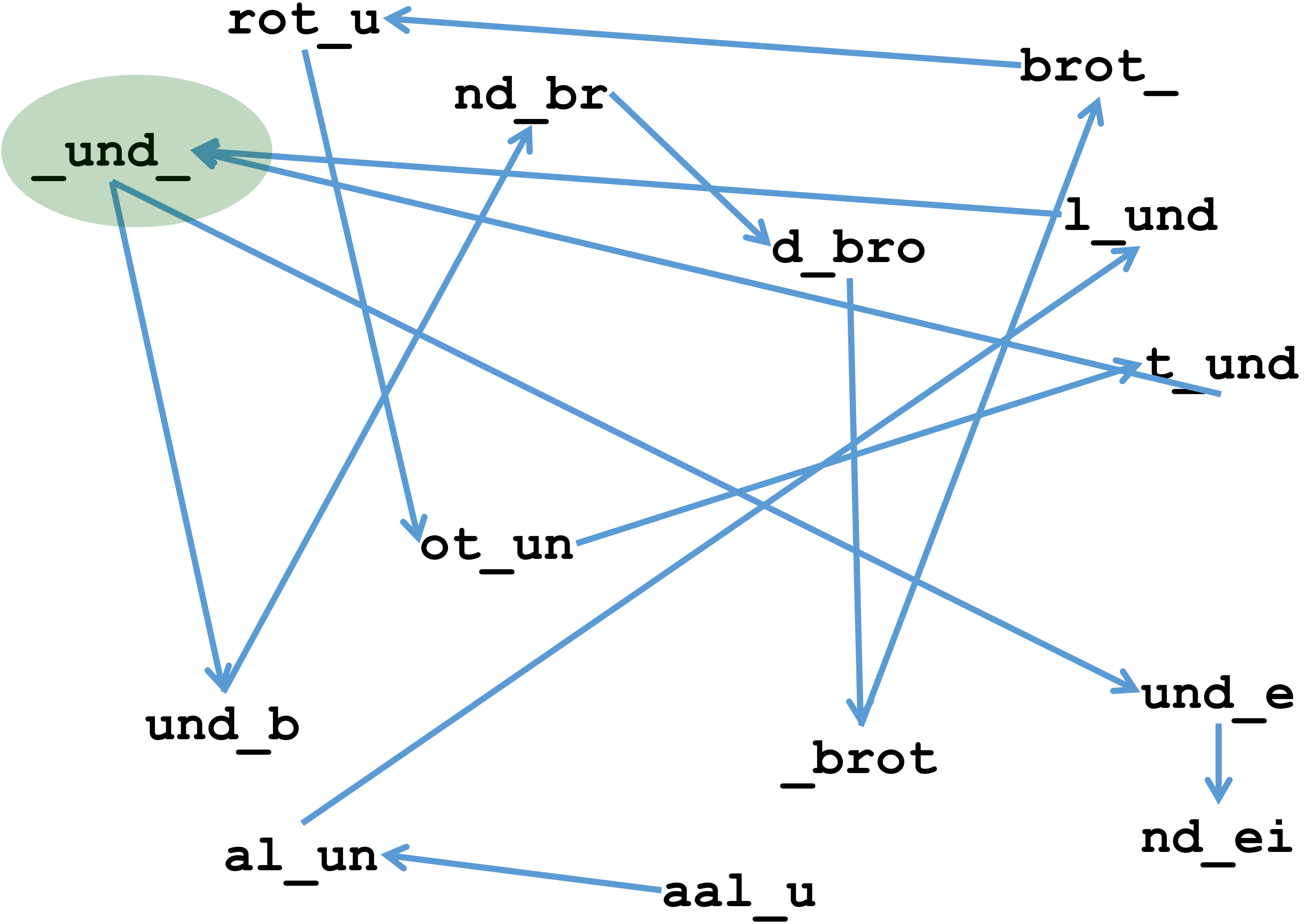
nd_ei

al_un

aal_u

aal und brot und ei

aal_und_brot_und_ei



rot_u

brot_

nd_br

und

l_und

d_bro

t_und

ot_un

und_e

und_b

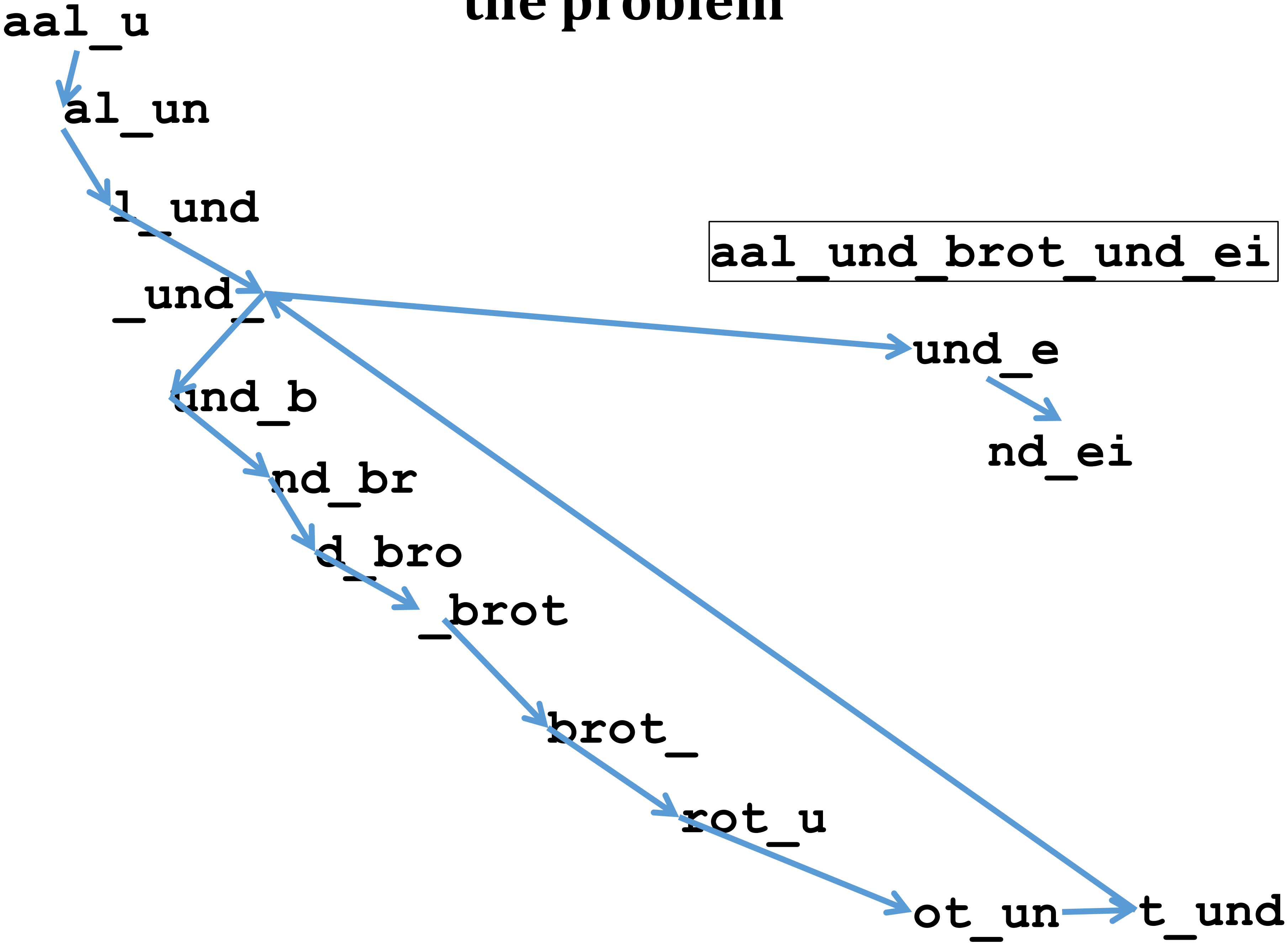
_brot

nd_ei

al_un

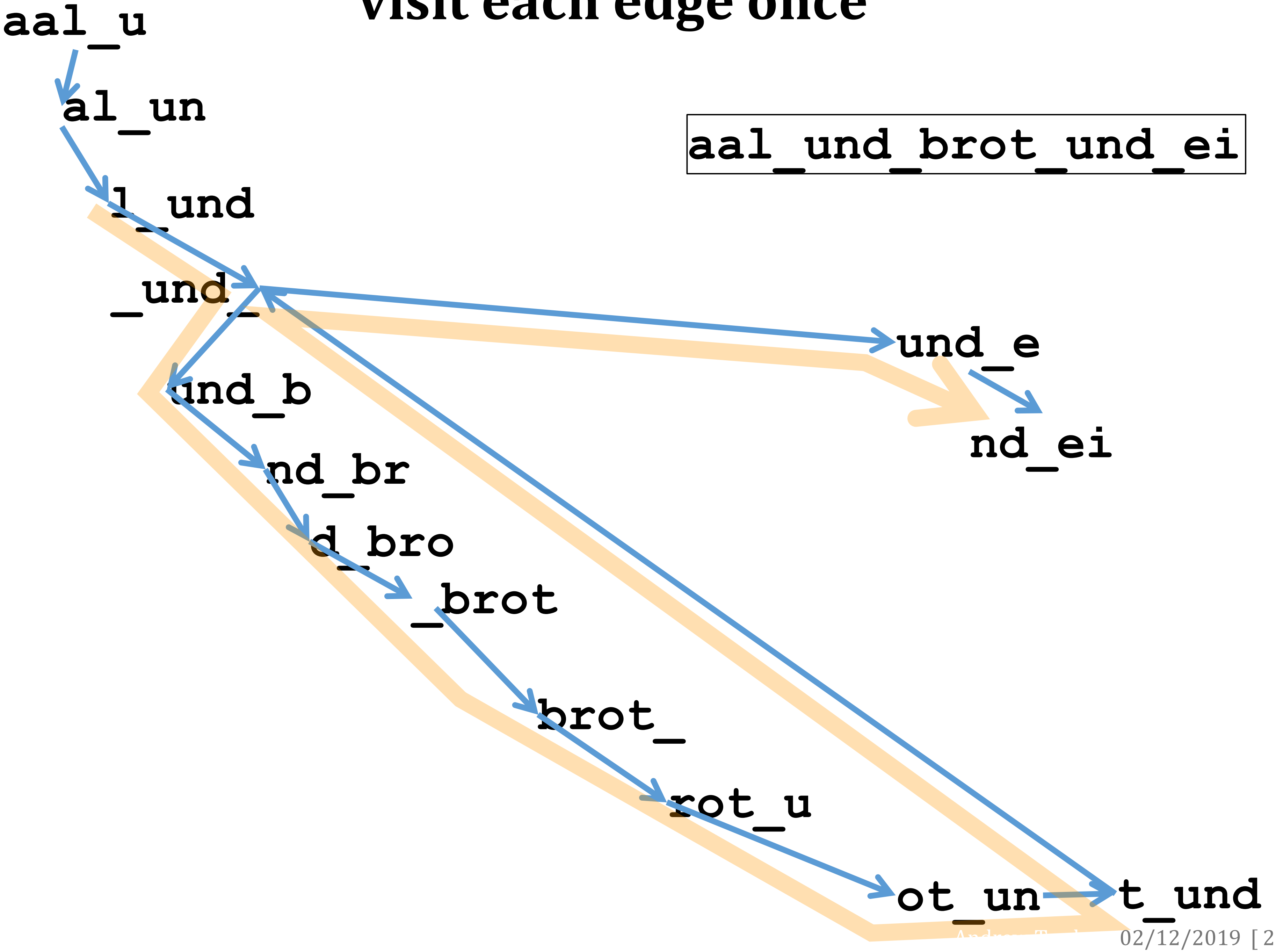
aal_u

the problem


aal_u
al_un
l_und
und
und_b
nd_br
d_bro
_brot
brot_
rot_u
ot_un
t_und
und_e
nd_ei
aal_und_brot_und_ei

visit each edge once

aal_und_brot_und_ei

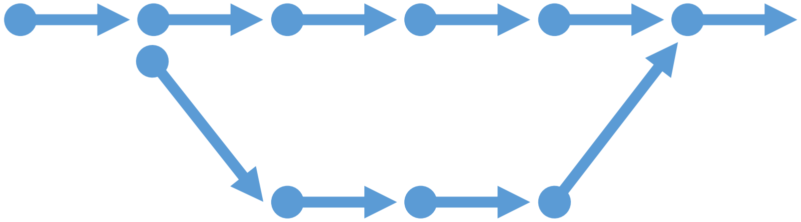


more steps to a practical version

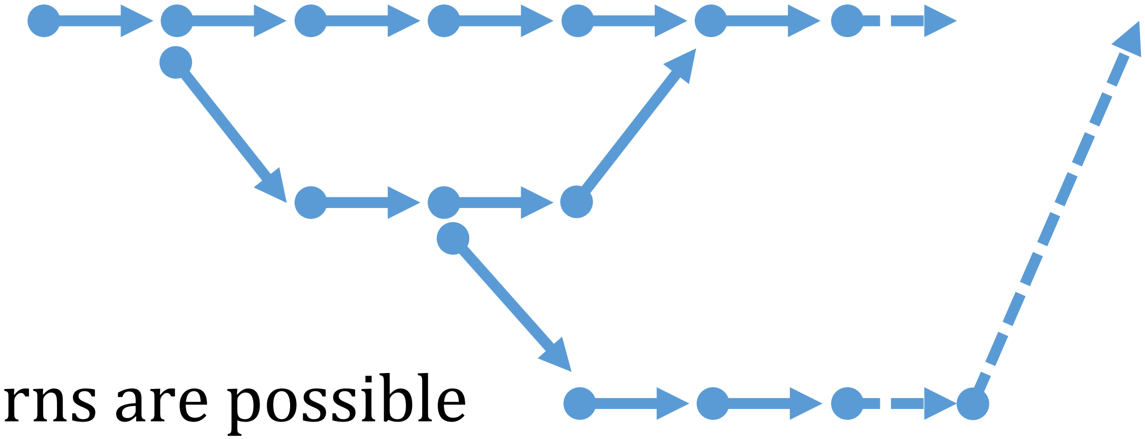
- merging multiple reads
- errors
- repeats
- missing pieces assembling

overlaps / merging

- lots of overlaps of different regions



- a good sequencing might be 30 or 100×
- these can be merged



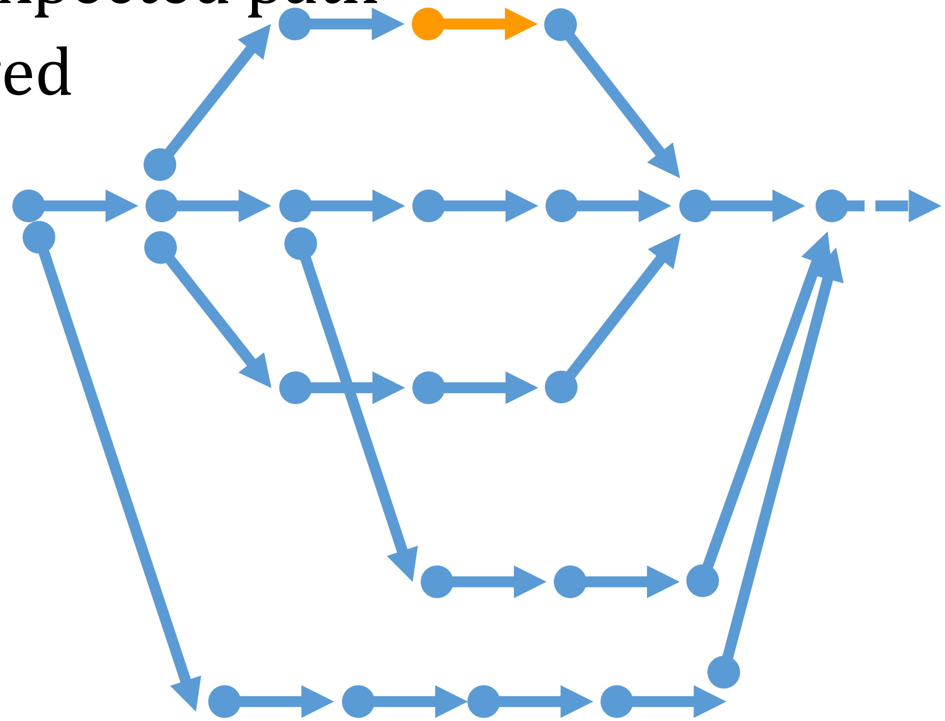
- all kinds of patterns are possible

errors

If we have enough coverage there are many paths over each fragment

Random errors give you an unexpected path

- bottom paths can all be merged



If many paths agree the orange one is an error

Discuss error sources later

k-mer length

- only relevant to de Bruijn method
- how long is ?

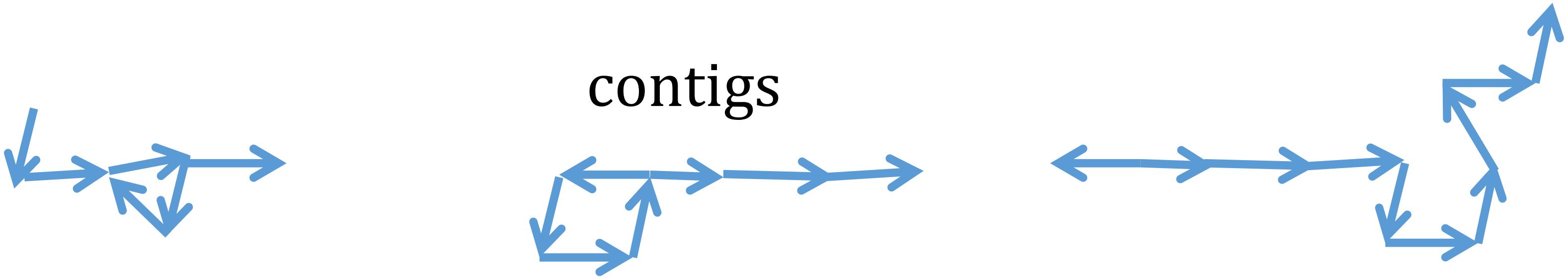
Tactics

- try values up to about 80 % of typical read length

- If *k* is too big you get many disconnected graphs (next slide)

can you assemble a genome yet ?

- do your best to follow graph
- visit each edge once..



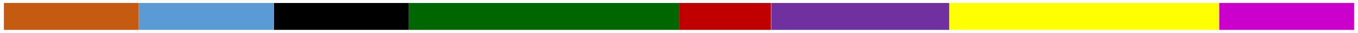
Data is not perfect

- many separate, contiguous pieces, not joined to each other

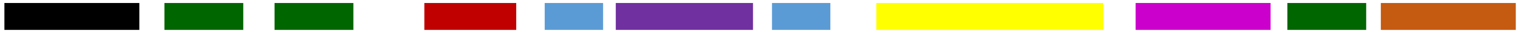
- solution – use some reference

reference genomes

correct answer.. (not known)



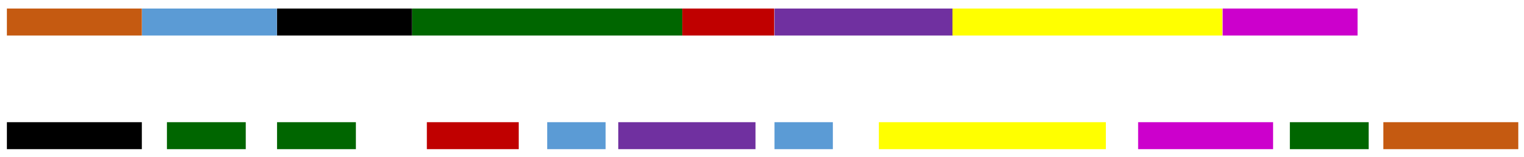
using available information you have…



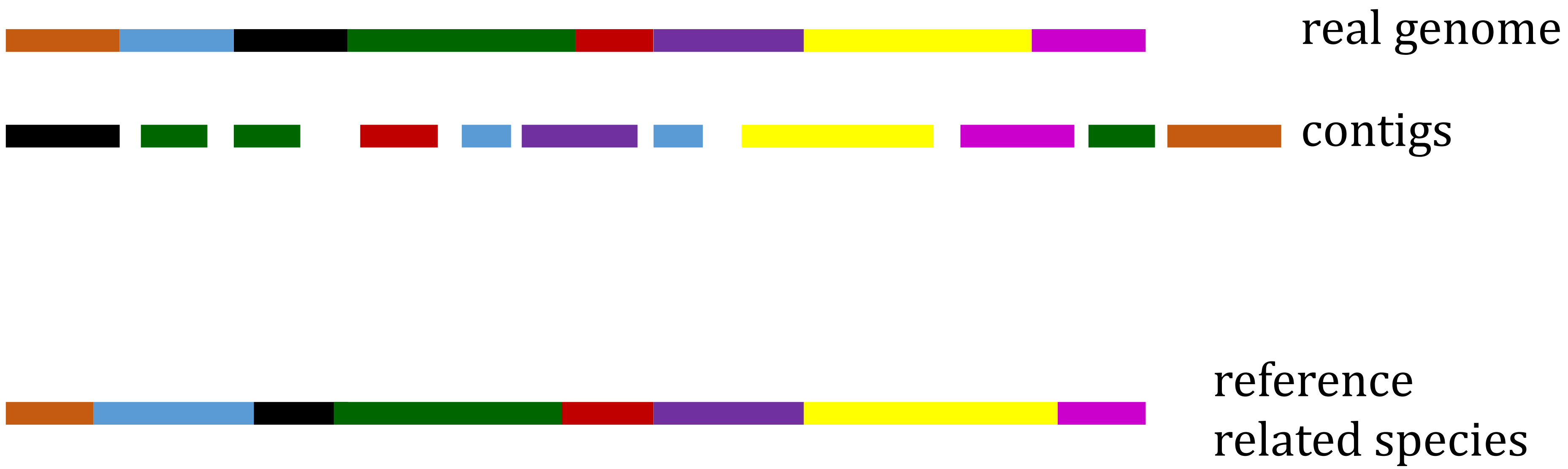
a set of contiguous pieces (contigs)

Need some way to assemble them to a best guess

- use some reference from the literature


real genome
contigs

using a reference genome



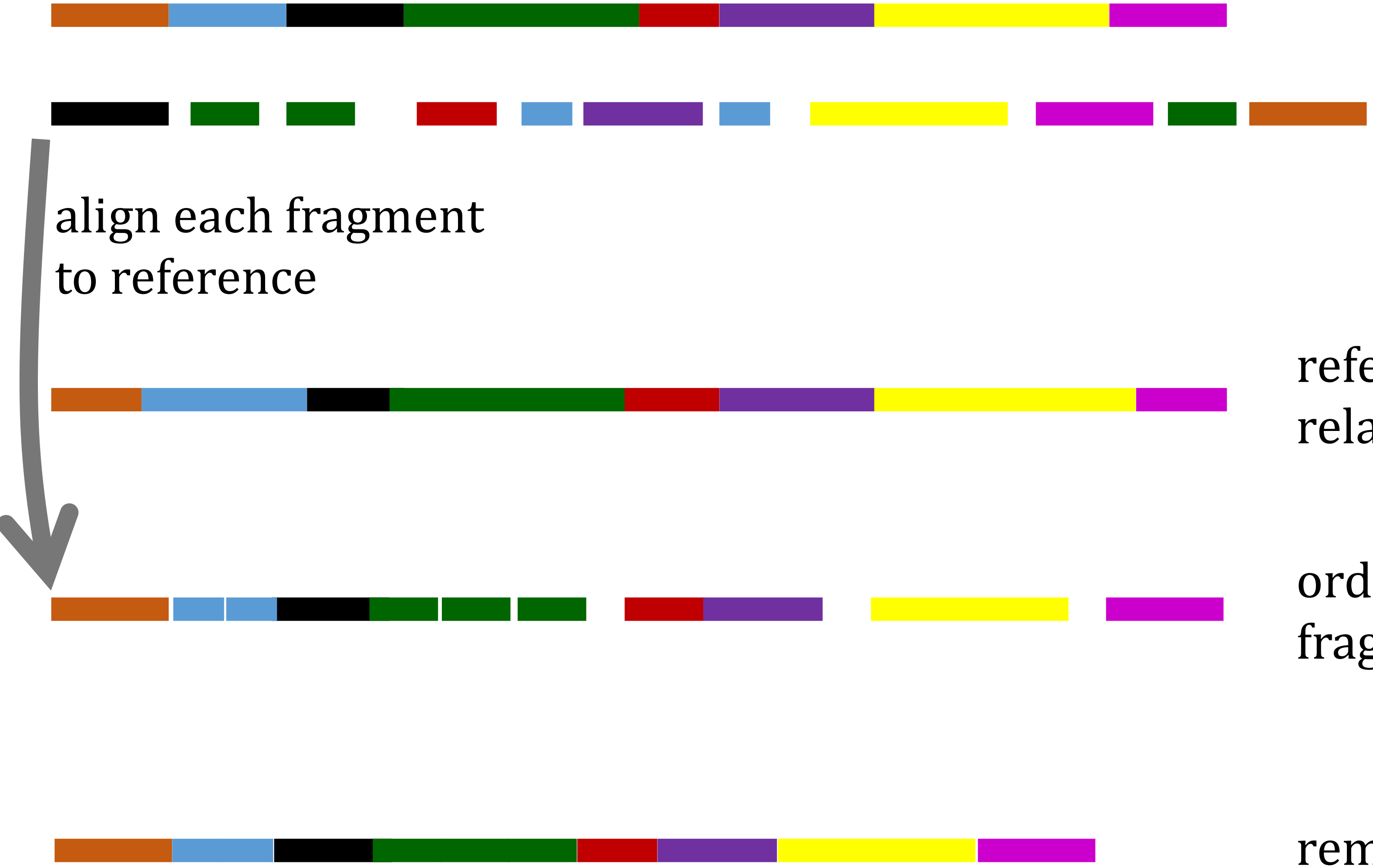
real genome

contigs

reference
related species

some related species

- good quality genome from literature
- for monkey, use man – for schäferhund use chihuaha..



real genome

contigs

align each fragment
to reference

reference
related species

order
fragments

remove gaps

Reference genome

Needs an *ab initio* genome

- expensive, long reads + short reads + computational effort

As of 2019

- 235 assemblies for human genome from 2014 to 2019

Most genomes today

- are not expensive slow *ab initio*
- use a reference genome

Danger

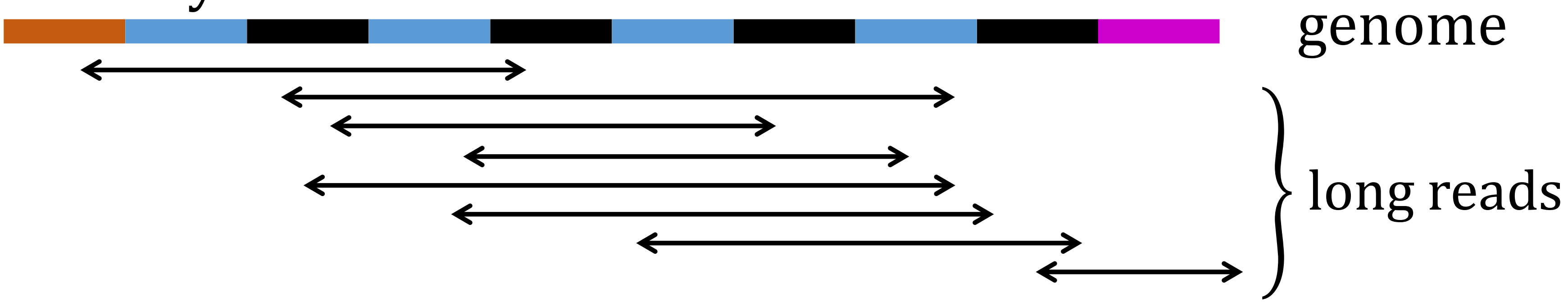
- if your reference is not close enough –there will be mistakes

Problems

- repeats
- errors
- natural variance

repeats

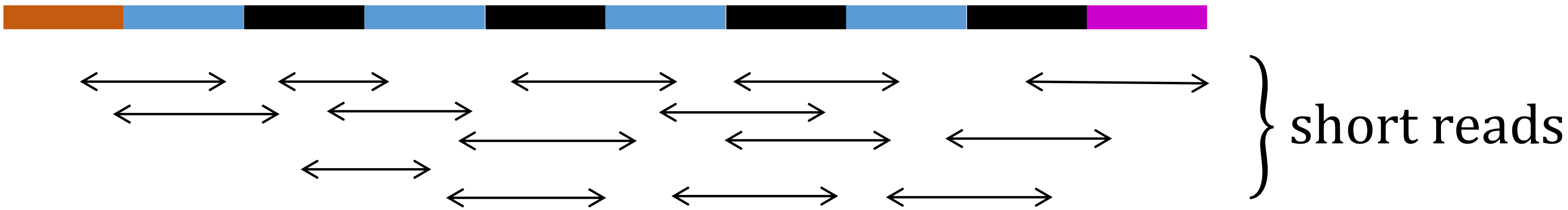
probably works



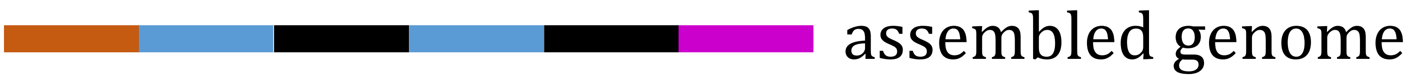
genome

long reads

will not be correct – short reads



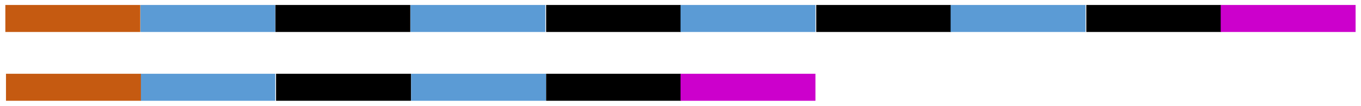
short reads



assembled genome

short reads and repeats

If two sequences can give the same patterns, you will pick the shorter



If read length < typical repeat length

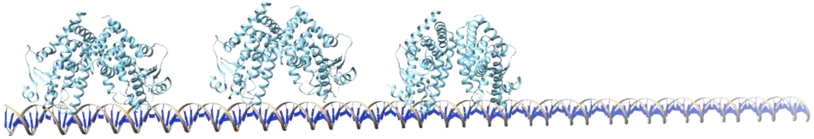
- you lose
- you cannot assemble an *ab initio* high quality genome

Can we just use longer reads ?

- not if you do not have the machine
- longer reads are error prone (better answer for klausur)

Repeats good and bad, but common

Useful repeat example – proteins that give DNA structure



- require similar non-binding sites on DNA
- repetitive, but important

Transposons, LTR = long terminal repeat

- substantial fraction of human genome
- evolutionary reason for a repeat to repeat ?

Repeats are hard to characterise

- sometimes not important – sometimes functional

Repeats – what to do

1. do not worry – live with it (not for klausur)
2. buy a machine with longer reads and keep acquiring data
3. paired ends and distances

`QRABABABABST`

can you use some experimental method (e.g. electrophoretic) to estimate distance `R..S` ?

Is there a correct genome ?

In one person

`ACTAG`	father
`ACCAG`	mother

Whose gene ? Yours ? Mine ?

- human genome project(s) – different people

Example problem

- you are recessive for haemophilia
- your recessive (bad) gene goes into the databases

How much variance is there ?

little

- we talk about the human genome

lots

- we can do DNA fingerprinting

How many bases in you cannot be explained by parents ?

- $10^1 – 10^2$

More detail later

An easy question

- …

error types

Random

- if I read more than once I will get different errors
- just make more reads

Systematic

- example: after a G, greater chance of error
- could recur with multiple reads

How big ?

What are errors

- wrong base C instead of T

	error rates
short reads	0.2 – 2.5 %
long reads	10 %
Sanger	< 0.1 %

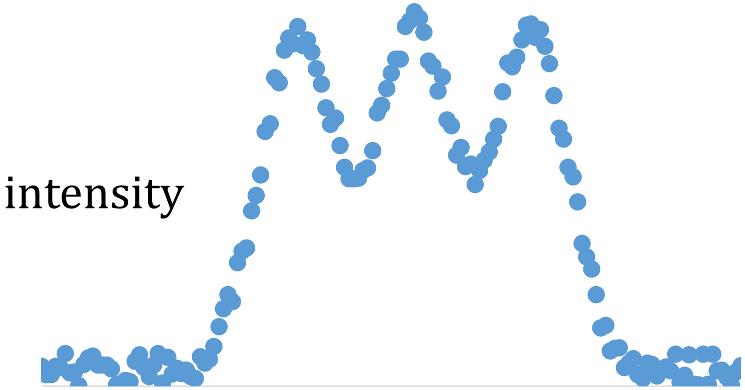
Errors

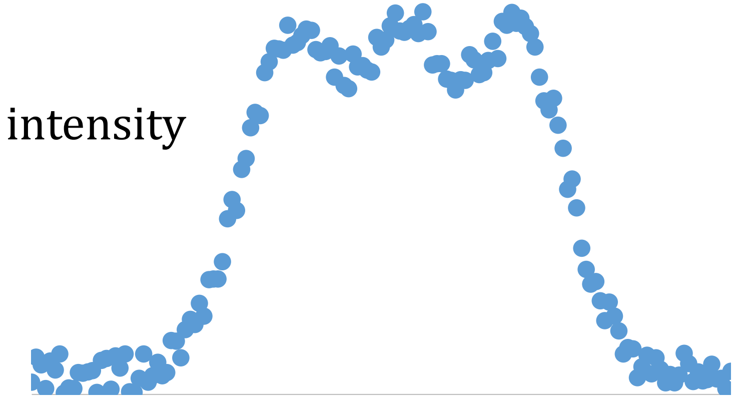
1.	before we start… contamination, mis-labelling, preparation, degradation, primer bias	not for these lectures
2.	machine reads wrong base / jumps over a base	relevant
3.	misassembly	

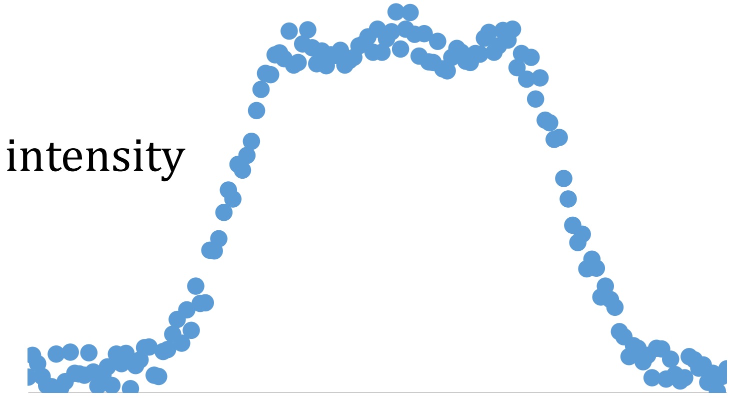
Machine errors – phase error

Different techniques, different properties – in general

- phase error







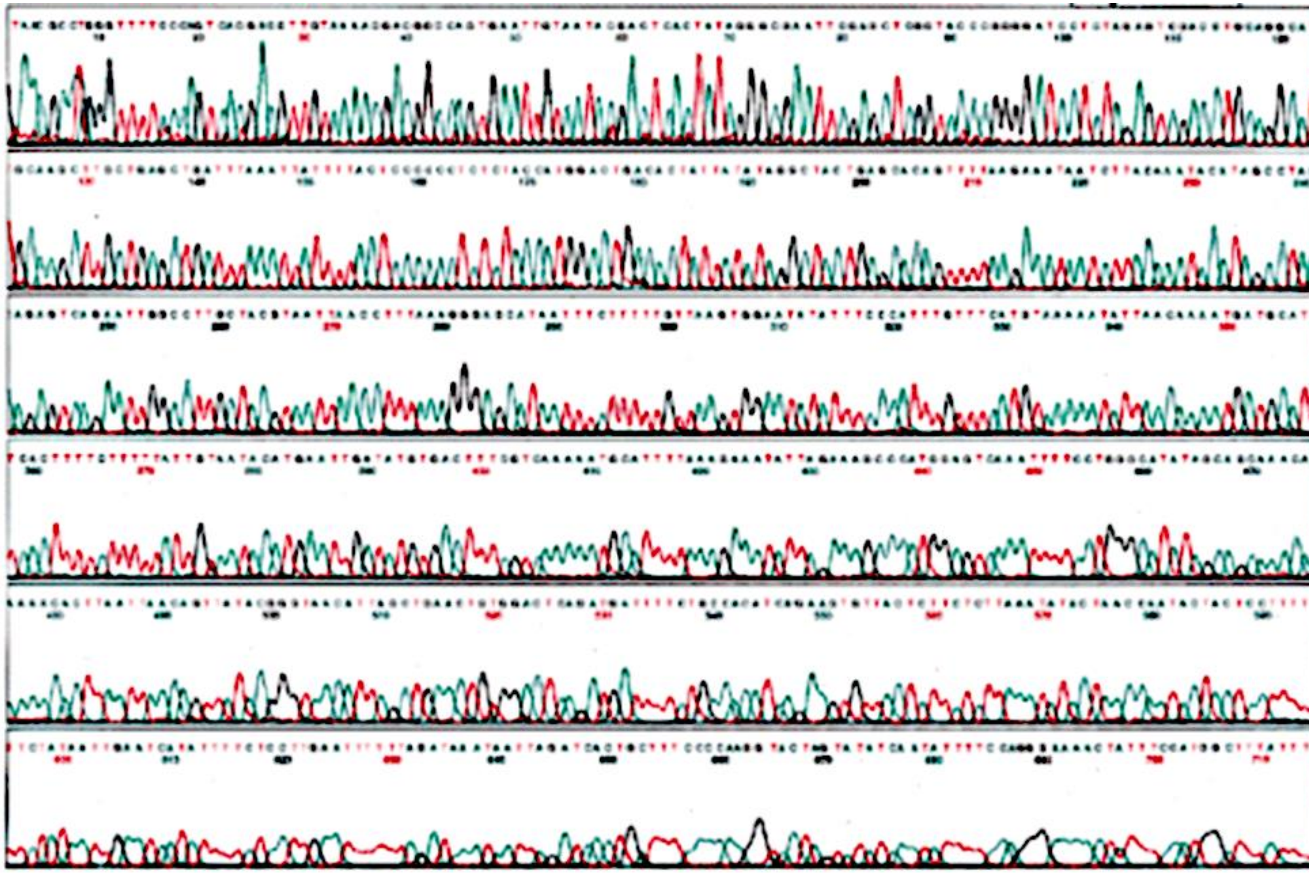
Machine errors – base calling

Intensity of A might be similar to G or

…

wrong base is read

bases read →



Rosenblum, B.B., LeeL.G., .. Chen, M. Nucleic Acids Res 25, 4500 (1997)

coverage

Most common quality measure

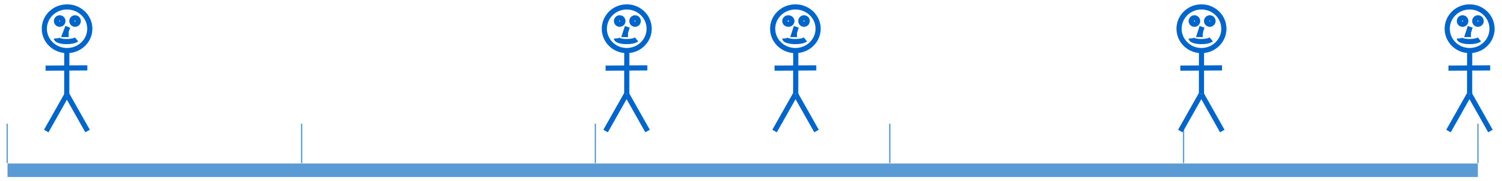
- my genome has n_g bases
- I sum up all my reads $n_r = \Sigma_i n_{read_i}$ so I have read n_r bases

On average, each base has been seen $\frac{n_r}{n_g}$ times = coverage

Good estimate ? Why not ?

Better statistical model

Customers in a shop, football goals, pedestrians at lights



time →

Average = 5 / 5 hr = 1/hour, but

- sometimes nobody comes for an hour
- events not correlated

Standard problem – use a Poisson distribution

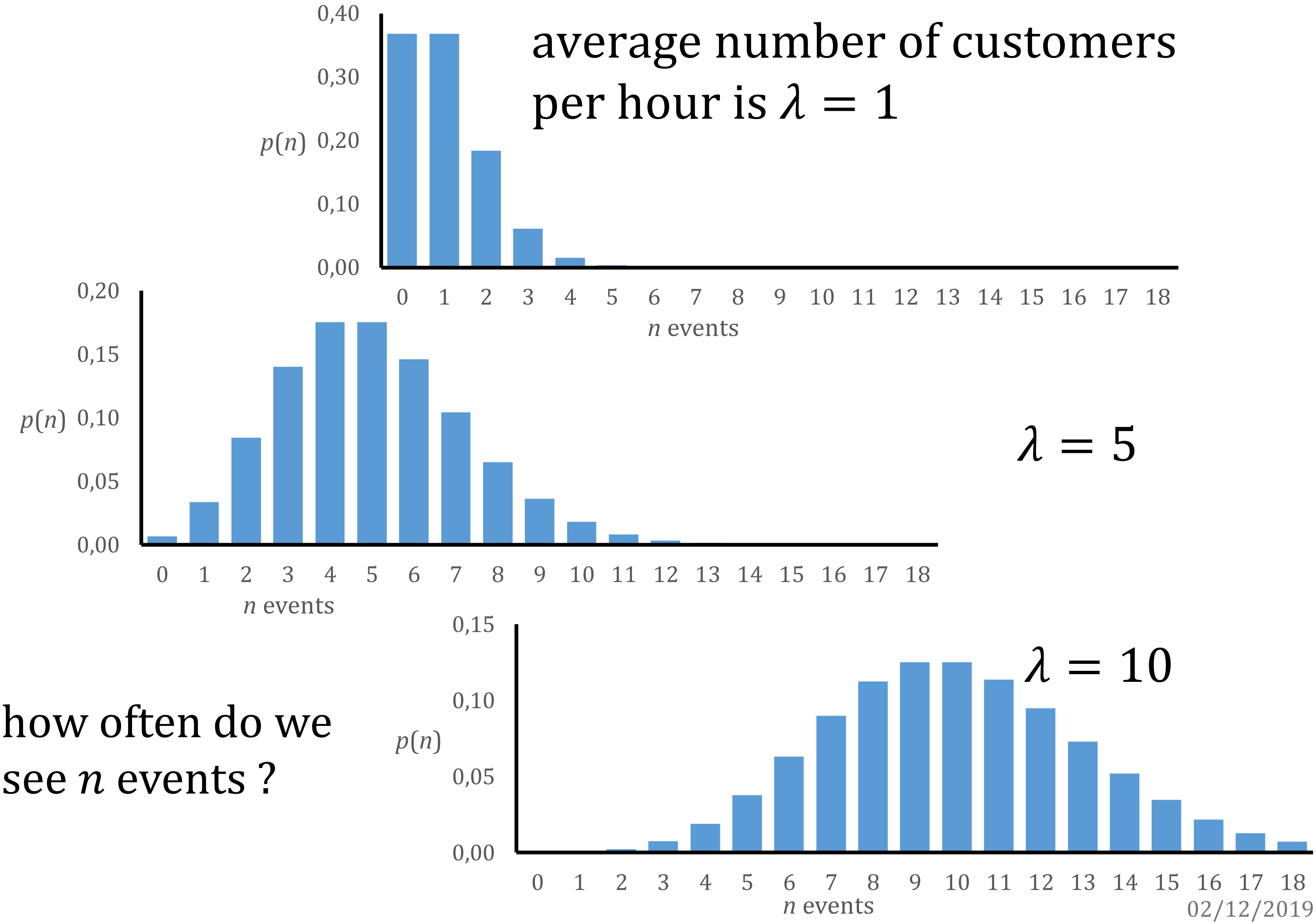
- λ is average number events / time

$$p(n) = \frac{e^{-\lambda}\lambda^n}{n!}$$

not for klausur

what does it look like ?

customers in shop, ion channel opening…



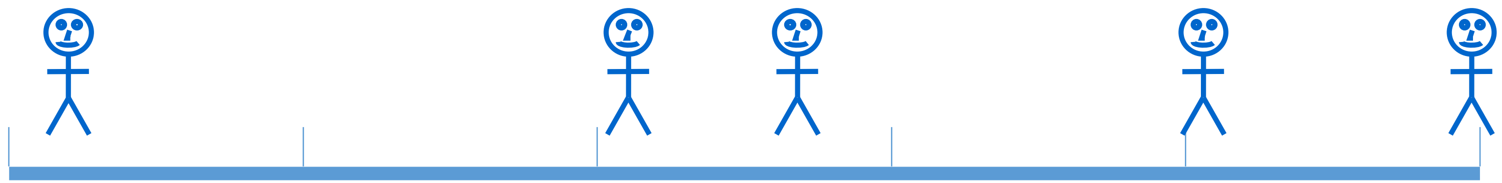
average number of customers per hour is $\lambda = 1$

$\lambda = 5$

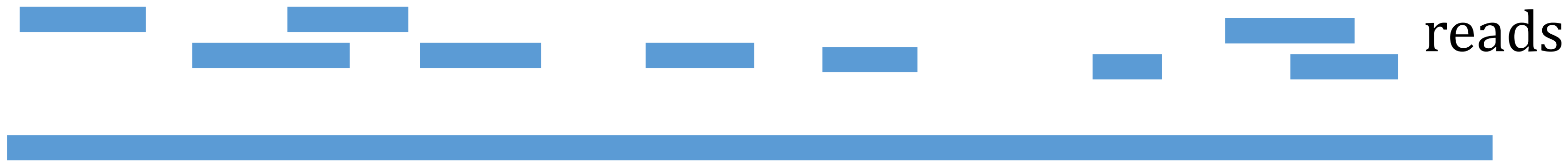
$\lambda = 10$

how often do we see n events ?

from customers to base reads



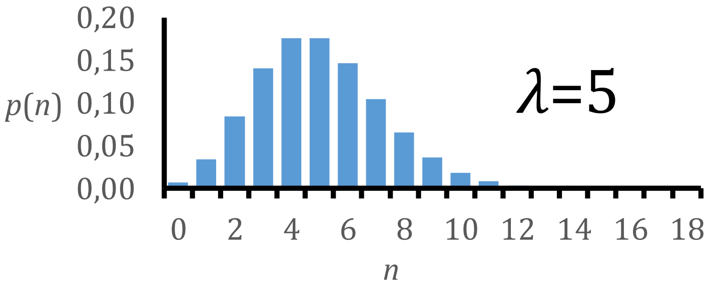
similar question



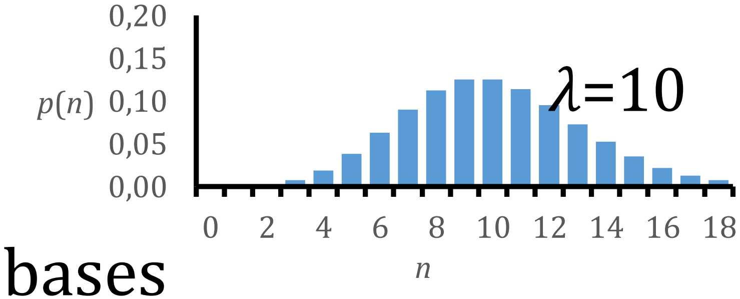
be practical…

Imagine a genome 3.3 ×10^9 bases

- if $\lambda = 5, p(0) = 0.007$
  2×10^7 bases are not sequenced



- if $\lambda = 10, p(0) = 4.5 \times 10^{-5}$
  1.5×10^5 bases not touched



- what if I have a bacteria with 10^6 bases
  $10^6 \times 4.5 \times 10^{-5} = 45$ bases not sequenced

Do not take numbers too seriously

approximations

- I have left out read lengths
- Poisson is not quite appropriate

Important

- 10-fold coverage does not mean each site has been seen 10 times
- n-fold does not mean…

- n-fold coverage does not mean there are no mistakes

executive summary

Is it possible to get a near perfect genome ?

- should we use lots of long reads ?
- lots of money and time (years for human genome)
  - probably never perfect

Practical genomes have errors

Errors

- random – can be removed with much sequencing
- systematic – need even more sequencing

Repeats

- rarely resolved, but very common in eukaryote genomes

What do you want ?

- Quickly compare two species ? Cheap genome with errors
- Find variants in human genes ? Expensive slow genome

Relevant to later topic

- a gene variant (single nucleotide variant) looks like a reading error
- a rearrangement looks like an assembly error

No more gene assembly

Open Reading Frames and genes

Lots of genomes – not many diseases cured, revelations

- what products are made in which cells ?
- how are they spliced ?
- how are they regulated ?...
- which proteins are made in children / under stress / … ?

More fundamental

- can you look at the human genome and say
- "here are the genes" ?

How much of genome is useful ?

- Prokaryotes ? Most of the genome
- People ? 2 – 60 %

General claim

- bacteria / archea
  - simpler, smaller, no ethical problems in experiments
- animals
  - nasty – most of genome does not code for proteins
- plants
  - very nasty – huge genomes, much duplication

human genes how many ?

- $2 – 3 \times 10^4$ protein genes
- experimentally likely (gencode)

protein coding	2.0×10^4
RNA non-coding	2.4×10^4
pseudogenes	1.5×10^4

What is the task in a popular eukaryote genome ?

- finding the few coding regions in a huge soup

In a prokaryote ?

- finding start points and removing the small amount of other material

Two tasks

Tasks

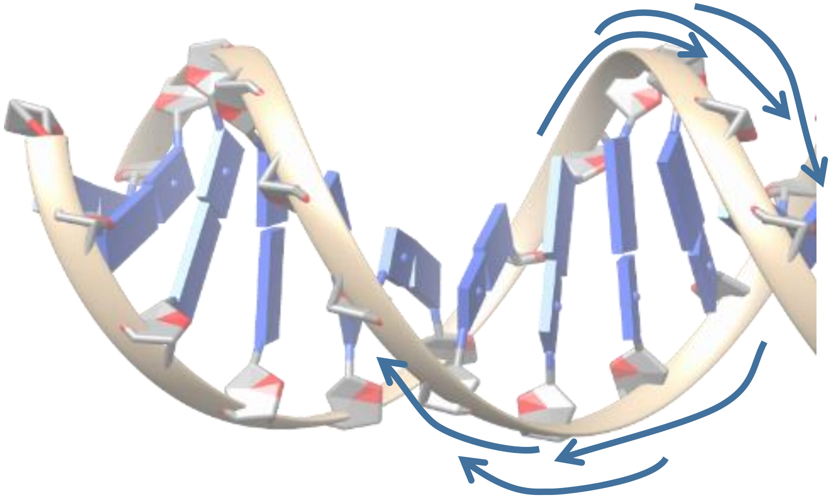
1. Find the reading frame
2. find start / stop / introns

Methods

- *de novo* / *ab initio* (look at just one genome)
- homology

finding the reading frame

- recurring theme
- six possibilities



three different reading frames

```
>A01592.1 Human haemoglobin A beta chain
GTGCACCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAAGGTGAACGTGGA[…
]TTCAGGCTCCTGGGCAACGTGCTGGTCTGTGTGCTGGCCCATCACTTTGGCAAAGAATTCACC
CCACCAGTGCAGGCTGCCTATCAGAAAGTGGTGGCTGGTGTGGCTAATGCCCTGGCCCACAAGT
ATCAC
```

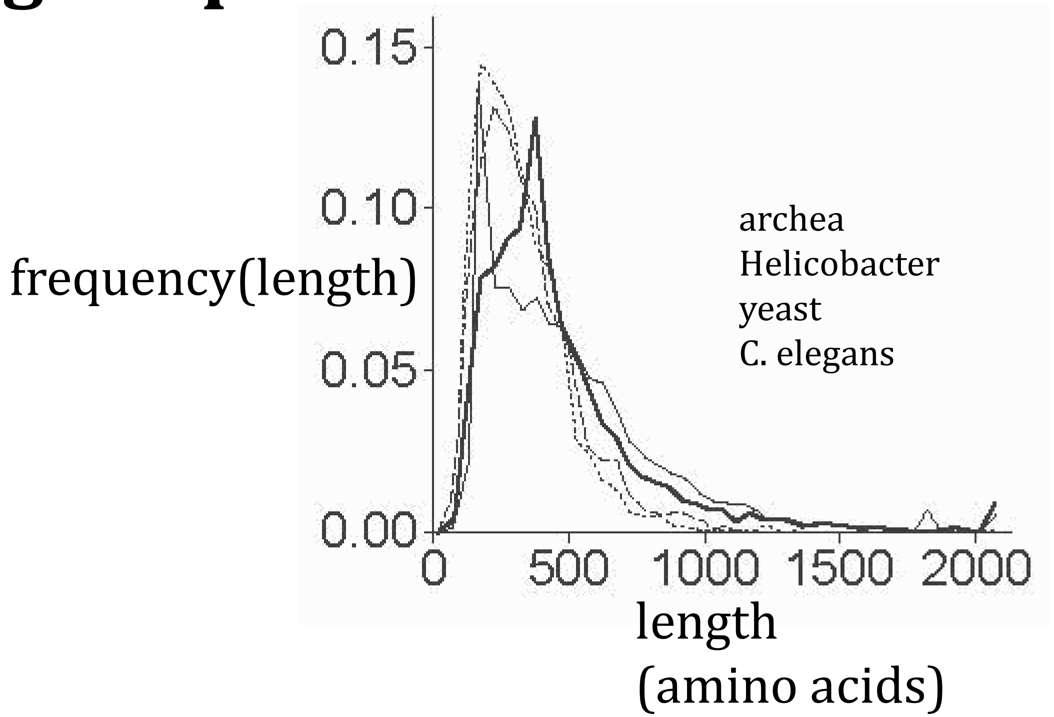
```
VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHG
KKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQA
AYQKVVAGVANALAHKYH
```

```
CTStopLLRRSLPLLPCGARStopTWMKLVVRPWAGCWWSTLGPRGSLSPLGICPLLMLLWATL
RStopRLMARKCSVPLVMAWLTWTTSRAPLPHStopVSCTVTSCTWILRTSGSWATCWSVCWPI
TLAKNSPHQCRLPIRKWWLVWLMPWPTSI
```

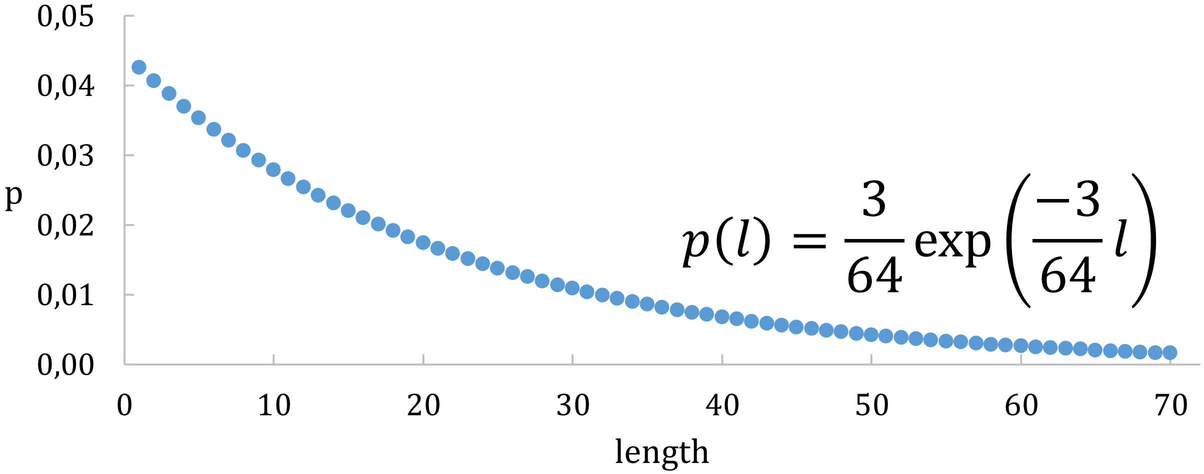
```
APDSStopGEVCRYCPVGQGERGStopSWWStopGPGQAAGGLPLDPEVLStopVLWGSVHSSt
opCCYGQPStopGEGSWQESARCLStopStopWPGSPGQPQGHLCHTEStopAALStopQAARG
SStopELQAPGQRAGLCAGPSLWQRIHPTSAGCLSESGGWCGStopCPGPQVS
```

How long are proteins ?

experimental



random stops
every 21 codons



$$p(l) = \frac{3}{64} \exp\left(\frac{-3}{64} l\right)$$

Zhang, J. Trends Genet. 16, 107-109 (2000)

use frame with longest sequences ?

Proposal

- Try six reading frames
- pick one which leads to longest sequences

Would it work ?

- often
- not enough for long genomes

length of random sequences

Random sequences about $\frac{3}{64}$ would be stop codons:

average length between stops $\approx 64/3$

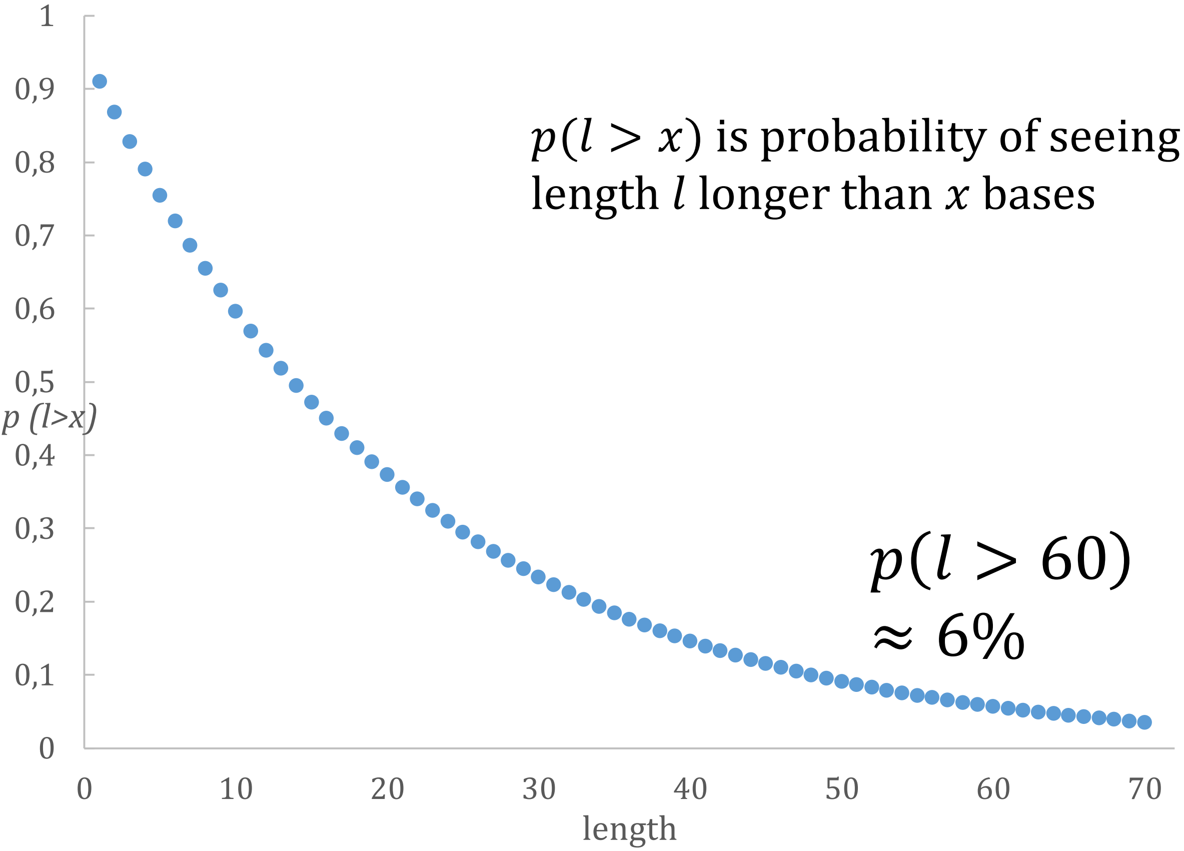
Random sequences

$p(l \geq 60) \approx 0.06$

$p(l \geq 200) \approx 10^{-5}$

Genome 10^9 bases

→10 000 stretches of more than 200 with no stop codon $(10^9 \times 10^{-5})$



$p(l > x)$ is probability of seeing length l longer than x bases

$p(l > 60)$
$\approx 6\%$

picking the wrong reading frame would give 10^4 long sequences

codon usage and reading frame

length is helpful, but not sufficient
What else characterises the reading frame ?

```
>A01592.1 Human haemoglobin A beta chain
GTGCACCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAAGGTGAACGTGGA[…
]TTCAGGCTCCTGGGCAACGTGCTGGTCTGTGTGCTGGCCCATCACTTTGGCAAAGAATTCACC
CCACCAGTGCAGGCTGCCTATCAGAAAGTGGTGGCTGGTGTGGCTAATGCCCTGGCCCACAAGT
ATCAC
```

reading frame 1: `GTG CAC` …

reading frame 2: `TGC ACC` …

reading frame 3: `GCA CCT` …

In each of six frames

- count how often each of 64 codons occurs

codon frequency

Example

E. coli has

- `CTG` (leu) as very frequent codon
- `AGG` (arg) very rare codon

You should not see `ACG` very often

- can we formalise this ? Invent a score ?

Score for codon usage

For some stretch of DNA

- we have codons 1, 2, 3, .. 64
- observed frequency of codon 1 in your sequence p_1^{obs}
- expected frequency of codon 1 is p_1^{exp}
- score

$$\prod_{i=1}^{64} p_i^{obs} \cdot p_i^{exp}$$

When is score maximal ?

Normalisation or complication ?

- we are only interested in comparing six reading frames

Start of reading frames ?

Start signals `AUG`

- only about 83% in *E. coli*
- many (thousands) of exceptions in eukaryotes

- eukaryotes ? Alternative protein forms + diseases

- Put all of this together… Find the reading frame

for each of six possible reading frames

probability based on protein length (longer = better)

× probability based on codon composition

× probability based on start codon

Finding a coding region

Prokaryotes and mitochondria… little junk

- coding
- regulatory
- RNA genes

Eukaryotes… > 98 % probably junk

Several programs

- general philosophy

gene signals

Find possible reading frames then

- Each protein-coding gene has
  - start codon
  - stop
  - stop – start position = length

  - ribosome binding site

Consider two approaches

rule-based

- find all possible starts (`ATG`/`GTG`/`TTG`) and score them
- find all stops and mark regions between as candidates

- use table of known ribosome binding sites – score probability
  - 3-4 bases early `GGA`, `GAG`, `AGG`…

  [… lots more]
  - 5-10 bases early `AGxAGG`, `AGGxGG`

- calculate length of implied protein
  - lookup probability of this length protein

- rank all guesses by their score/probability

more sophisticated

- do not explicitly think of ribosome initiation
- do not rely too much on known initiation sites

belief

- there is some pattern that precedes each protein
  - maybe within base-pairs
- proteins start with one of a small set of triplet
  - maybe `ATG`/`GTG`/`TTG` sometimes not
- introns and exons have characteristic sequences

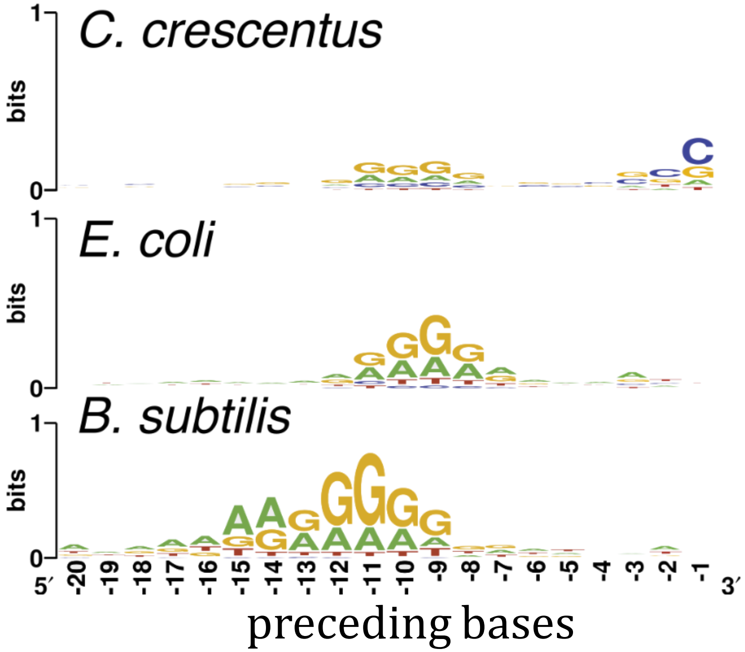
requires

- some list of correct proteins / corresponding sequence

Example

For three bacteria

- what is the probability of each base type at preceding positions ?



Summarise these ideas:

Gene finders are

- trained / calibrated using species-specific, known information
- using
  - initiation codon
  - probabilities of preceding bases
  - expected protein lengths

Hockenberry, A.J., Stern, A.J., Amaral, L.A.N., Jewett, M.C., Mol Biol Evol, 35, 582 (2018)

Using homology / similarity

Everything so far

- based on looking at one genome alone.. Much better:

Imagine you have databases full of proteins

- from many species
- mostly correct

Take any DNA sequence

  - get 6 reading frames
  - translate each to amino acid sequence
  - do a database search

- Only one reading frame will find known proteins
- in this region – search for a start and stop codon

Homology searching for genes

Preferred approach

- very fast (blastx)
    - translates in six reading frames and does a search
    - gives you the literature function (annotation) for a gene if present

Will you find genes

- for some new monkey ? (lots of primate sequences)
- for an exotic fungus that causes some nasty disease ?
- a south American plant which is a possible future food ?

Summarise all the problems

Problems

de novo searching

- lots of false negatives (missed genes)
    - unusual initiation properties
    - too long or too short

Searching with homology

- requires a source of related proteins
- propagates existing errors
    - wrong annotations / functions
    - genes that really do not exist
- pseudogenes

All methods – suffer from errors in genome assembly

What proteins are made and how much ?
RNAseq

Two questions about transcription

- what genes are transcribed ?
- how are proteins spliced ?

First.. how helpful are genomes ?

problems with genomes

So far most of semester has focussed on either

- protein sequences or
- DNA sequences

Most of genome is junk DNA (controversial)

You do not know every reading frame

- of those frames (putative proteins)
  - which are dead genes / pseudo genes ?
  - which genes are active in which cell types
  - what are the splice products ?

You do not really know what is being made from genome

genomic products

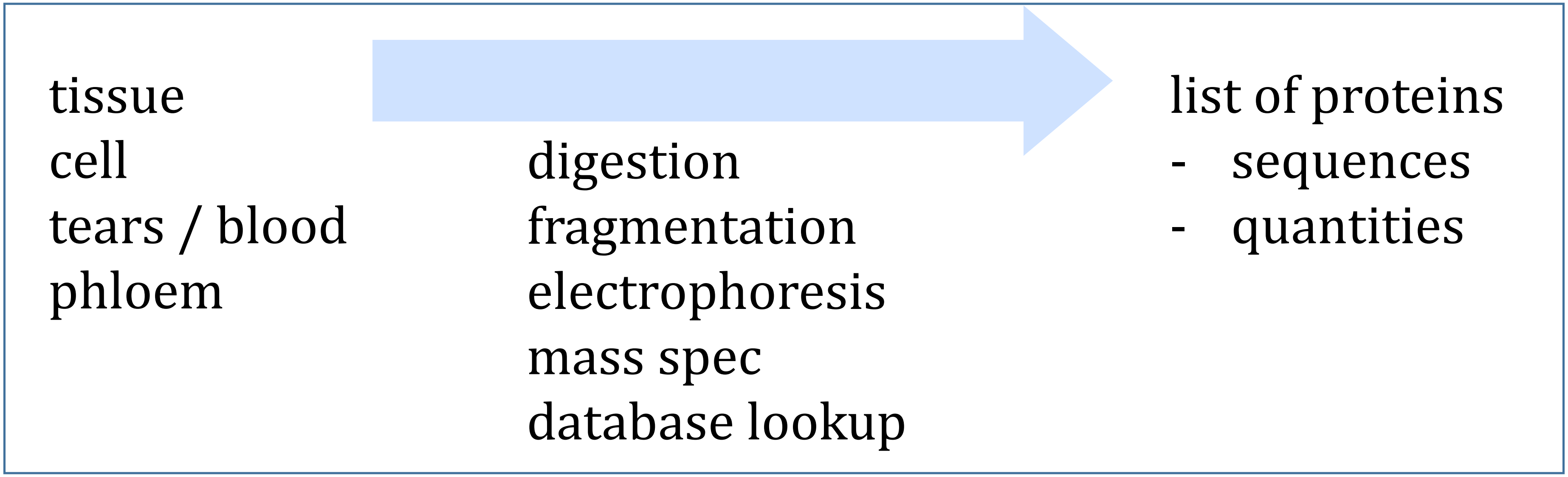
Regulators – interesting but not for today

- proteins
  - look at directly ? proteomics
- nucleotide approaches (to measuring proteins) ?

First consider proteomics

Proteomics (not here)

What proteins have been made in a cell ? some fluid ? some sample ?



tissue
cell
tears / blood
phloem

digestion
fragmentation
electrophoresis
mass spec
database lookup

list of proteins

- sequences
- quantities

Proteomics

Does measure proteins, but..

- analysis is very dependent on known proteins
- distinguish α- / β- haemoglobin (sequences similar)?
- could you recognize a new splice form ?

Can you look at nucleotides instead ?

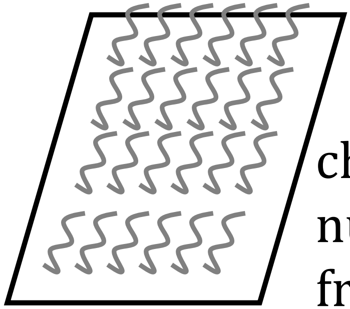
- cheap and fast
- very sensitive

- not so direct

Microarrays

Looking for some known products

- wash sample over chip
- detect fluorescence, precipitation

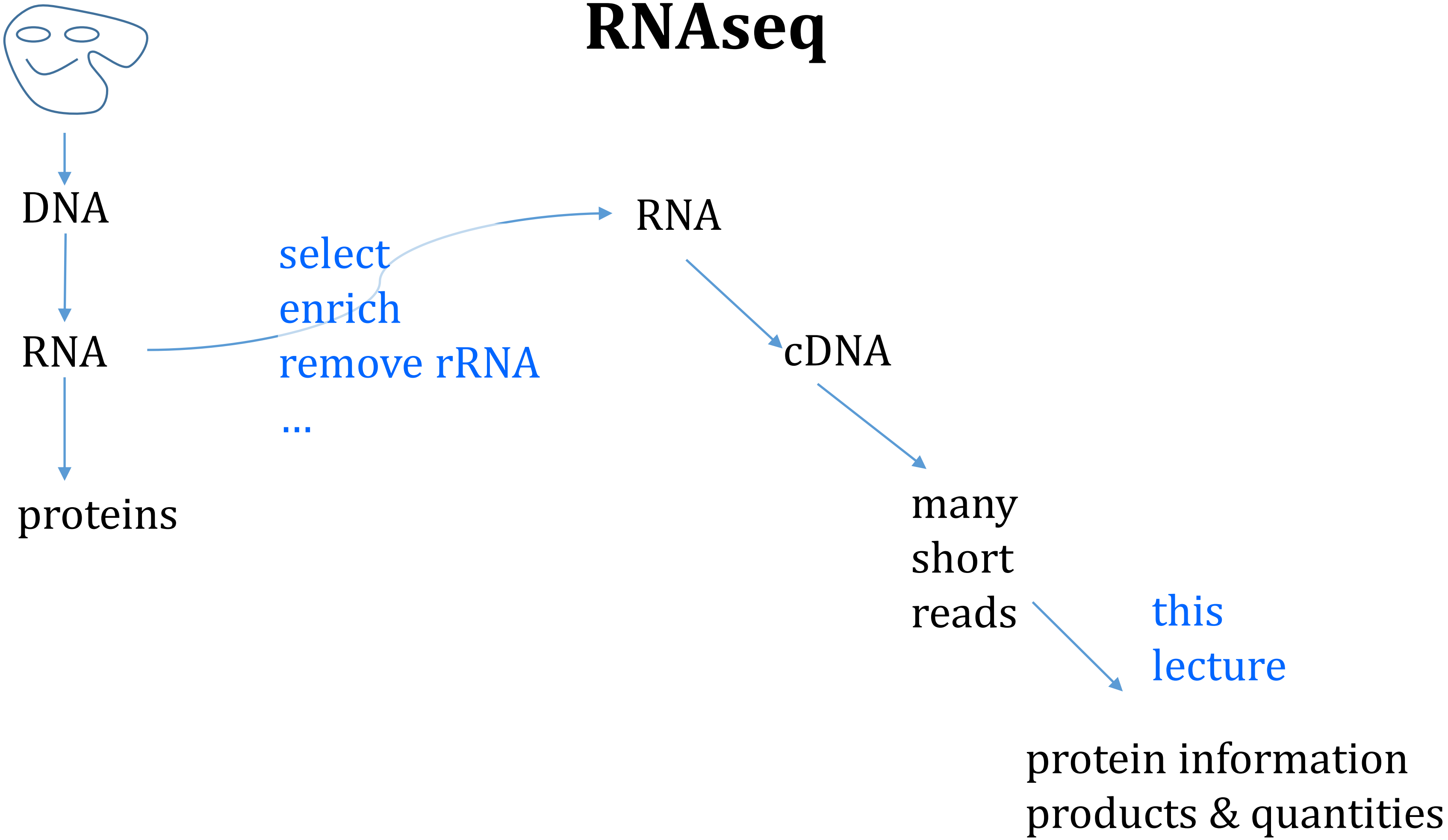


chip with
nucleotide
fragments bound

- fast
- can work with modified bases
  - used by Prof Ignatova to look at tRNA

- limited to known sequences

RNAseq



DNA

RNA

proteins

select

enrich

remove rRNA

...

RNA

cDNA

many short reads

this lecture

protein information

products & quantities

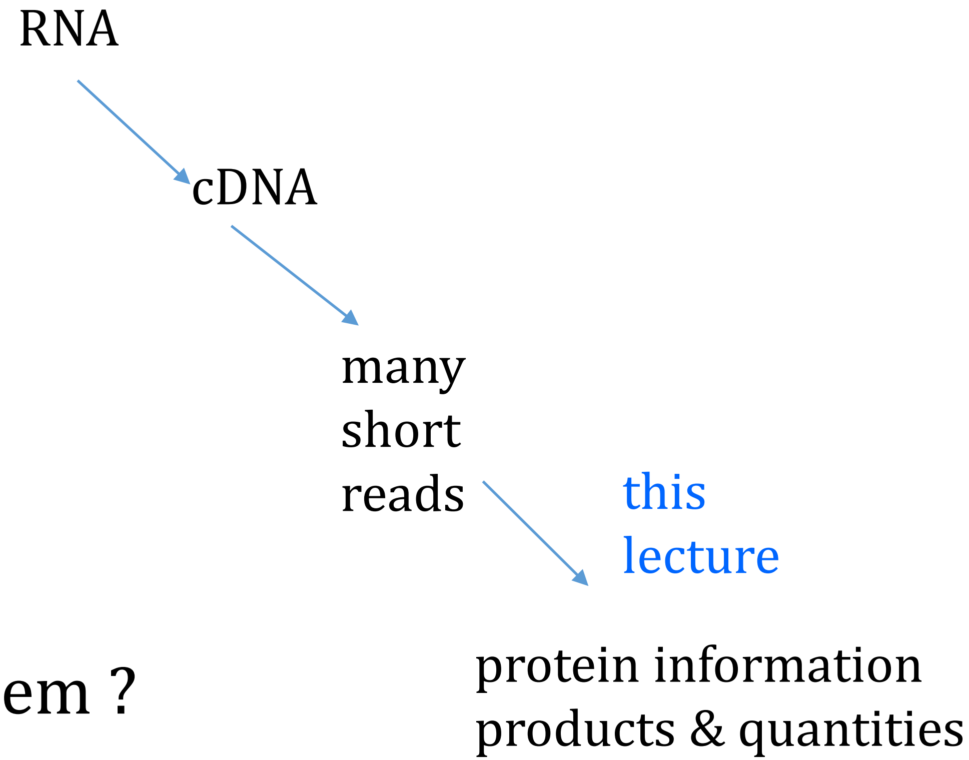
- very indirect
- very sensitive
- not limited to lists of known products

RNAseq

Two issues

- mapping
- quantification

What is the mapping problem ?



RNA

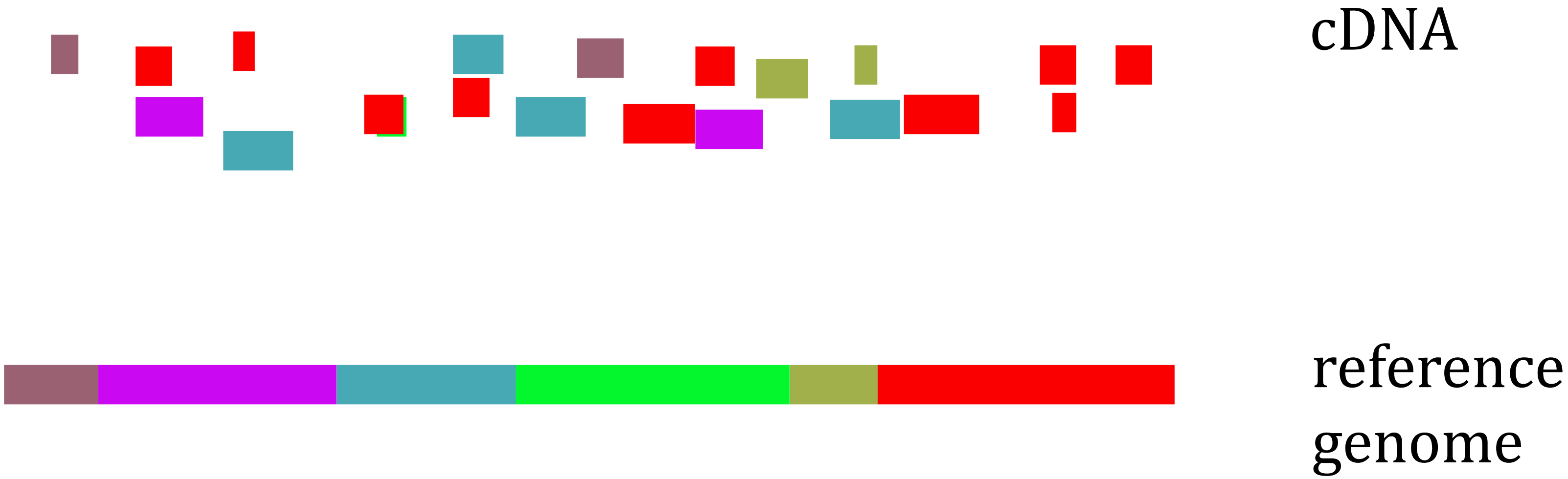
cDNA

many short reads

this lecture

protein information products & quantities

Mapping –simple quantification



cDNA

reference genome

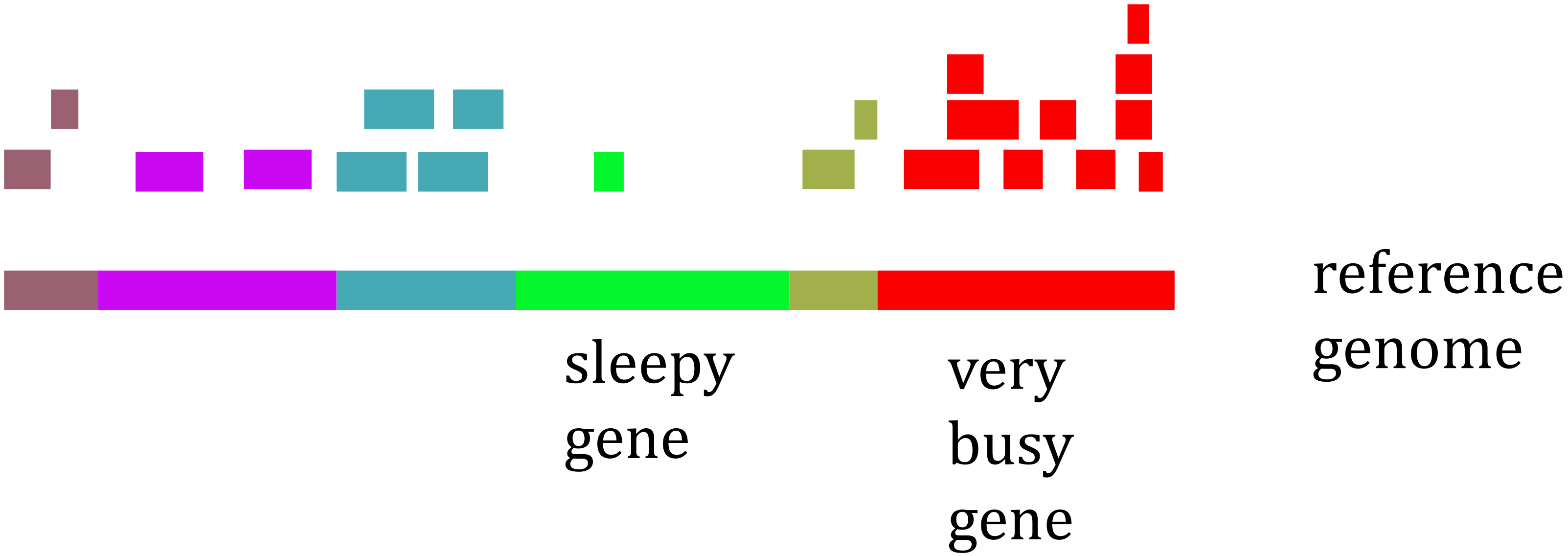
starting point

- soup of short DNA reads

Get a reference genome

- map to reference

Mapping –simple quantification



sleepy gene

very busy gene

reference genome

Intuitively

- 🟥 is very much expressed
- 🟩 not expressed
- Assumptions – we measure DNA and align it
- amount of DNA reads depends on amount of RNA
- amount of RNA reflects protein being synthesised (or RNA or some other biochemistry)

Quantification

Different questions

1. response to changing conditions

- hot / cold, food / hungry , antibiotic…

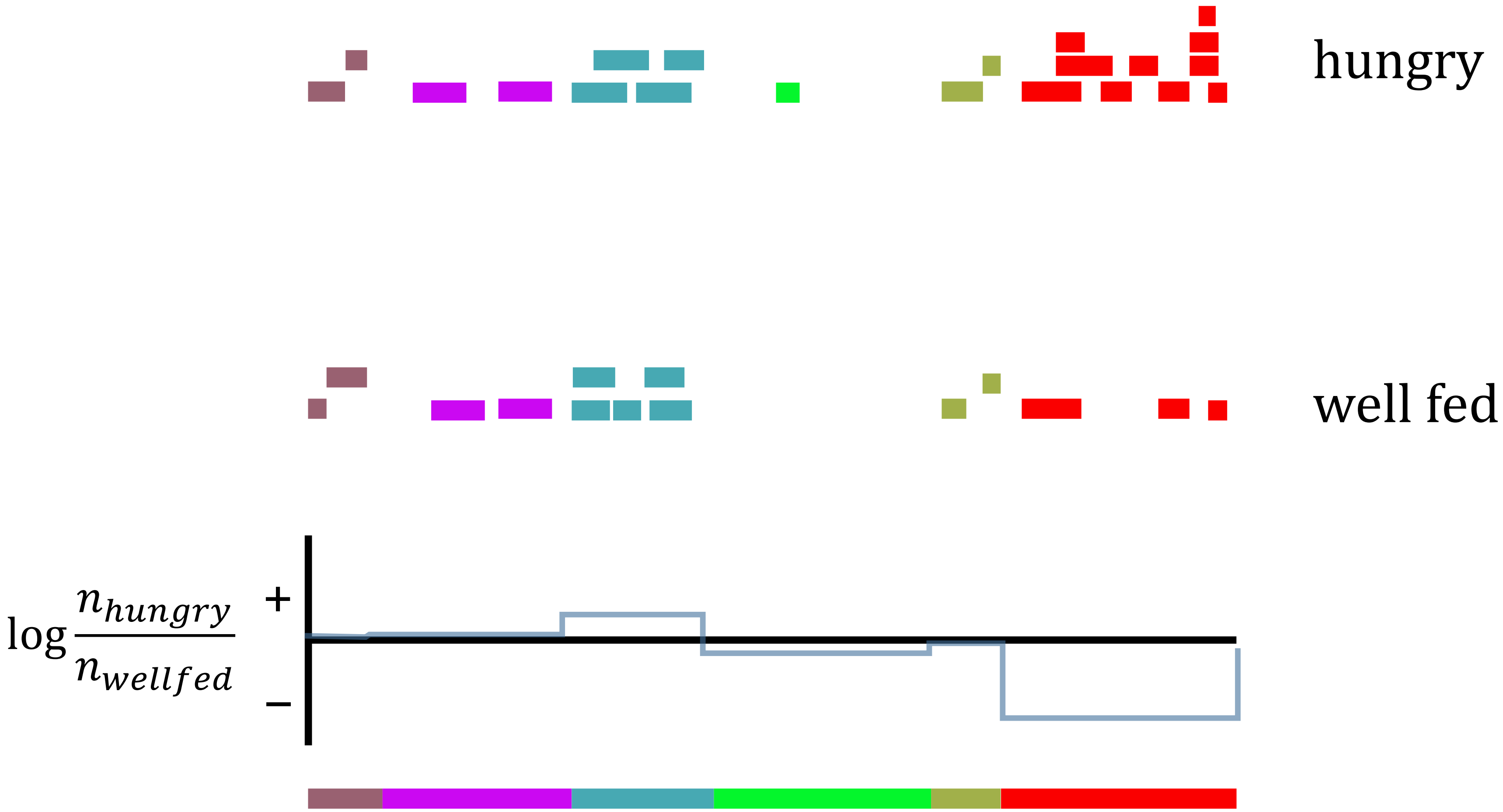
2. what proteins are being made ?

- is this protein unique to nerves / liver / … ?

The issue

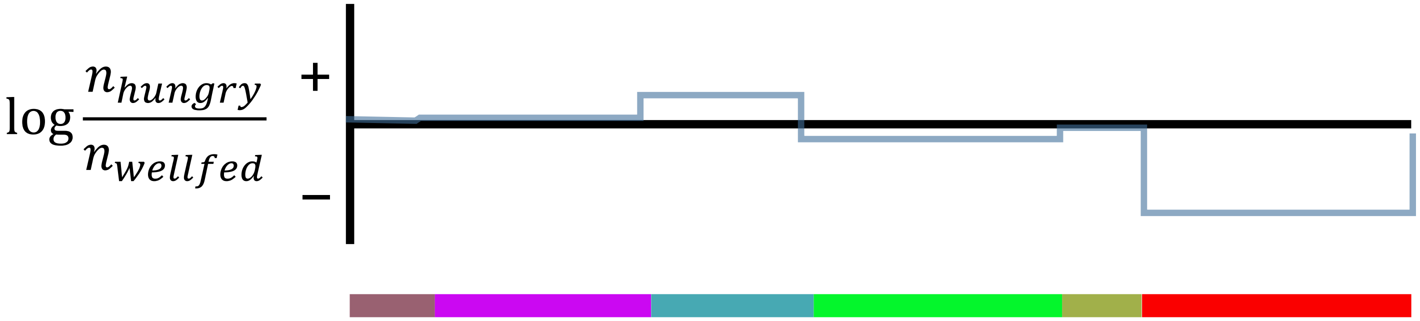
- normalising / references / absolute levels

Relative changes



Plot logarithm of changes from experiment$_1$ and exp$_2$

systematic relative changes



We have two measurements

- what will happen with sensitivity differences ?
- we want $\log\left(\frac{n_{hungry}}{n_{wellfed}}\right)$ but we measure $\log\left(\frac{n_{hungry}}{c{\cdot}n_{wellfed}}\right)$

  $\log\left(\frac{x}{ay}\right) = \log\frac{x}{y} - \log a$ so we get a shift of blue line

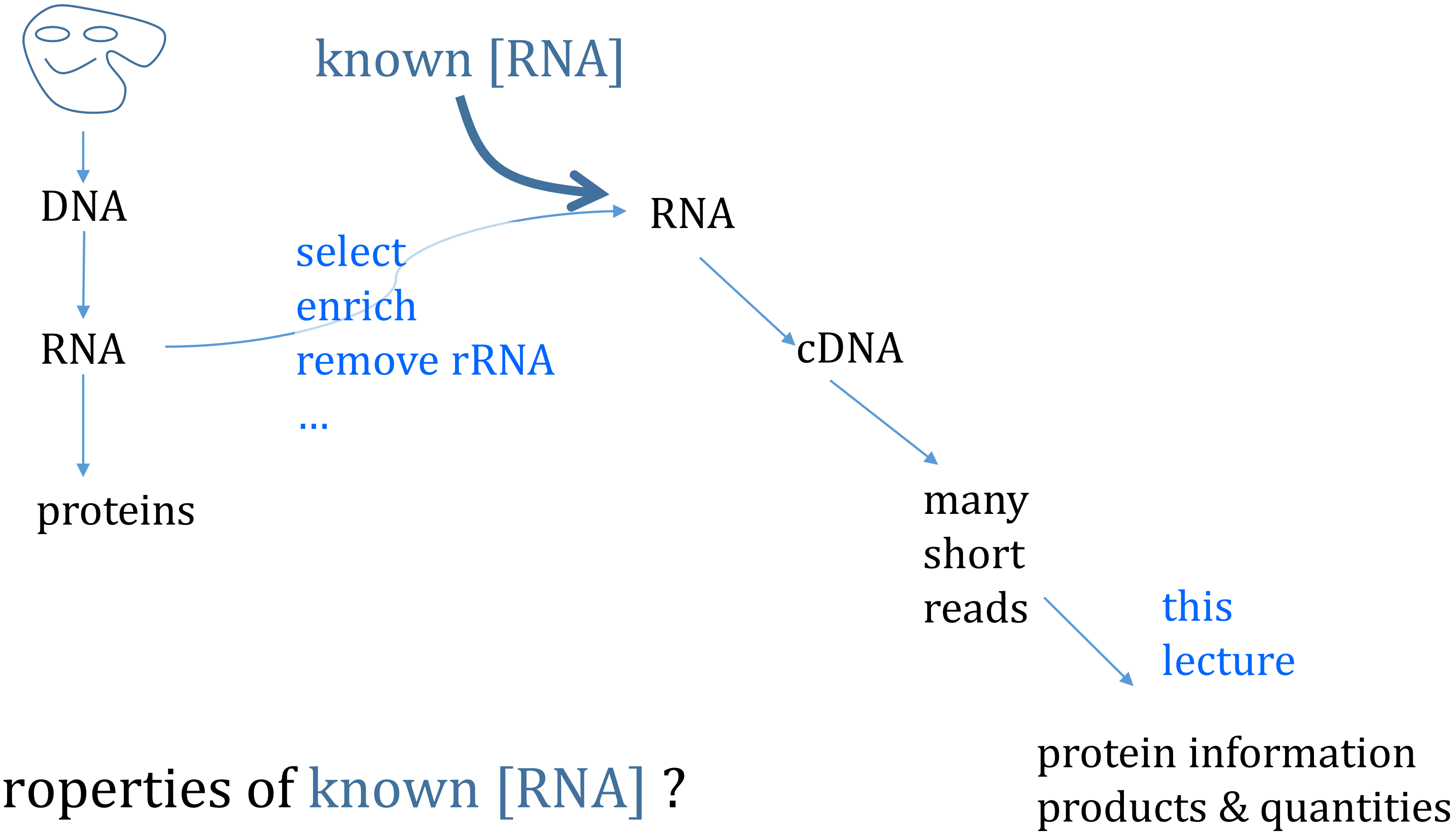
If we are just looking at changes

- do not need absolute quantities
- can tolerate some systematic change in sensitivity

Problem with relative measurements

What if conditions generally suppress/enhance translation ?

- needs another reference



known [RNA]

DNA

RNA

proteins

select

enrich

remove rRNA

...

RNA

cDNA

many short reads

this lecture

protein information products & quantities

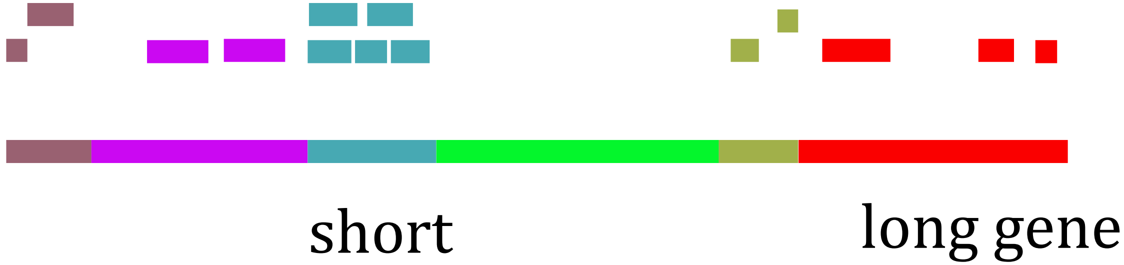
Properties of known [RNA] ?

Known RNA quantities

- kits available
  - called "spike-in"
  - should have
    - GC content – should be similar to your sample
    - length – lets you check for length bias

Last part of measuring gene expression…

How much of a gene is expressed ?



By chance you expect more reads from red gene 🟥

- if you want to talk about how much protein is made

$$\frac{n_{reads}}{n_{length}}$$

Only works if you have

- reference genome
- annotated genes

summarise all these steps

Quantification

Relative levels of expression

- simplest

Control for overall suppression / enhancement, different measurements

- "spike-in" known [RNA]

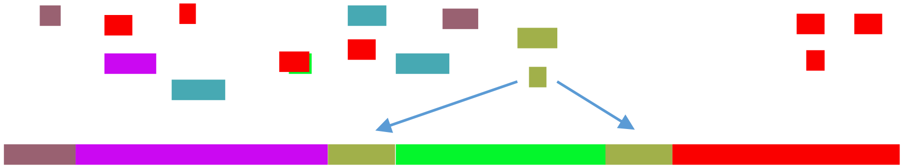
Quote amount of protein

- divide by sequence length

read mapping problems

General problems – sequence information

- sequencing errors
- genomic variation
  - my protein may not be found on your reference
- repetitive elements



RNA-seq specific problems

- spliced alignments – makes it much more interesting

Splice variants

How many genes do we have ?

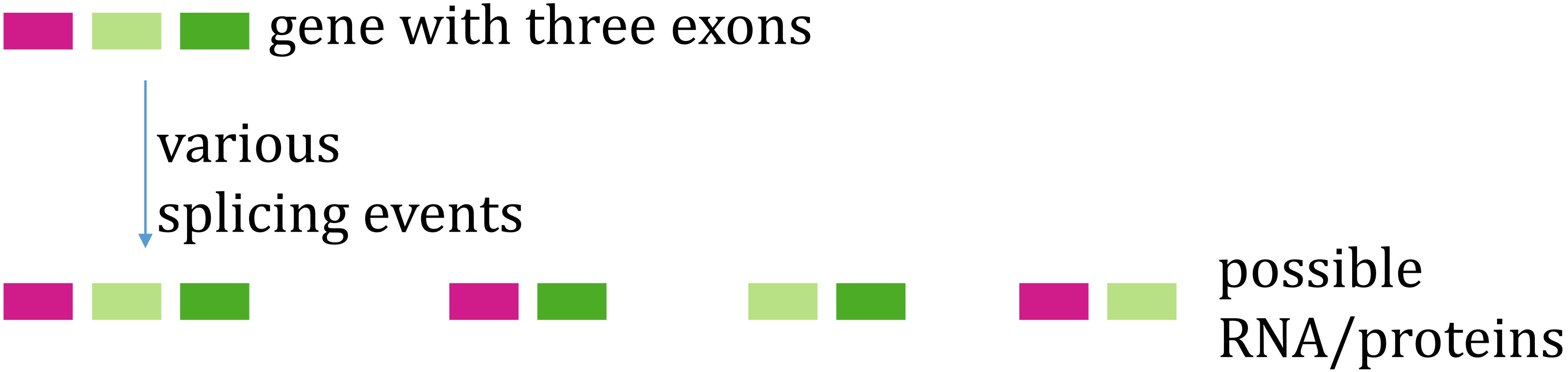
- 2×10^4

How many proteins do we have ?

- 10^5 to 10^7 or …

What is the difference between a nerve cell and a liver cell ?

1. which genes are turned on
2. how the pieces are put together



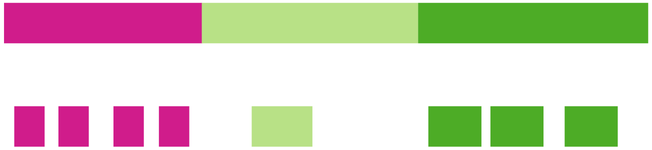
gene with three exons

various
splicing events

possible
RNA/proteins

What if we had four, five, .. exons ?

What does one need to see the variants ?

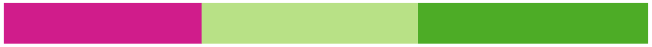


one annotated gene in reference genome

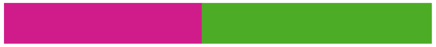
DNA reads

Enough to say this gene has been seen 8 times ?

Better explanation of data:



rare



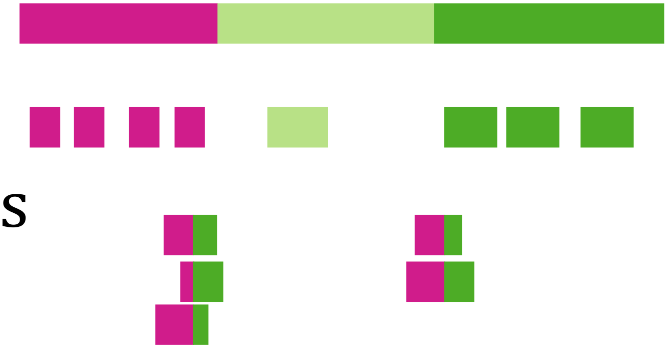
important

but would you know this ?

What does one need ?

Essential: reads covering the splice site

- basis of most isoforms in databanks



Computational considerations

- early description:

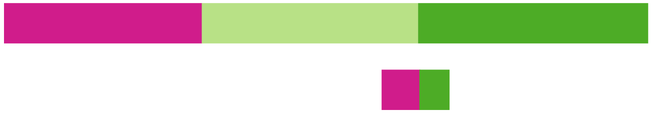
```
pick reference genome
for each fragment
        map to gene        fast
        add to counts
```

quick, but not helpful

finding splices

More complicated

```
pick reference genome
for each read
    if maps to genome
        do counts
    else            slow
        look for partial maps
        look for nearby mapping of other parts
```



Is this practical ?

- expensive, but can be done
- not routine

Limited splice searching

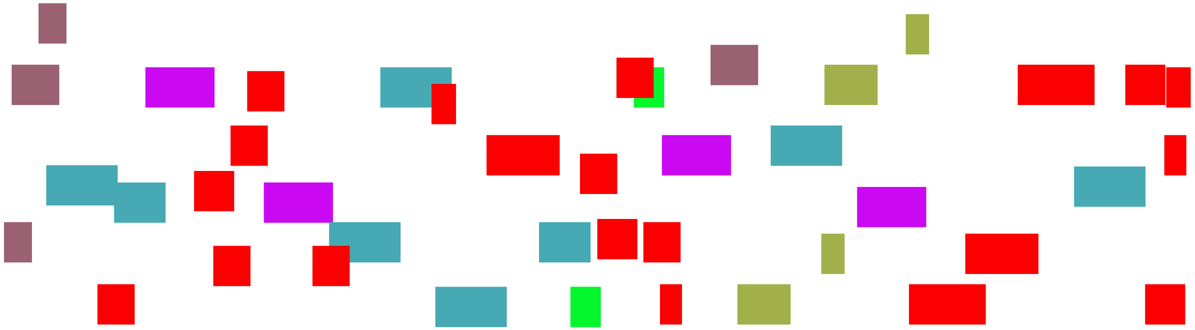
Maybe you are only interested in characterising one family

- from proteomics/other RNAseq you suspect that gene X behaves differently in some disease

```
pick reference genome
for each read
     if maps to gene X
          if maps to genome
               do counts
          else
               look for partial maps
               expensive careful search
```

- very practical
- assumes you have an idea what you are doing

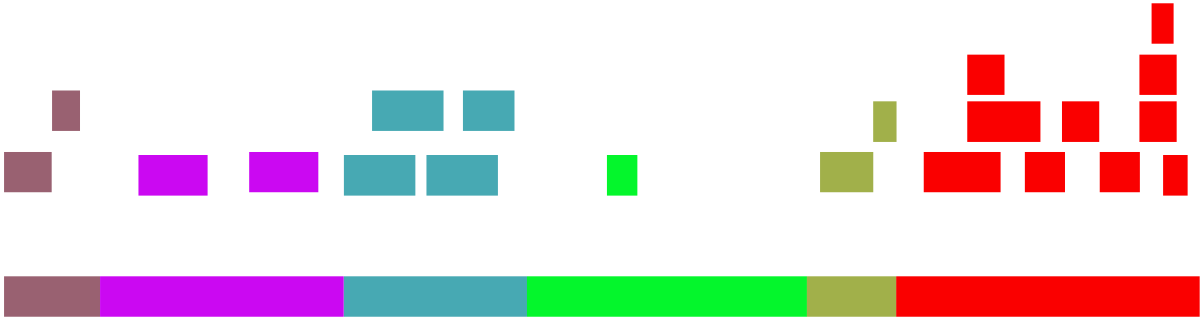
Do you need a reference genome ?



No

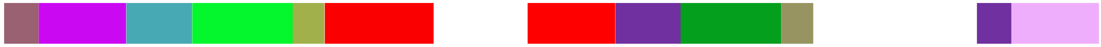
- If you have enough reads
    - lots of overlap…

Do you need a reference genome ?



Could treat the problem as genome assembly

- results will be very different
  - only get sequences that are translated to RNA



  - but this would be terrifc

The transcriptome

The transcriptome

Invaluable

- everything that is transcribed in an organism/tissue/sample
- If coverage is good enough
    - includes all splice products

Needs

- lots of coverage
- lots computer time

Problems

Most of problems of sequencing

- RNA (cDNA) may
  - not map to any place
    - errors variation
  - map to more than one place
    - short repetitive
    - similar sequences
      α- vs β-haemoglobin chains, dead / pseudo-genes
- you may not have a reference or good reference
  - use wrong bacteria strain
  - use monkey genome for man