3. New eukaryotic genomes sequencing, gene prediction; RNA seq/ Transcriptomics data analysis

Human, Mouse, Rat, Cow, Sheep, Cat, Dog, Pig, Chicken, Drosophila, Bee, Zebrafish, Fugu, Nematodes

Arabidopsis, Rice, Medicago, Soybean, Barley, Poplar, Tomato, Oat, Wheat, Corn

S.cerevisiae, S.pombe, Aspergillus nidulans,
Coprinus cinereus
Cryptococcus neoformans,
Fusarium graminearum
Magnaporthe grisea
Neurospora crassa
Ustilago maydis

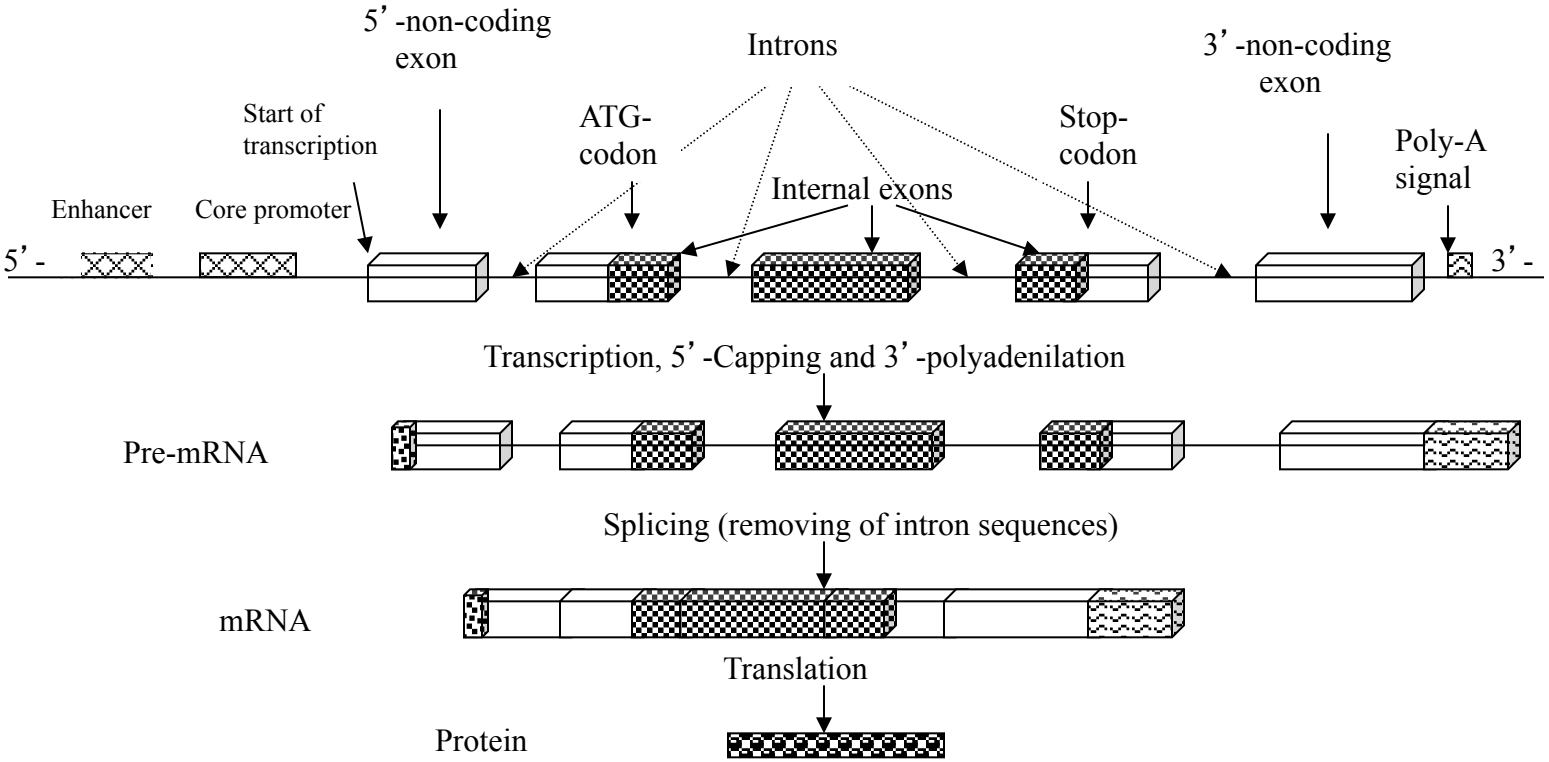
Computational gene finding in genomic DNA is a problem of central importance to molecular biology due to the lack of extensive experimental information for many organisms

Anopheles, P. falciparum, E. cuniculi, Chlamy, Ciona, Diatom, White rot, P. sojae

Victor Solovyev
Computer, Electrical and Mathematical Sciences and Engineering Division
KAUST, Saudi Arabia

The lecture 3 uses personal as well as publicly available WEB and publications materials

Expression stages and structural organization of typical eukaryotic protein-coding gene



The human fragile X mental retardation gene (HUMFMR1S) presents a typical example: 17 exons (40 – 60 bp long) occupy just 3% of 67,000 bp gene sequence.

the human pleiotrophin gene (HUMPLEIOT) includes a 1 bp exon and one of the alternative forms of the human folate receptor (HSU20391) gene contains a 3 bp exon.

Ab initio multiple gene prediction approaches using single genome sequence

Genescan **(Burge, Karlin,1997)**
HMMgene **(Krogh, 1977)**
Fgenesh **(Salamov, Solovyev,1998)**
Genie **(Reese et al., 2000)**
Augustus **(Sankem Waack, 2003)**
GenMarkHmm **(Besemer, Borodovsky, 2005)**

HMM: Likelihoods of gene components

Balanced score as production of likelihoods, simple probabilistic features

GeneID **(Guigo at al. 1992)**

Neural networks

Fgenes **(Solovyev,1997)**

Discriminant analysis

Flexible combinations of any discriminative features

Formal Definition of HMMs

- A hidden Markov model describes a sequence X of symbols and a path π of states:

X = (X1, X2,…,XL); π = (π1, π2,…, πL):

1. a finite set of states, Π
2. a finite set of symbols, S
3. transition probabilities between states:

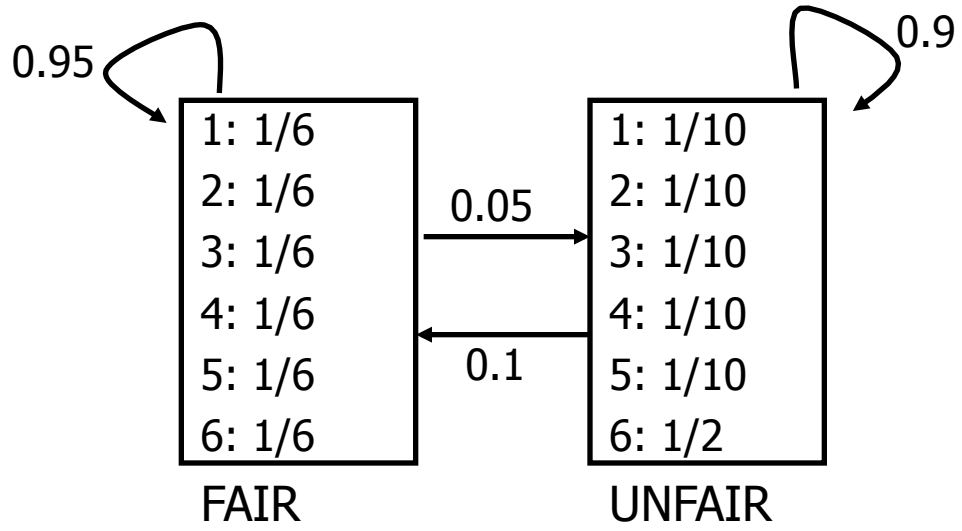
   $k, l \in \Pi : a_{kl} = P(\pi_i = l / \pi_{i-1} = k)$

4. emission probabilities

   $e_k(b) = P(Xi = b / \pi_i = k)$

Example – the dishonest casino

- In a casino, they use a fair die most of the time, but occasionally switch to an unfair die. The **switch between dice** can be represented by an HMM:



Dishonest casino - continued

- The symbols (observations) are the sequence of rolls:
  3 5 6 2 1 4 6 3 6…
- What is hidden?
  If the die is fair or unfair:
  f f f f u u u f f
  This is a Markov chain. Except for that, we have:
- Emission probabilities:
  Given a state, we have 6 possible symbols, each with an emission probability.

Joint probability of X and π

It is easy to derive the formula for the joint probability of a sequence X and a path π :
X = (X1, X 2,…,XL); π = (π1, π2,…, πL): The probability for Xi to be the emission from π_i is $e_{\pi_i}(x_i)$

The transition probability for given π_i it is followed by π_{i+1} is given by $a_{\pi_i \pi_{i+1}}$

- Let $a_{\pi 1}$ denote the probability for the path to start with π1. Then

$$P(x,\pi) = a_{\pi 1} \prod_{i=1}^{L} e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}}$$

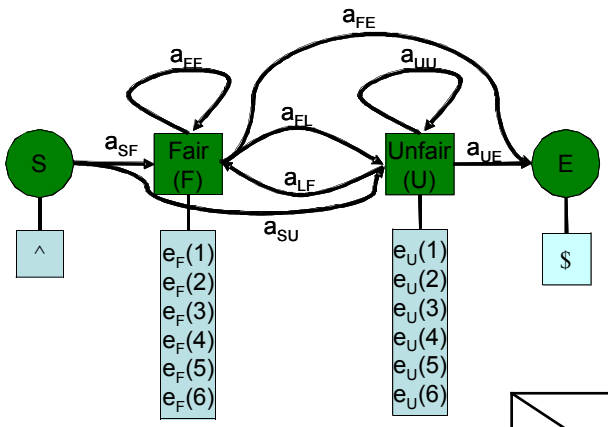
Hidden Markov Models

- Problem:
  - Path is hardly ever known
- Calculate:
  - Most Probable Path (Viterbi Algorithm)

Viterbi Algorithm

- Most probable path through an HMM
- Can be calculated recursively
- Implementation: Dynamic Programming
  - Initialization; Recursive Step; Trace-Back

Viterbi DP Matrix



	Sequence						
k \ *i*	^	5	3	1	5	6	$
(F)			$V_k(i)$				
(U)							

Viterbi Algorithm: Recursion

For sequence position i = 0,1,…,L+1:

For state l = 0,1,…,n:

$$V_l(i+1) = \max_k [V_k(i)\, a_{kl}]\, e_l(X_{i+1})$$

- $V_l(i+1)$: probability of best path ending in l at time $i+1$
- k: Come from state k
- $V_k(i)$: probability of best path ending in k at time i
- a_{kl}: transition to state l
- $e_l(X_{i+1})$: emit X_{i+1} in state l

Testing the Viterbi Algorithm

A sequence of 300 tosses of fair and loaded dice

```
Rolls   315116246446644245311321631164152133625144543631656626566666
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLL

Rolls   651166453132651245636664631636663162326455236266666625151631
Die     LLLLLLFFFFFFFFFFFLLLLLLLLLLLLLLLLFFLLLLLLLLLLLLLLFFFFFFFFF
Viterbi LLLLLLFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLFFFFFFFF

Rolls   222554416665665635643243641315134651463534111264146262533 56
Die     FFFFFFFLLLLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLL
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL

Rolls   366163666466232534413661661163252562462255265252266435353336
Die     LLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Viterbi LLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Rolls   233121625364414432335163243633665562466662632666612355245242
Die     FFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFF
Viterbi FFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLFFFFFFFFFF
```

Example of Decoding Problem

Have observation sequence ***O***, find state sequence ***Q***.

(1) Text Shakespeare (*s*) or monkey (*m*)

O = ..*aefjkuhrgnandshefoundhappinesssdmcamoe*…
Q = ..*mmmmmmsssssssssssssssssssssssssssmmmmmm*…

(2) Dice fair (*F*) or loaded (*L*) dice

O = …
13245564436636345566116345621661124536…
Q = …*LLLLLLLLLLLLFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLL*…

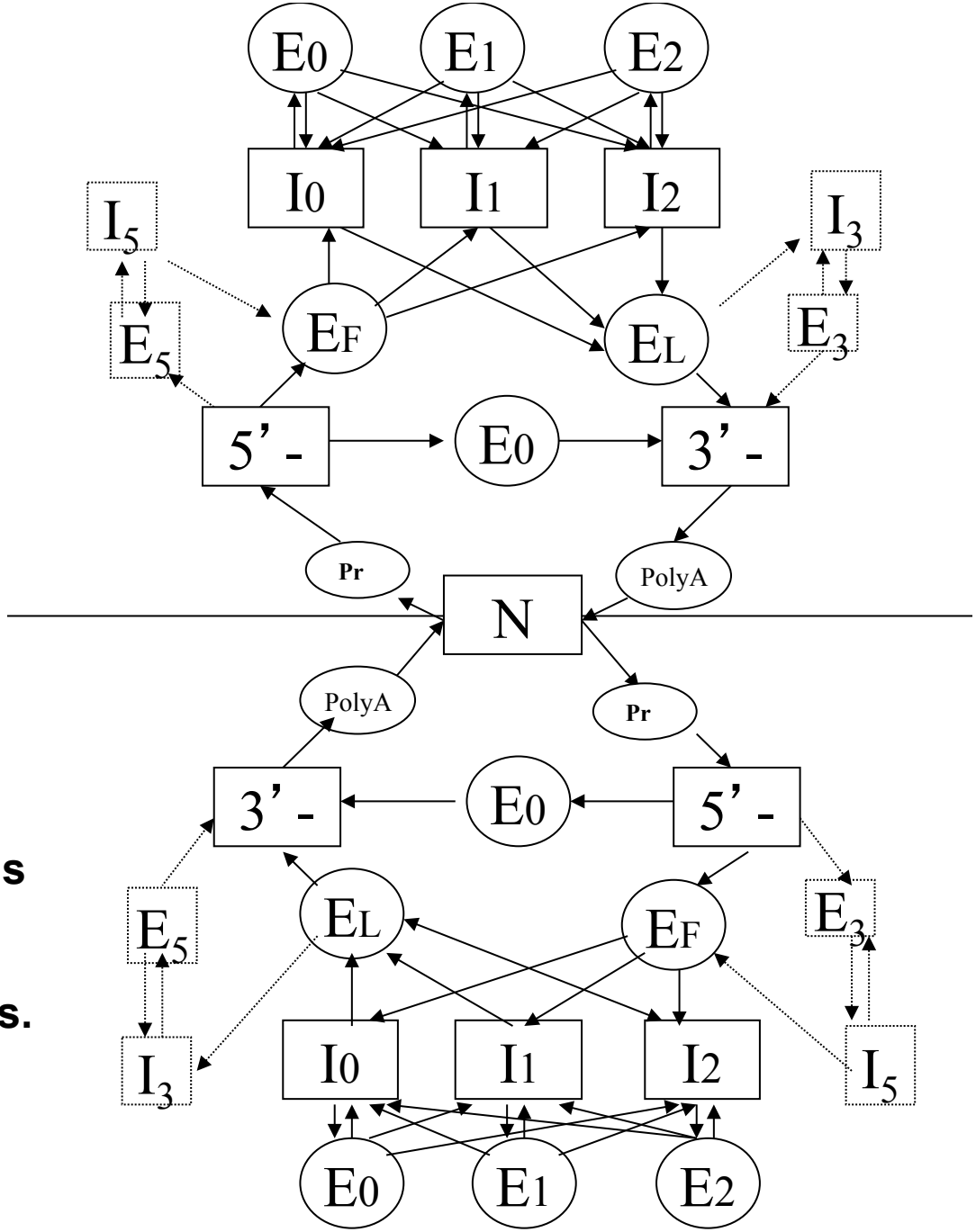
(3) DNA coding (*C*) or non-coding (*N*)

O = …*AACCTTCCGCGCAATATAGGTAACCCCCGG*…
Q = …*NNCCCCCCCCCCCCCCCCCCCCNNNNNNNNN*…

Hidden Markov model of multiple eukaryotic genes

Used in HMM based programs

E_i and I_i are different exon and intron states, respectively (*i*=0,1,2 reflect 3 possible different ORF). E 5/3 marks non-coding exons and I5/I3 are 5' - and 3' -introns adjacent to non-coding exons.



Gene prediction task:

- 27 states consist of 6 exon states (first, last, single and 3 types of internal exons due to 3 possible reading frames) and 7 non-coding states (3 intron, non-coding 5’- and 3’-, promoter and polyA) in each chain plus non-coding intergenic region.

Gene prediction task:

A gene structure can be considered as an ordered set of state/sub-sequence pairs, ϕ={(q1,x1),(q2,x2),…,(qk,xk)}, called the parse. We call the predicted gene structure such parse ϕ that the probability of generating X according to ϕ is maximal over all possible parses.

The parse probability

$$P(X,\pi) = P(q_1)\left(\prod_{i=1}^{k-1} P(x_i/l(x_i),q_i)P(l(x_i)/q_i)(P(q_{i+1},q_i)\right) P(x_k/l(x_k),q_k)P(l(x_k)/q_k)$$

where $P(q_1)$ denotes the initial state probability;

$P(x_i \mid l(x_i),q_i)P(l(x_i) \mid q_i)$ and $P(q_{i+1},q_i)$ are the independent joint probabilities of generation the subsequence x_i of length l in the state q_i and transitioning to q_{i+1} state.

$P(x_i \mid l(x_i),q_i)P(l(x_i) \mid q_i)$ is a production of a probability of generation l-length sequence x_i and the probability to observe such l-length sequence in the state q_i, which are computed using the sequence of x_i and the statistical data from a training set of known genes.

- Successive states of this HMM model are generated according to the Markov process with inclusion of explicit state duration density.
- The optimal parse is identified by a dynamic programming method called the Viterbi algorithm (Forney, 1973).
- The algorithm requires $o(N^2D^2L)$ calculations,

where N is the number of states, D is the longest duration and L is the sequence length (Rabiner, Juang, 1993).

(Speech recognition: Rabiner, 1989).

FGENESH

HMM-based gene structure prediction (multiple genes, both chains)

Paste nucleotide sequence here:

Alternatively, load a local file with sequence in Fasta format:

Local file name: Browse...

Organism: (•) Bos taurus ( ) Chicken ( ) Fish ( ) Frog (Xenopodinae) ( ) Human ( ) Mouse

( ) Anopheles gambiae ( ) Culex ( ) Drosophila ( ) Honey Bee ( ) Tribolium (red flour beetle)

( ) Brugia malayi (parasitic nematode) ( ) C.elegans ( ) Sea urchin

( ) Diatom ( ) Plasmodium falciparum ( ) Phytophthora

( ) Dicot plants (Arabidopsis) ( ) Medicago (legume plant) ( ) Monocot plants (Corn, Rice, Wheat, Barley)
( ) Tomato ( ) Vitis vinifera

( ) Chlamydomonas (single celled green algae)

( ) Aspergillus ( ) Batrachochytrium ( ) Botrytis ( ) Coccidioides immitis ( ) Coprinopsis cinerea ( ) Cryp
( ) Fusarium graminearum ( ) Histoplazma (fungus) ( ) Magnaporthe ( ) Neurospora crassa
( ) Phanerochaete chrysosporium (white rot) ( ) Rhizopus_oryzae ( ) Schizosaccharomyces pombe ( ) S
( ) Stagnospora nodorum ( ) Uncinocarpus reesii ( ) Ustilago

SEARCH RESET

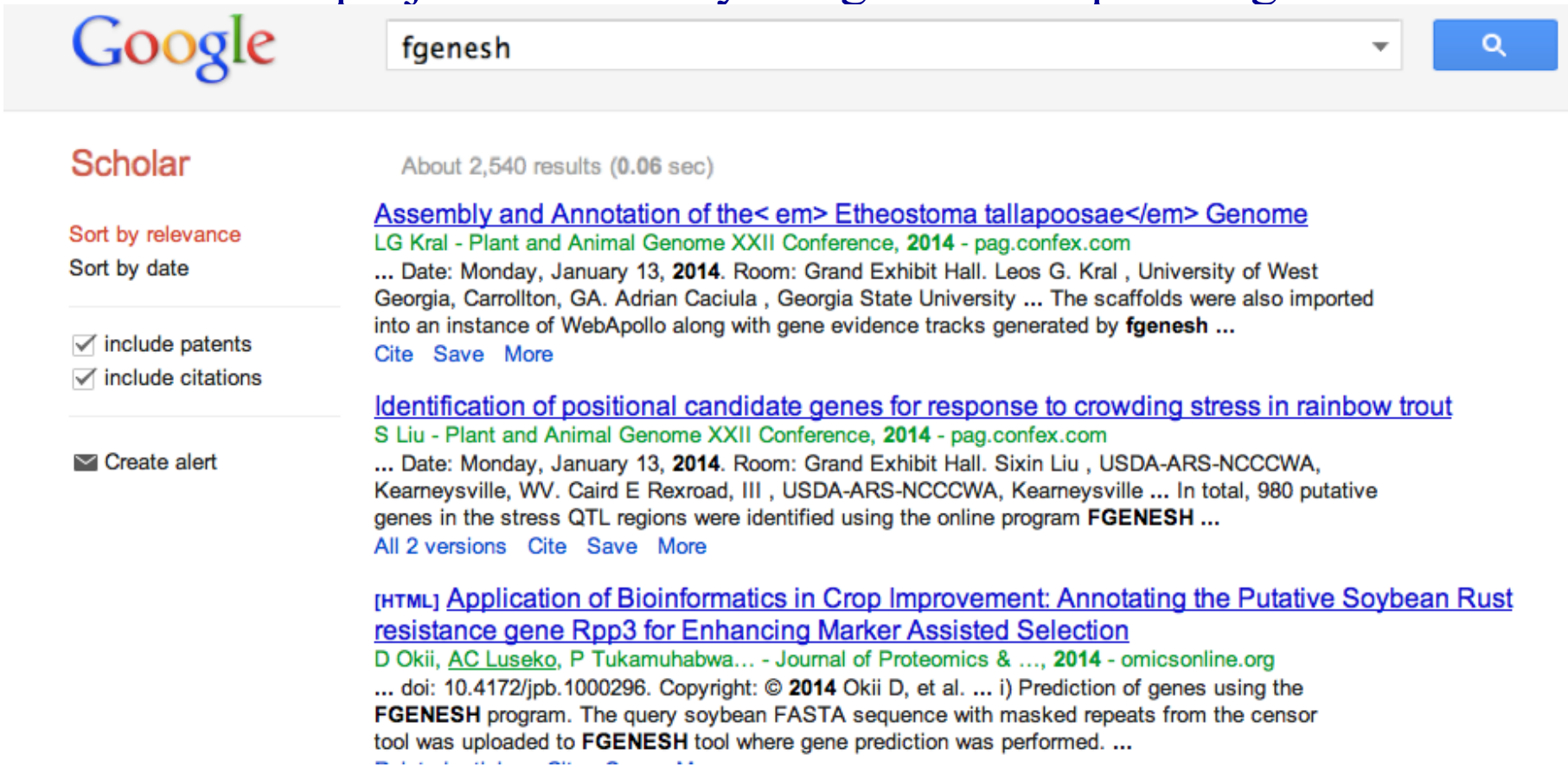
Show picture of predicted genes in PDF file

FGENESH 2.5 Prediction of potential genes in Homo_sapiens genomic DNA
Time : Sun Feb 25 09:58:3 9 2007
Seq name: 0
Length of sequence: 13903
Number of predicted genes 1 in +chain 0 in -chain 1
Number of predicted exons 9 in +chain 0 in -chain 9
Positions of predicted genes and exons: Variant 1 from 1,Score:27.782177

G	Str	Feature	Sta	rt	End	Score	ORF			Len
1	-	PolA	18			-5.68				
1	-	1 CDSl	151	-	222	6.45	151	-	222	72
1	-	2 CDSi	477	-	575	0.18	477	-	575	99
1	-	3 CDSi	1350	-	1415	5.34	1350	-	1415	66
1	-	4 CDSi	2238	-	2311	3.81	2238	-	2309	72
1	-	5 CDSi	2782	-	2950	9.34	2783	-	2950	168
1	-	6 CDSi	4127	-	4283	9.00	4127	-	4282	156
1	-	7 CDSi	4980	-	5166	7.86	4982	-	5164	183
1	-	8 CDSi	9808	-	9946	-0.90	9809	-	9946	138
1	-	9 CDSf	10759	-	10761	5.00	10759	-	10761	3
1	-	TSS	11307			-6.29				

Predicted protein(s):
>FGENESH: 1 9 exon (s) 151 - 10761 321 aa, chain -
MNPPTDPHPSLVPVTAALAFRPCQLLQALIKEASVHGVRLRGGFWEEGLLECCARCLVGA
PFASLVATGLCFFGVALFCGCGHEALTGTEKLIETYFSKNYQDYEYLI NVIHAFQYVIYG
TASFFFLYGALLLAEGFYTTGAVRQIFGDYKTTICGKGLSATFVGITYALTVVWLLVFAC
SAVPVYIYFNTWTTCQSIAFPSKTSASIGSLCADARMYGVLPWNAFPGKVCGSNLLSICK
TAEFQMTFHLFIAAFVGAAATLVSLQAPYDSKSLGHIDVAKPNIVHFPEENSVLDQTELT
FMIAATYNFAVLKLMGRGTKF

Fgenesh/Fgenesh++ pipline applied in ~2500 published research projects on eukaryotic genome sequencing



Plant Molecular Biology (2005), 57, 3, 445-460:

"Five *ab initio* programs (FGENESH, GeneMark.hmm, GENSCAN, GlimmerR and Grail) were evaluated for their accuracy in predicting maize genes. FGENESH **yielded the most accurate** and GeneMark.hmm the second most accurate predictions" (FGENESH identified 11% more correct gene models than GeneMark on a set of 1353 test genes).

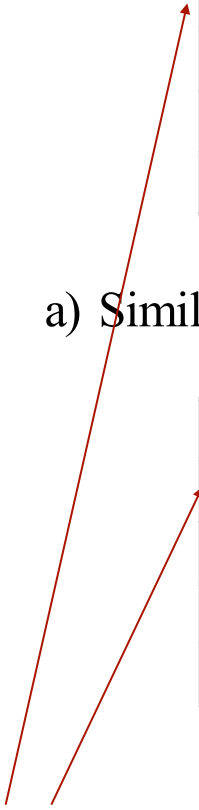
Accuracy of human gene prediction using similar Mouse or Drosophila proteins.

a) Similarity of mouse protein > 90% in 921 sequences *)

	Sn ex	Sno ex	Sp ex	Sn nuc	Sp nuc	CC	%CG
Fgenesh	86.2	91.7	88.6	93.9	93.4	0.9334	34
Genwise	93.9	97.6	95.9	99.0	99.6	0.9926	66
Fgenesh+	97.3	98.9	98.0	99.1	99.6	0.9936	81
Prot_map	95.9	98.3	96.9	99.1	99.5	0.9924	73

a) Similarity of Drosophila protein > 80% - 66 sequences

	Sn ex	Sno ex	Sp ex	Sn nuc	Sp nuc	CC	CG%
Fgenesh	90.5	93.8	95.1	97.9	96.9	0.950	55
Genewise	79.3	83.9	86.8	97.3	99.5	0.985	23
Fgenesh+	95.1	97.8	97.0	98.9	99.5	0.9914	70
Prot_map	86.4	95.3	88.1	97.6	99.0	0.982	41



Ab initio

Prot_map example of alignment

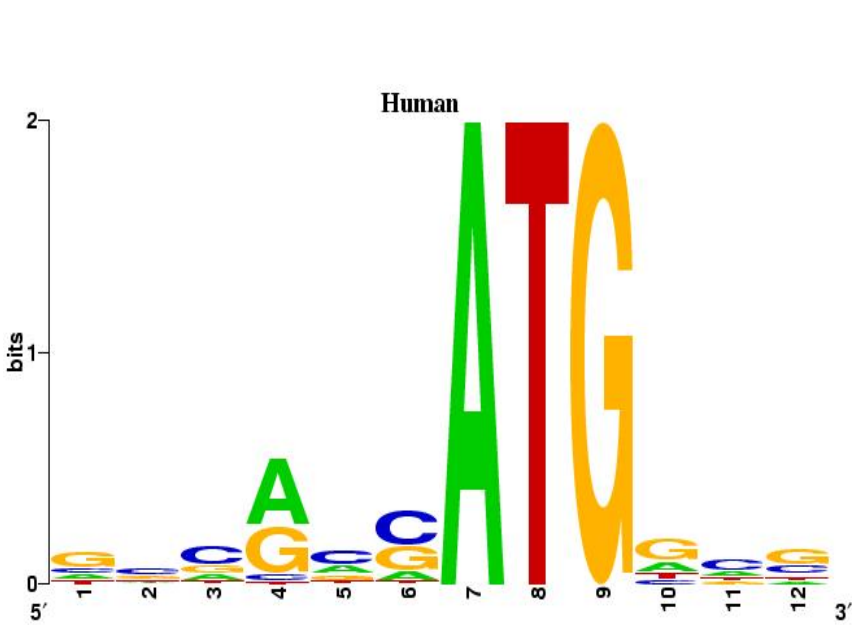
```
     1         11    2146713    2146723    2146739    2146769
       gatcacagaggctgg(..)agtgtctgtgtttca?[GGRIVSSKPFAPLNFRINSRNLSg
       ..............(..)evdhqlkerfanmke  GGRIVSSKPFAPLNFRINSRNLS-
     248        248        249        259        267        277

2146797    2146806    2147558    2147568    2147581    2147611
       ]gtaagaaactctcat(..)ctgtggctcctgcag[acIGTIMRVVELSPLKGSVSWTGK
        ---------------(..)--------------- -dIGTIMRVVELSPLKGSVSWTGK
     286        286        286        286        289        299

2147641    2147671    2147686    2148919    2148926    2148937
       PVSYYLHTIDRTI]gtgagtatctcgctg(..)ctttcttcttttag[LENYFSSLKNP
       PVSYYLHTIDRTI ---------------(..)--------------- LENYFSSLKNP
     309        319        322        322        322        323

2148967    2148982    2150384    2150391    2150402    2150432
       KLR]gtaagtttgtgtgtt(..)ctgctctccttccag[EEQEAARRRQQRESKSNAATP
       KLR ---------------(..)--------------- EEQEAARRRQQRESKSNAATP
     333        336        336        336        337        347

2150462    2150492    2150513    2150523    2150609    2150619
       TKGPEGKVAGPADAPM]gtaaggccccagcct(..)ccttgtgtcctccag[DSGAEEEK
       TKGPEGKVAGPADAPM ---------------(..)--------------- DSGAEEEK
     357        367        373        373        373        373
```

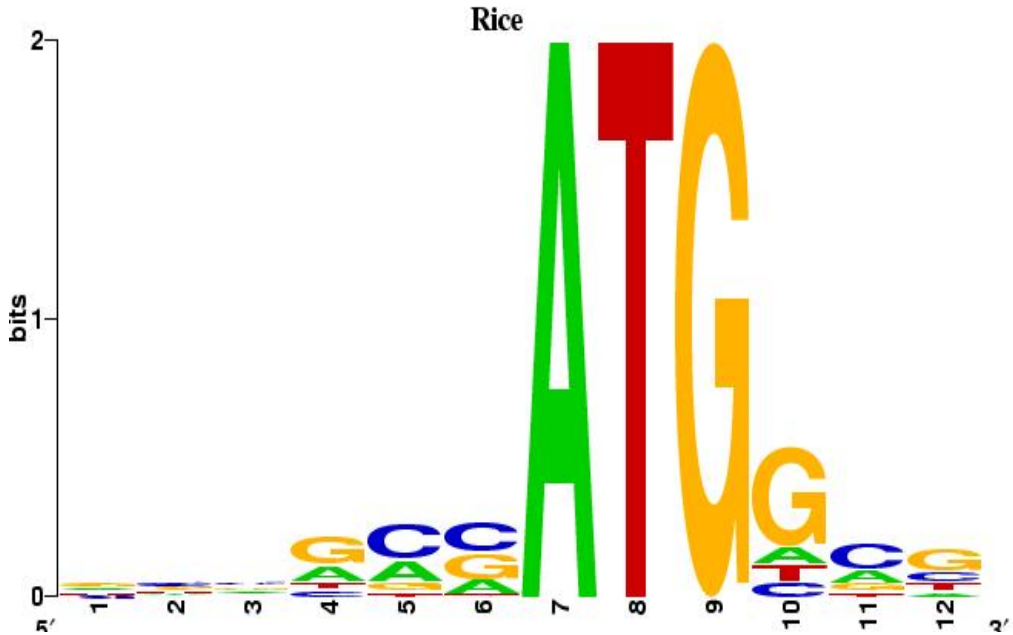
FGENESH++: AUTOMATIC EUKARYOTIC GENOME ANNOTATION PIPELINE

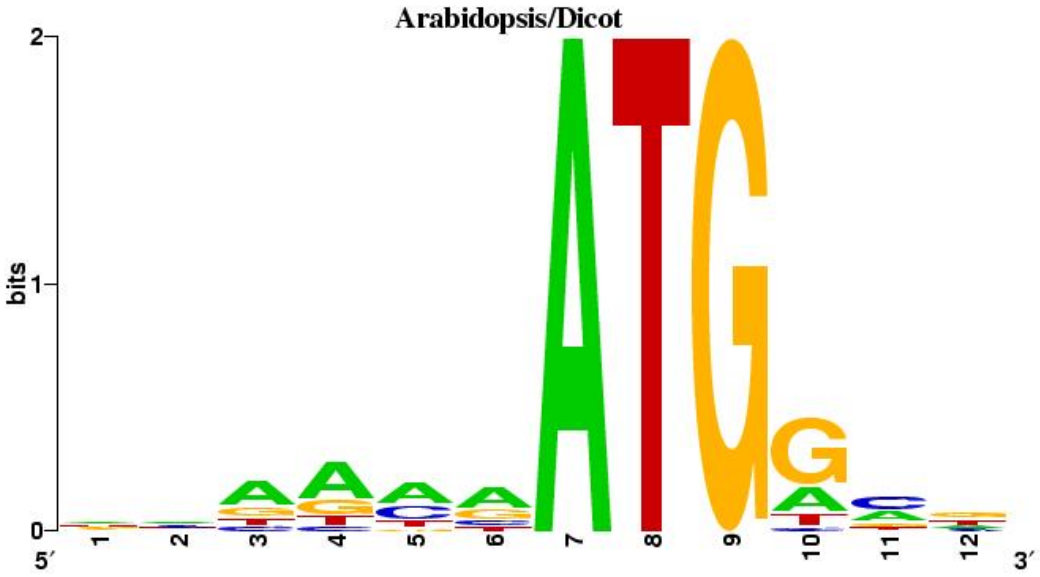
1. **RefSeq mRNA mapping by *Est_map* program - mapped genes are excluded from further gene prediction process.**
2. **Map all known proteins (NR) on genome by *Prot_map* program with gene structure reconstruction (find regions occupied by genes)**
3. **Run *Fgenesh+* using mapped proteins and selected genome sequences**
4. **Run ab initio *Fgenesh* HMM gene prediction on the rest of genome.**
5. **Run of *Fgenesh* gene predictions in large introns of known and predicted genes.**

Fgenesh++ can use NGS data such as Transcripts and RNASeq reads mapping information on splice sites positions

Organism specific signal differences: start of translation







Developed organism-specific parameters for Fgenesh group of programs:

Totally: 128 eukaryotic organisms

- **Human, Mouse, Cow, Drosophila, Bee, Tribolium, C. elegans, Frog, Fish (WUSTL, Baylor, CSHL, JGI)**
- **Dicots (Arabidopsis), Nicotiana tabacum, Tomato, Grape; Monocots (Corn, Rice, Wheat, Barley) (TIGR, Rutgers University)**

  Medicago (University of Minnesota)
- **Schizosaccharomyces pombe, Neurospora crassa,Aspergillus nidulans, Coprinus cinereus, Cryptococcus neoformans, Fusarium graminearum, Magnaporthe grisea, Ustilago maydis, Histoplasma, Coccidioides immitis, Rhizopus_oryzae, Sclerotinia sclerotiorum, Stagnosporam nodorum, Uncinocarpus reesii (MIT/Broad Institute), Brugie malayi (TIGR)**
- Chlamydomonas (single celled green algae), Dictyostelium discoideum (amoeba), Entamoeba histolytica, Giardia lamblia,Guillardia theta, Hyaloperonospora arabidopsidis, Leischmania major, Phaeodactylum tricornutum, Plasmodium falciparum, Toxoplasma gondii, Trypanosoma_brucei

Velasco R. et al (2007) A High Quality Draft Consensus Sequence of the Genome of a Heterozygous Grapevine Variety. ***PLoS ONE*** 2(12): e1326.

all'Adige (IASMA) in Trentino, Italy, announced that they were almost done sequencing the genome of a Pinot Noir grape used in many countries to make red and sparkling wines. Velasco had been involved in

Sweet finish. Riccardo Velasco samples wine from the grape he raced to sequence.

first fleshy fruit and
g plant to have its

plintered into rival
rt sequencing was
ess has brought both

Wine woes. Powdery mildew (*above*) and other fungal diseases can devastate vineyards.

A key motivation for deciphering the grape genome is to prevent a repeat of the economic devastation that struck the European wine industry in the late 1800s. At that time, phylloxera, sap-sucking insects from North America, ravaged European grapevines. Today, winemakers and grape researchers are struggling to combat new threats, particularly downy and powdery mildew, diseases that have made their way to Europe from the United States over the past century. These fungi are an environmental as well as an economic nightmare: Although only about 5% of Europe's farmland is dedicated to wine vineyards, they account for about 70% of the region's fungicide use.

Draft genome sequence of the oilseed species ***Ricinus communis***

Nature Biotechnology 28, 951–956 (2010)

J. Craig Venter Institute (JCVI), United States Department of Agriculture

Castor bean **is a highly valued oilseed crop for lubricant, cosmetic, medical and specialty chemical applications.** It has also been proposed as a **potential source of biodiesel.**

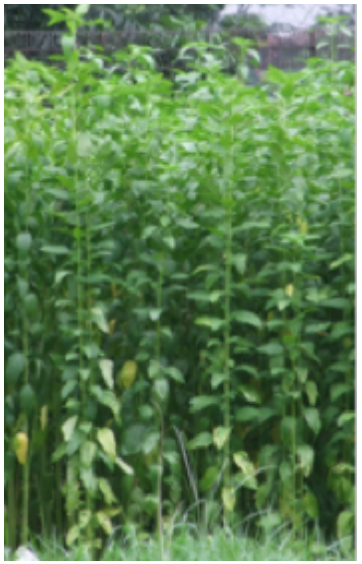
Fgenesh++ pipeline used to identify genes in these NGS projects

Rubber tree (*Hevea brasiliensis*) genome

The genome information will enable researchers **to understand genetic characteristics of different breeds of rubber trees**

Jute Genome Project

A major trait that needs to be manipulated for jute is its fiber length and fiber quality.



Many gene variants are completely absent in genomic sequence annotations

- Non canonical splice sites
- Alternatively spliced genes
- Alternative promoters
- Alternative poly-A

While a decade ago, alternative splicing of a gene was considered unusual. It turns out that it's a nearly universal feature of human genes.

Report of total cell mRNA sequencing to investigate alternative splicing in more than a dozen human tissue and cell lines (*Nature*, 2011) indicates that 92-94%of human genes undergo alternative splicing, 86% with a minor isoform frequency of 15% or more.

This new genes/gene variants can be discovered from RNASeq NGS data

NCBI

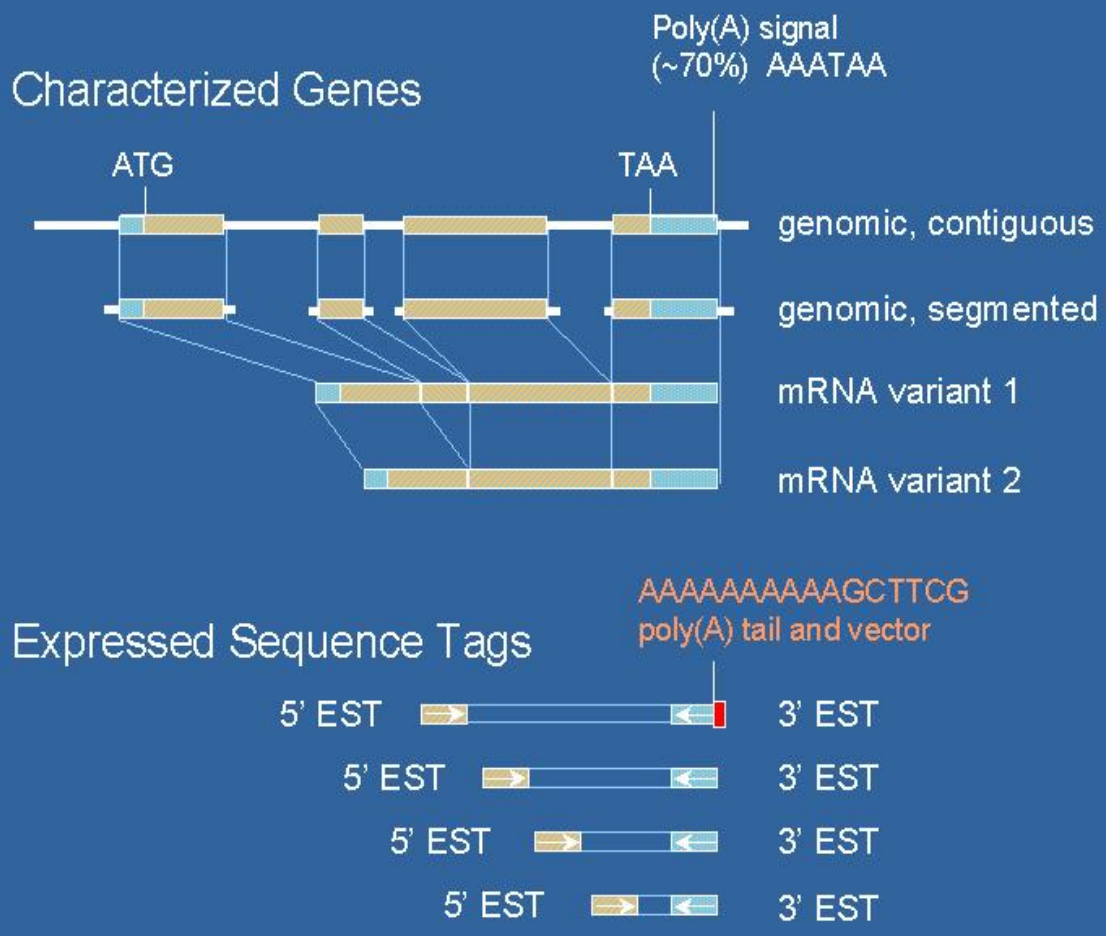
One Gene, Many Sequences, One Cluster

GenBank is an archive of published sequences

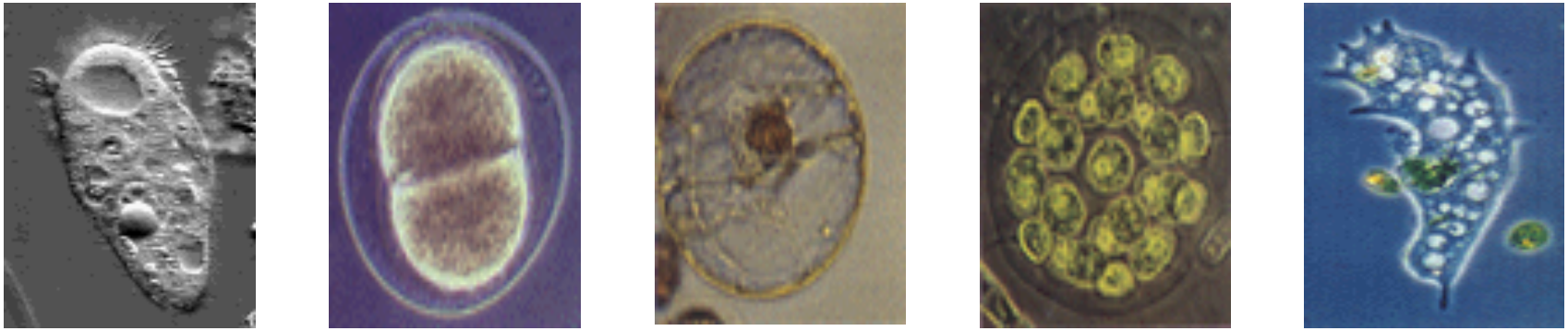
May be many representatives of a given gene

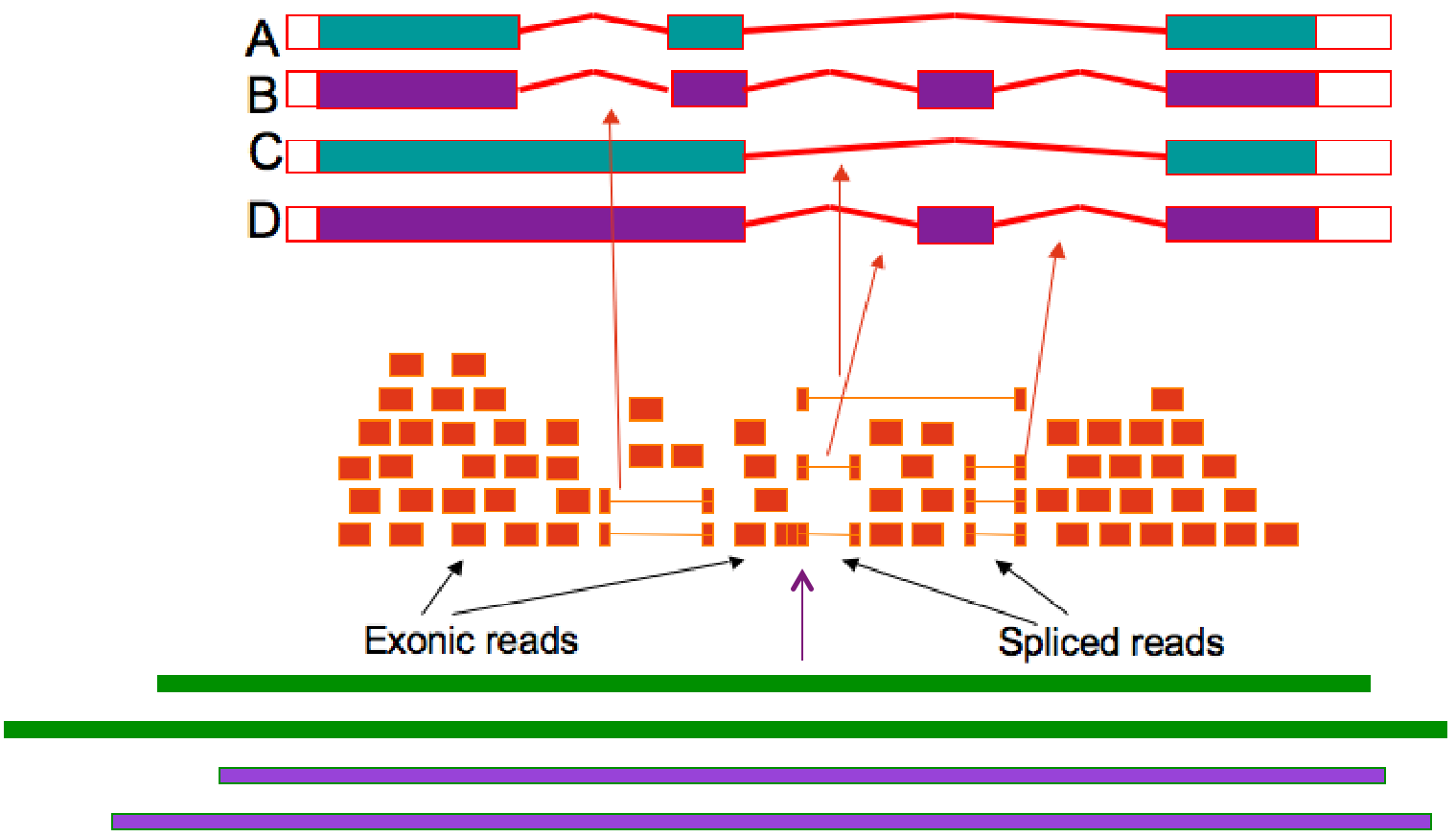
UniGene is an automated system for cataloging putative gene sequences

Goal is one cluster per gene, including alternate splice forms



RNA-Seq: Whole Transcriptome Sequencing





RNASeq can be used to reveal **tissue-specific alternative splicing**, **novel genes** and transcripts and **genomic structural variations.**

As many genes have **multiple isoforms**, many of which share exons, and many genes families have **close paralogs**, some reads cannot be assigned unequivocally to a transcript.

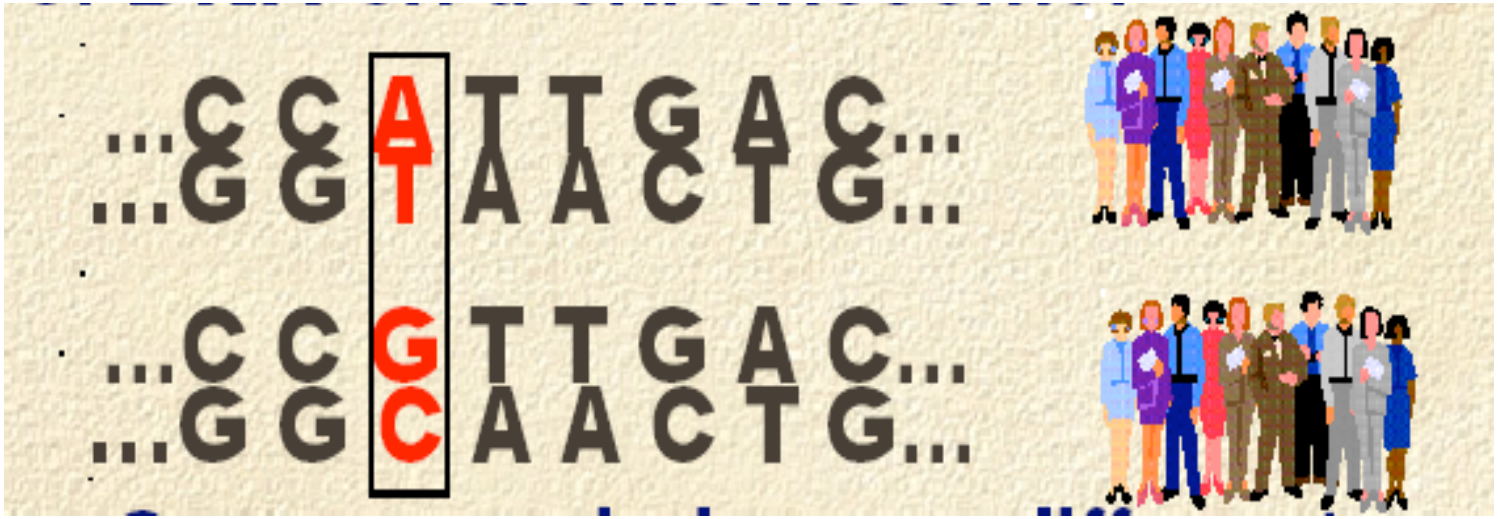
The analysis of RNA-Seq data presents major challenges
in transcript assembly and abundance estimation,
arising from the ambiguous assignment of reads to isoforms

These computational challenges fall into three main categories:

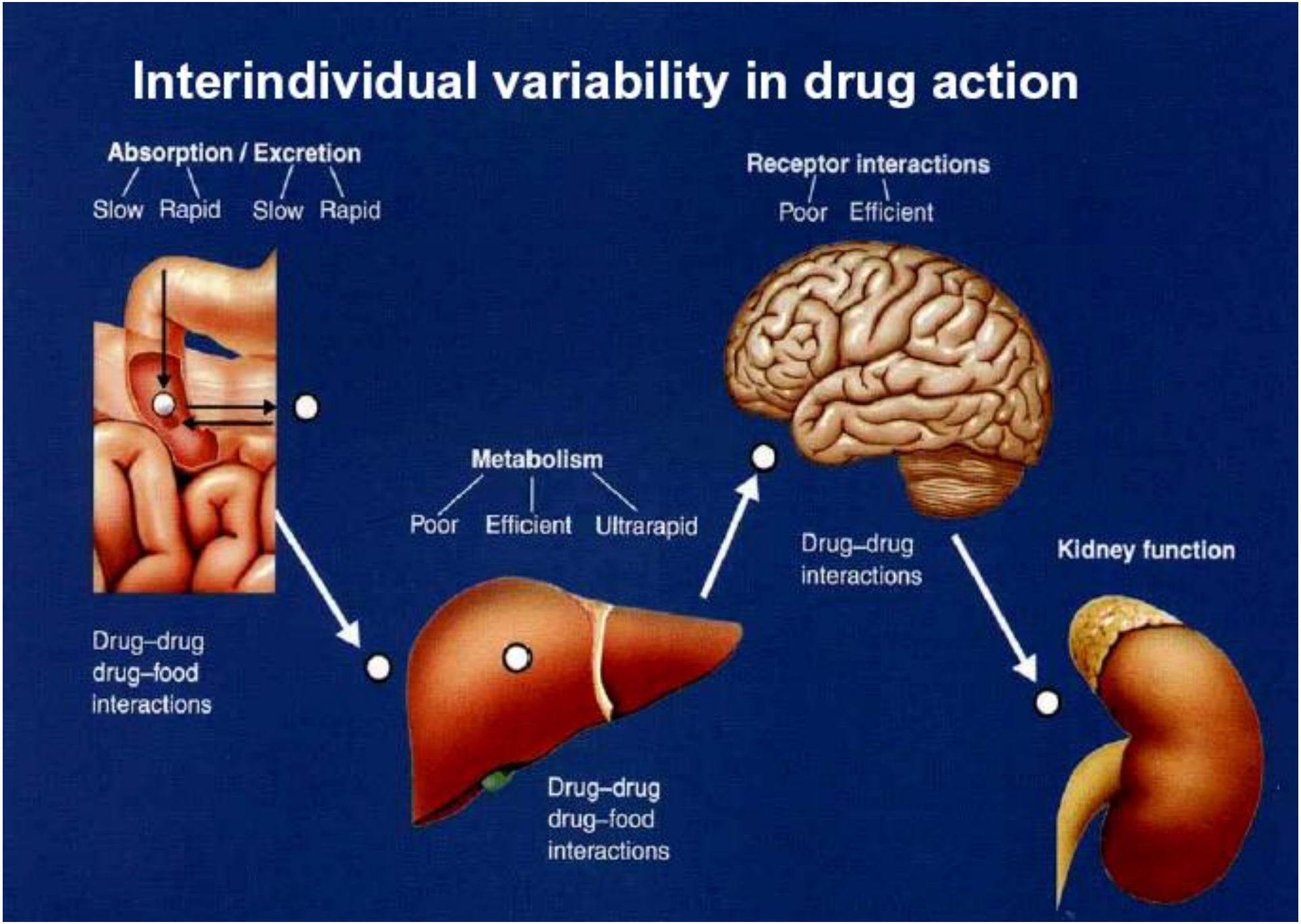
(i) read mapping,
(ii) transcriptome reconstruction and
(iii) expression quantification.

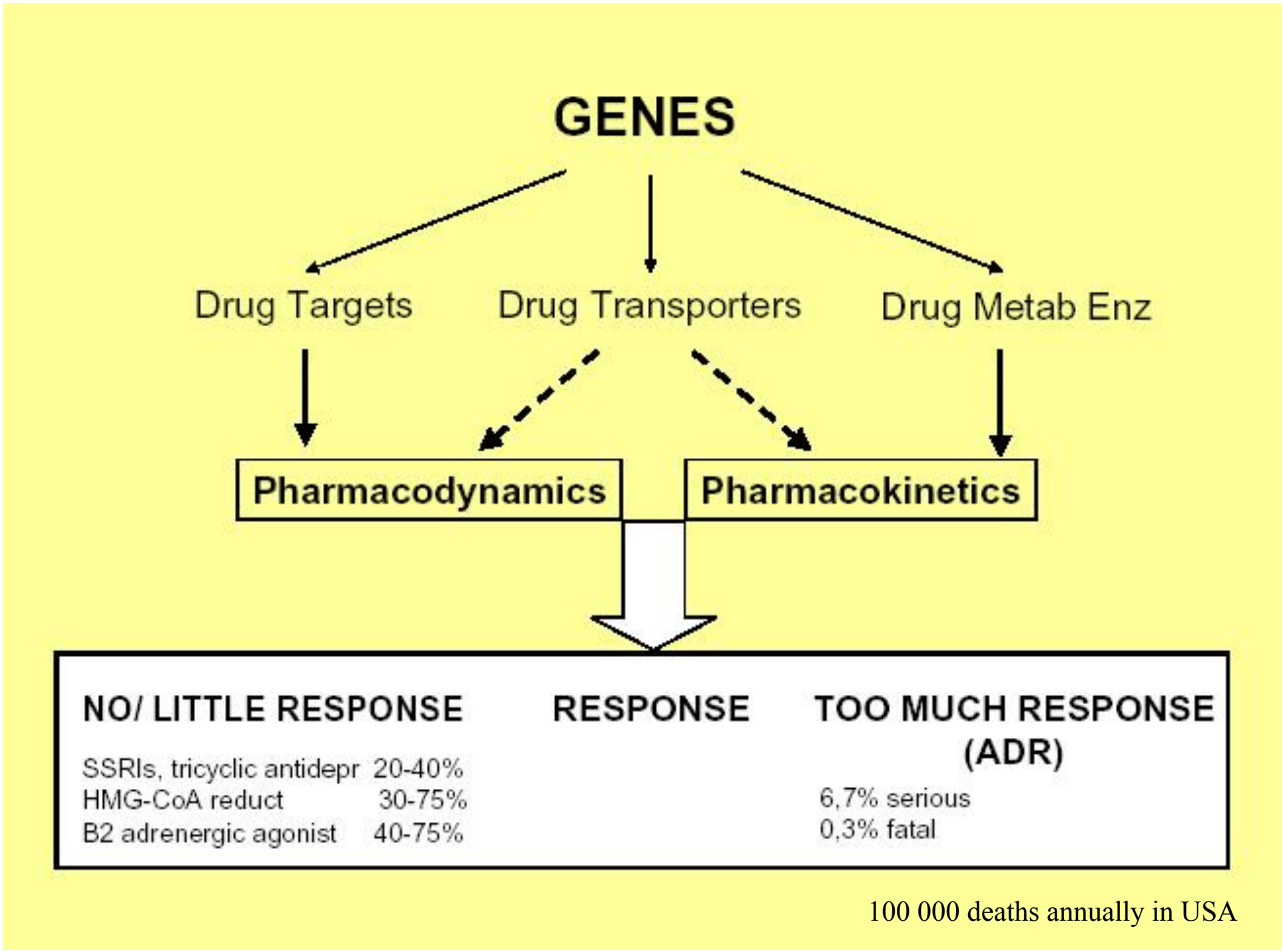
Single Nucleotide Polymorphism

- Occurrence: once in every 300-1000 bases.
- SNPs (“snips”): Naturally occurring variants that affect a single nucleotide.
- SNPs are responsible e.g. for hair colour, but are also the reason for individual differences in respons to drugs.



Interindividual variability in drug action

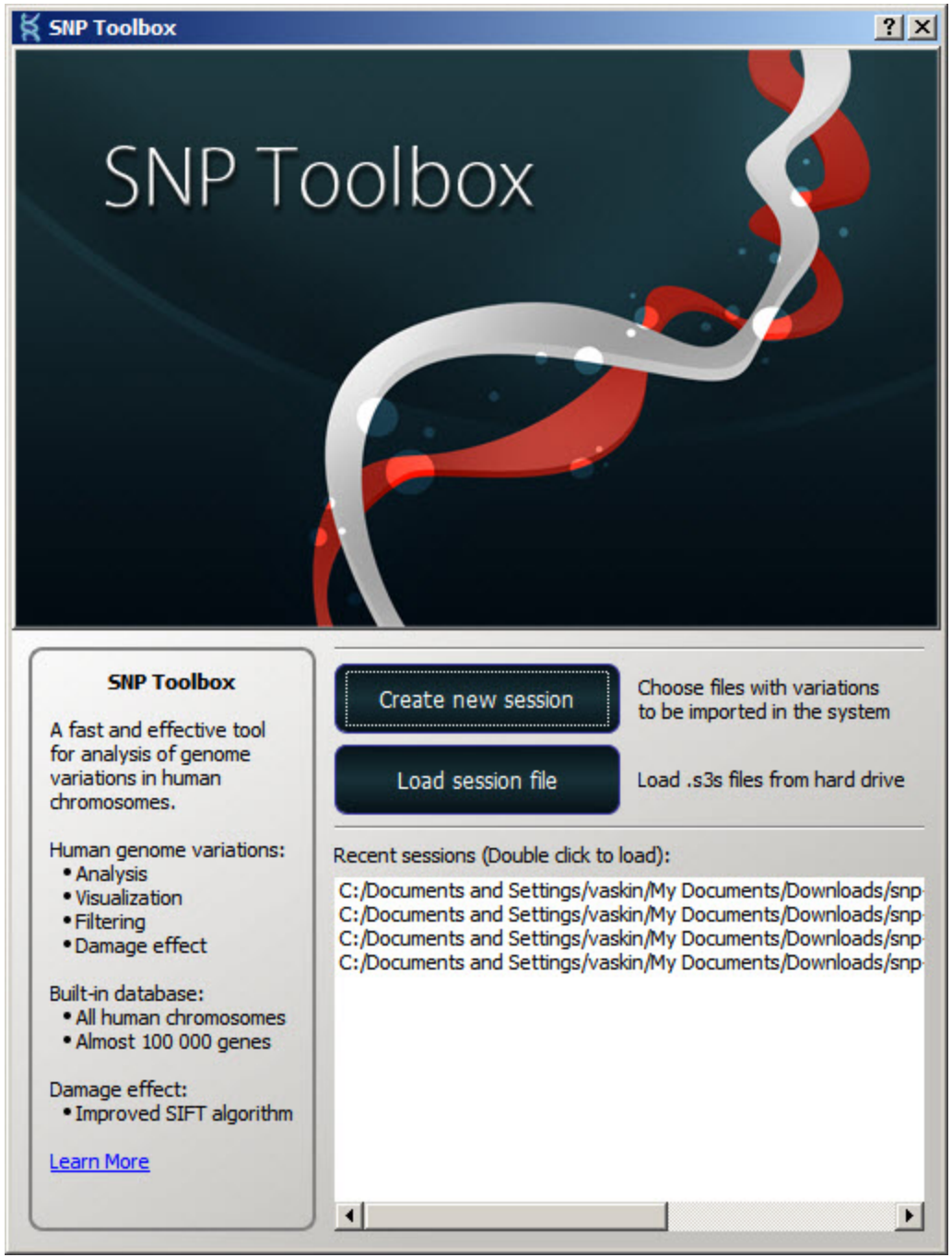




100 000 deaths annually in USA

1000 Genomes Project

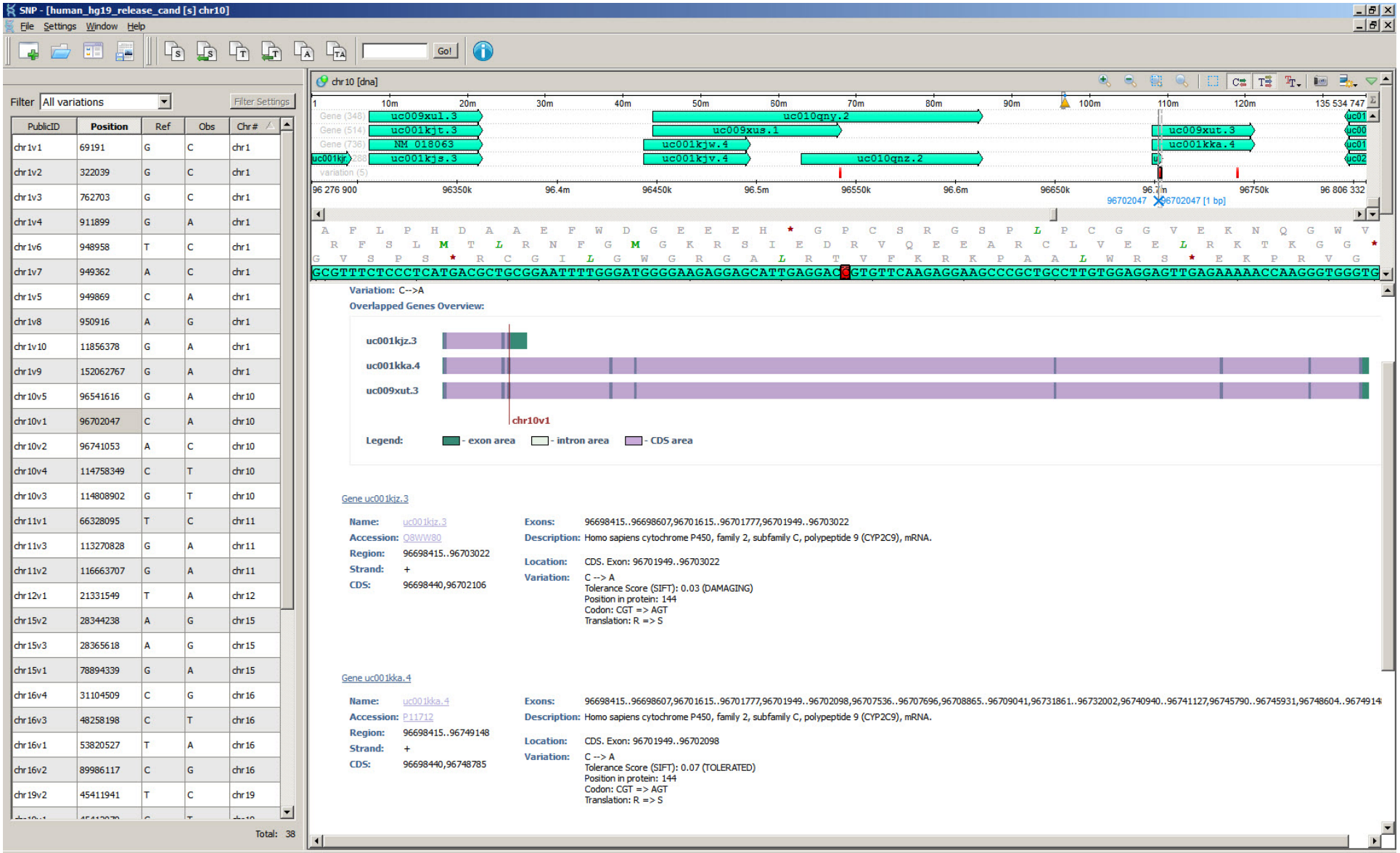
Enzyme → Characterization of enzyme → **Prediction of drug response**



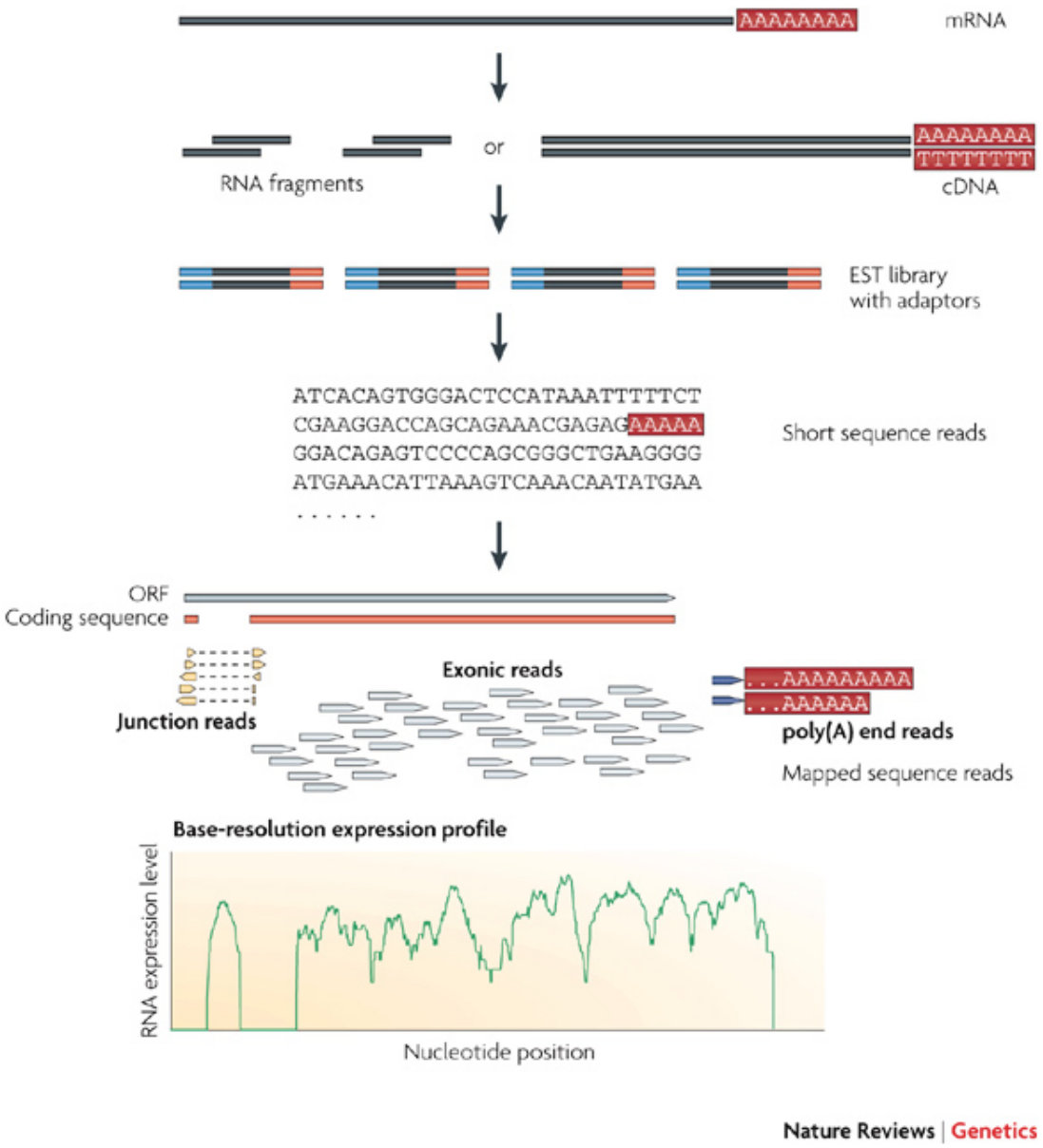
SNP discovery and their effect analysis

```
ATTTTATATTACATTAACAAGCTAATTTGCA
|||||||||||||||||||||||||||||
889898998884888988888889889888
ATTTTATATTACATTAACAAGCTAATTTGCA
ATTTTATATTACATTAACAAGCTAA......
ATTTTATATTACATTAACAAGCTNA......
ATTTTATATTACATTAACAANCTAA......
ATTTTATATTATATTAACAAGCTAA......
ATTTTATATTACATTNNCANNNNAA......
NTTTTATATTACATTAACNNGCTAA......
ATTTTATATTATATTAACAAGCNNN......
NTTTTATATTNCATTAACAAGCTNA......
ANNTTATATTATATTAACAAGCTAA......
ATTTTATATTATATTAACAANNTNA......
NTTTTATATTATATTAACAAGNTNN......
ATTTTATATTACATTAACAAGCTAAT.....
ATTTTATATTACATTAACNAGCTNNT.....
NNTTTATATTATATTAACAAGCTAAT.....
ATTTTATATTACNTTAACAAGCTNNT.....
ATTTTATATTANATTAACAANCTAAN.....
ATTTTATATTATATTAACAANCTAAT.....
ATTTTATATTACATTAACAAGCTAATT....
ATTTTATATTACATTAACAAGCTAATT....
ANNTTATATTACATTAACAAGCTAATT....
ATTTTATATTACATTAACAAGCNAATT....
NTTTTANATTACATTAACAAGCTAATT....
ATTTTATATTATATTAACAAGCTAATT....
ATTTTATATTATATTAACAAGCTAATT....
```

SNP Toolbox: to analyze and select SNPs with given characteristics genome group or or disease-specific



How RNA-seq works



Sample preparation

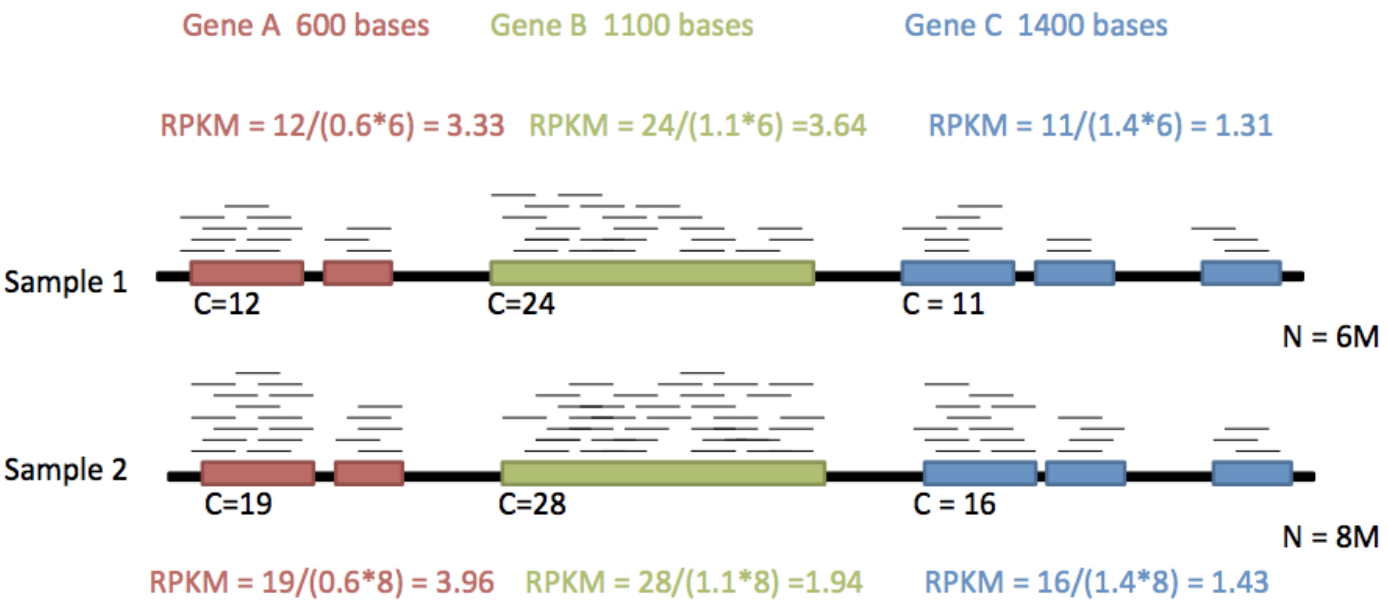
Next generation sequencing (NGS)

Data analysis:
- ✓Mapping reads
- ✓Visualization (Gbrowser)
- ✓De novo assembly
- ✓Quantification

Figure from Wang et. al, **RNA-Seq: a revolutionary tool for transcriptomics,** Nat. Rev. Genetics 10, 57-63, 2009)**.**

How do I quantify expression from RNA-seq?

RPKM: Reads per Kb million (Mortazavi et al. Nature Methods 2008)

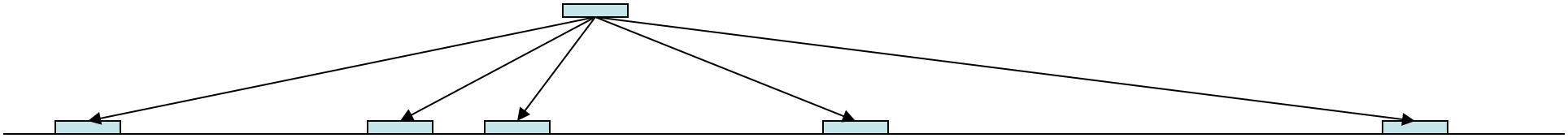


Longer and more highly expressed transcripts are more likely be represented among RNA-seq reads

RPKM normalizes by transcript length and the total number of reads captured and mapped in the experiment

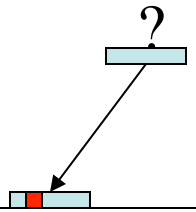
Sequencing depth can alter RPKM values

Multiple mapping



- A single tag may occur more than once in the reference genome.
- The user may choose to ignore tags that appear more than *n* times.
- As *n* gets large, you get more data, but also more noise in the data.

Inexact matching



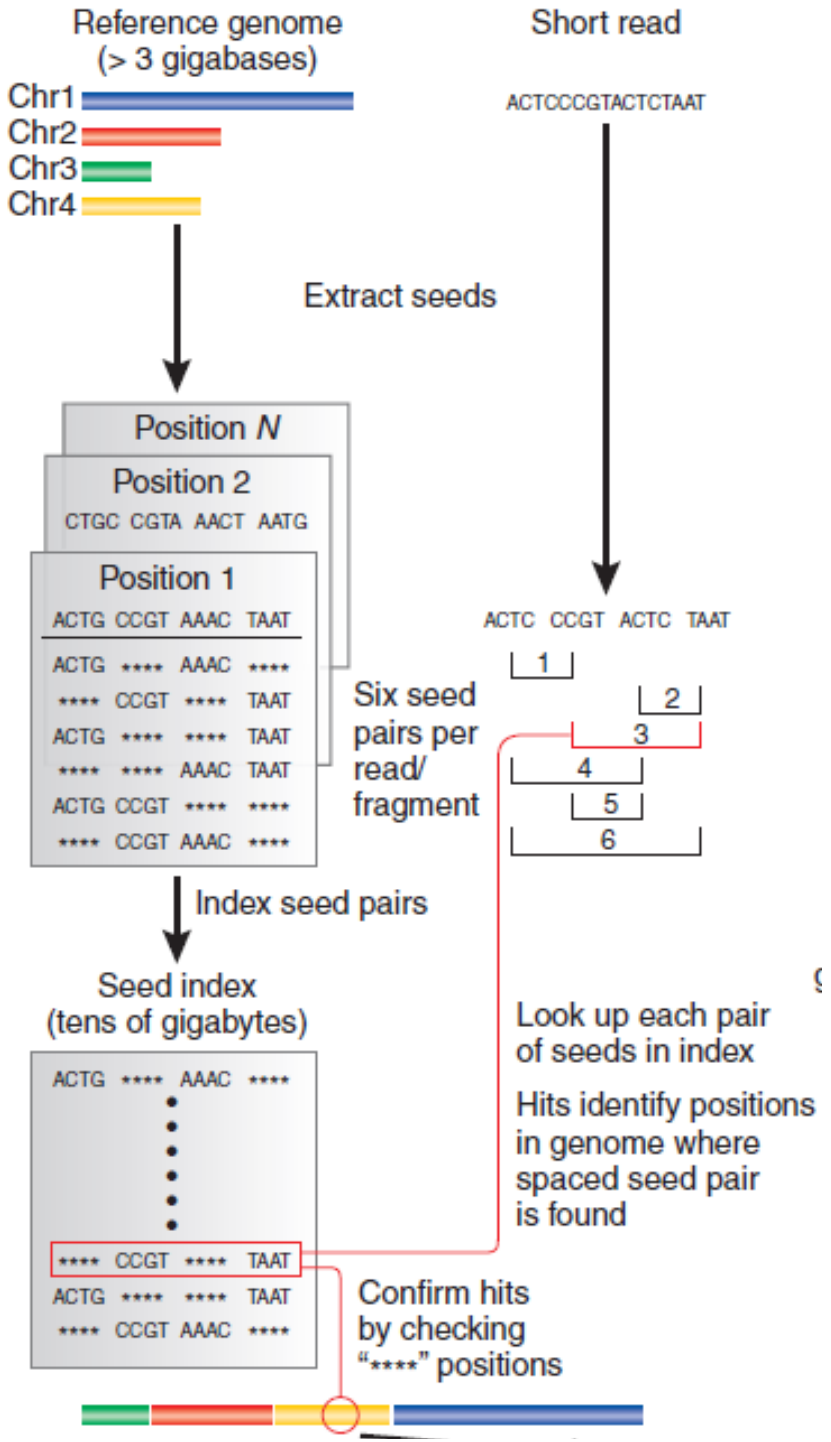
- An observed tag may not exactly match any position in the reference genome.
- Sometimes, the tag *almost* matches one or more positions.
- Such mismatches may represent a SNP (single-nucleotide polymorphism, see wikipedia) or a bad read-out.
- The user can specify the maximum number of mismatches, or a phred-style quality score threshold.
- As the number of allowed mismatches goes up, the number of mapped tags increases, but so does the number of incorrectly mapped tags.

Mapping Reads to genomic sequence

- Hash Table (Lookup table)
  - FAST, but requires perfect matches.
- Dynamic Programming (Smith Waterman)
  - Indels
  - Mathematically optimal solution
  - Slow (most programs use Hash Mapping as a prefilter)
- Burrows-Wheeler Transform (BW Transform)
  - FAST (without mismatch/gap)
  - Memory efficient.
  - But for gaps/mismatches, it lacks sensitivity

Spaced seed alignment

- Tags and tag-sized pieces of reference are cut into small “seeds.”
- Pairs of spaced seeds are stored in an index.
- Look up spaced seeds for each tag.
- For each “hit,” confirm the remaining positions.
- Report results to the user.



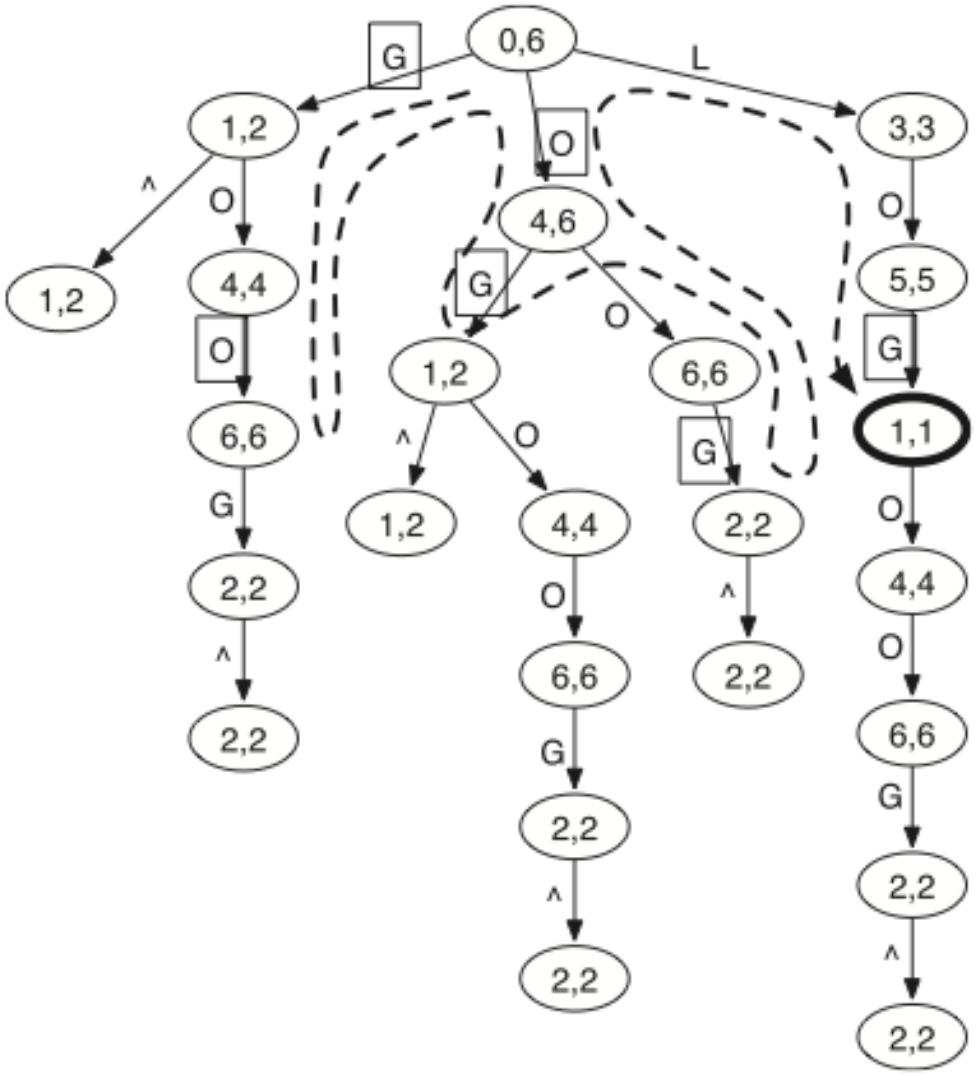


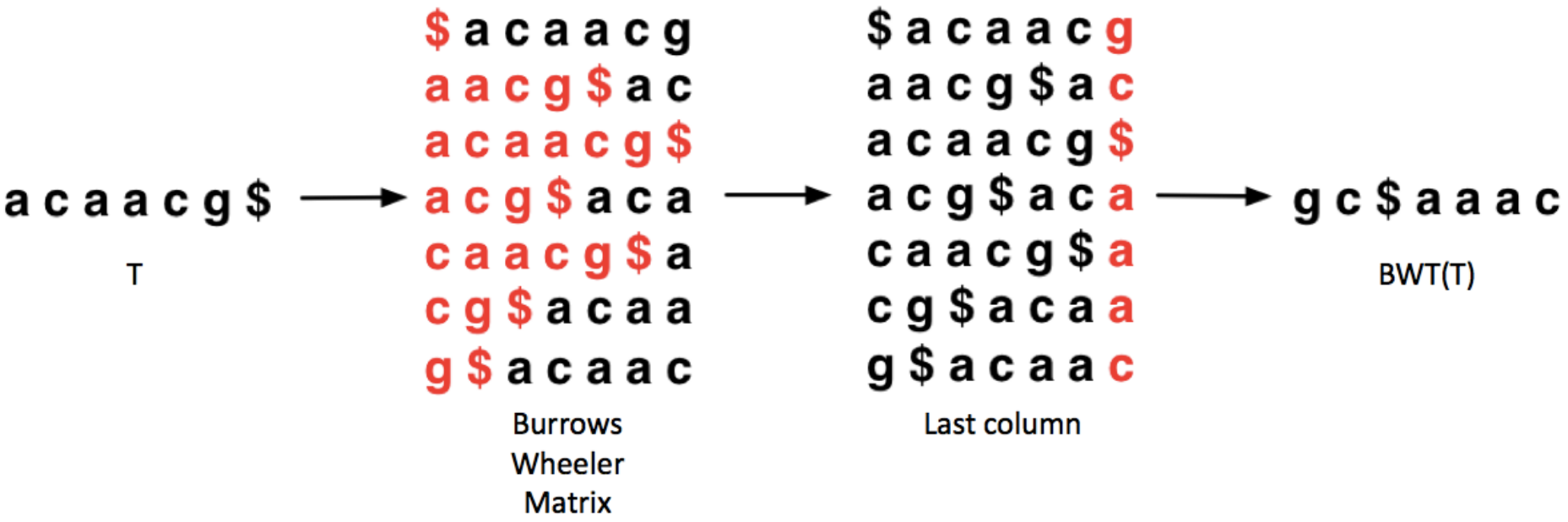
Fig. 1. Prefix trie of string 'GOOGOL'. Symbol ∧ marks the start of the string. The two numbers in a node give the SA interval of the string represented by the node (see Section 2.3). The dashed line shows the route of the brute-force search for a query string 'LOL', allowing at most one mismatch. Edge labels in squares mark the mismatches to the query in searching. The only hit is the bold node [1, 1] which represents string 'GOL'.

Prefix trie and string matching

The prefix trie for string X is a tree where each edge is labeled with a symbol and the string concatenation of the edge symbols on the path from a leaf to the root gives a unique prefix of X.

Burrows-Wheeler Transform

- Reversible permutation used originally in compression



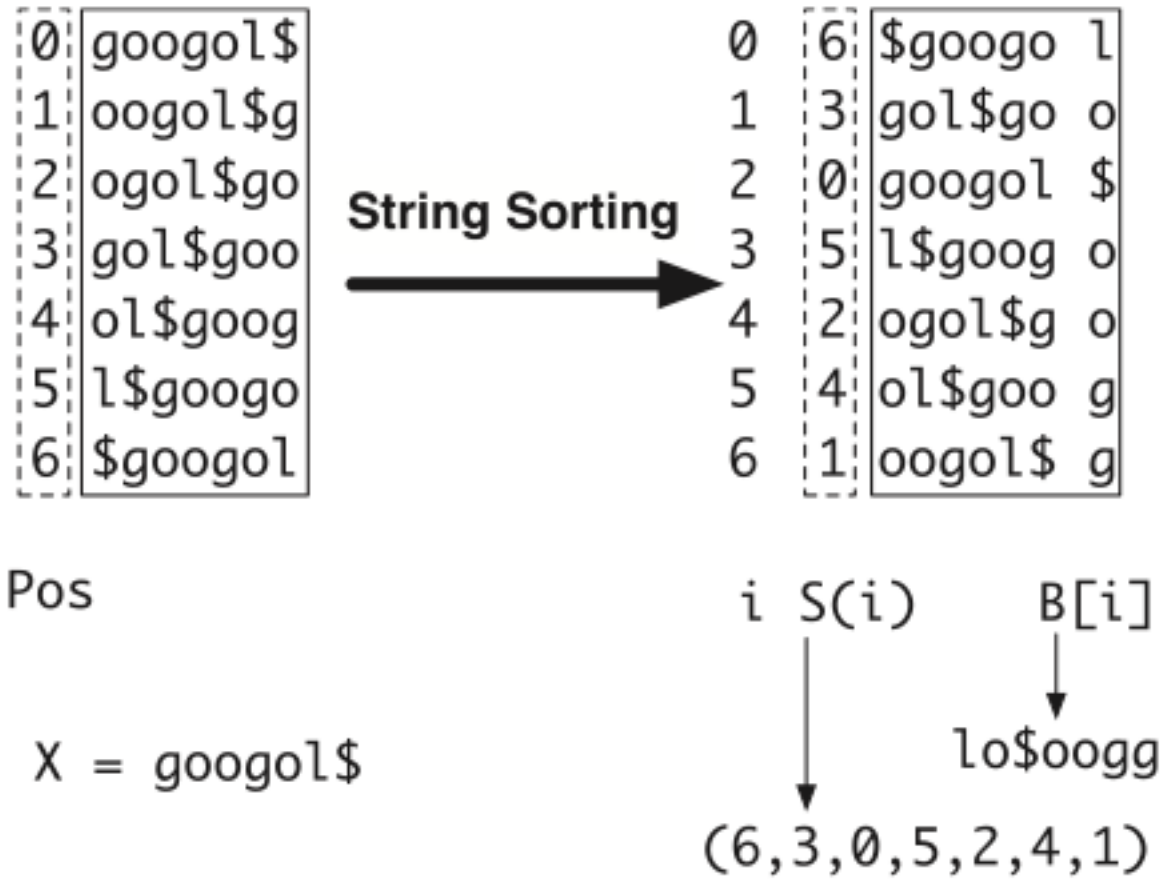
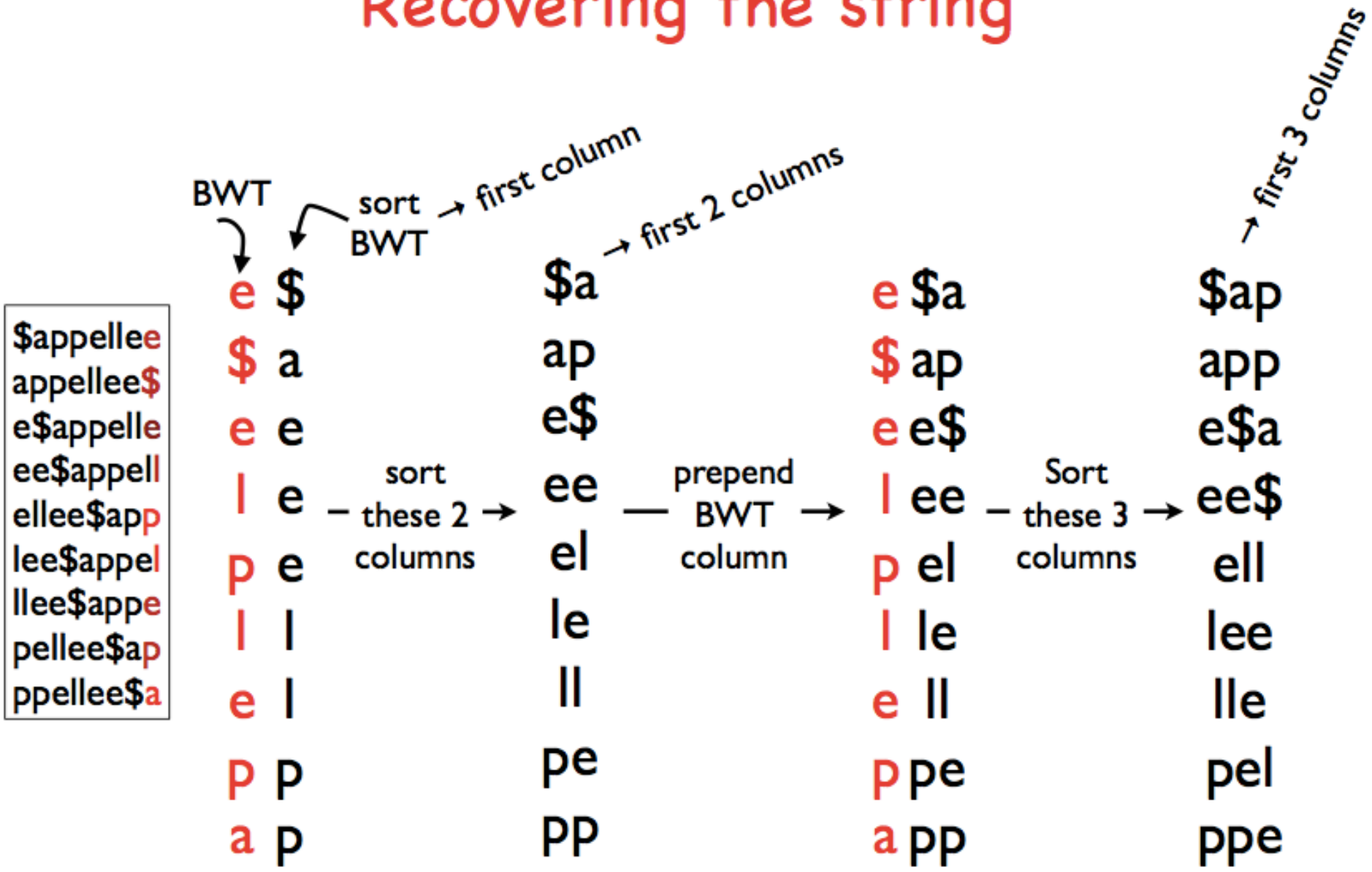


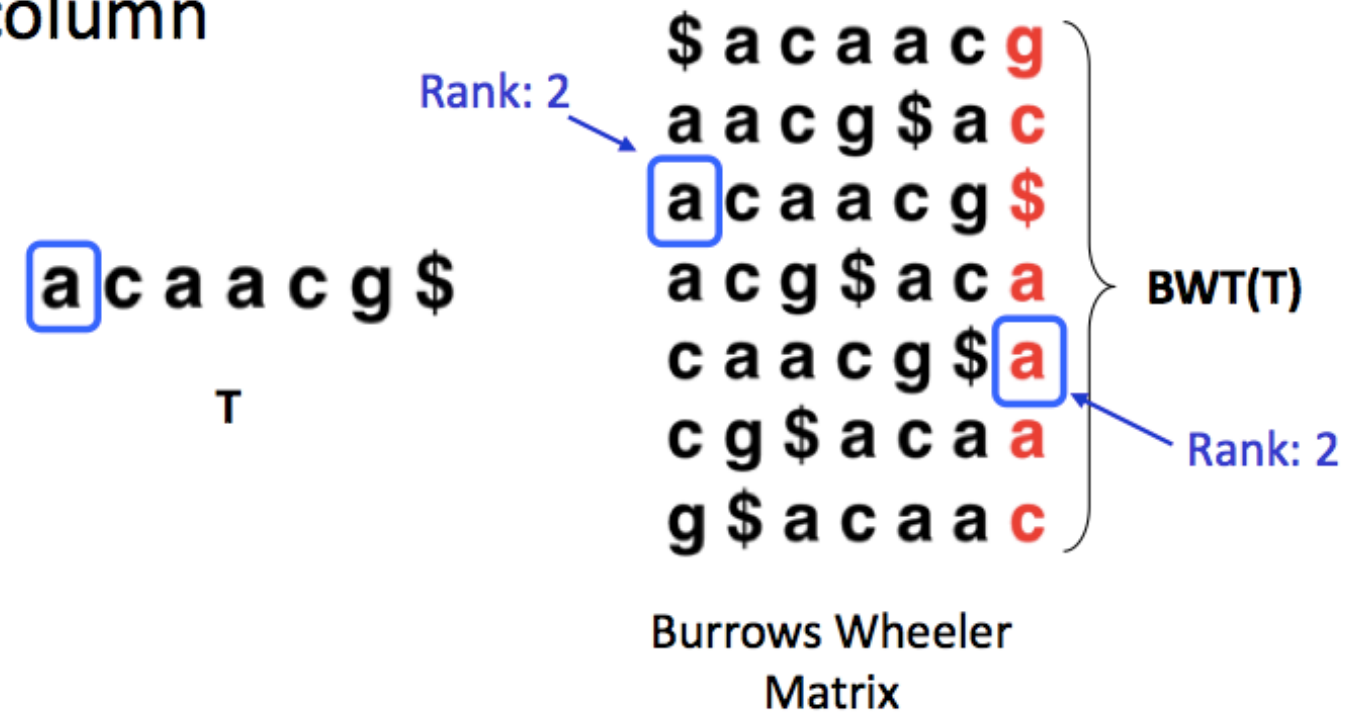
Fig. 2. Constructing suffix array and BWT string for $X = \texttt{googol\$}$. String X is circulated to generate seven strings, which are then lexicographically sorted. After sorting, the positions of the first symbols form the suffix array (6,3,0,5,2,4,1) and the concatenation of the last symbols of the circulated strings gives the BWT string `lo$oogg`.

Recovering the string



Burrows-Wheeler Transform

- Property that makes BWT(T) reversible is "LF Mapping"
  - i^{th} occurrence of a character in **L**ast column is same *text* occurrence as the i^{th} occurrence in **F**irst column

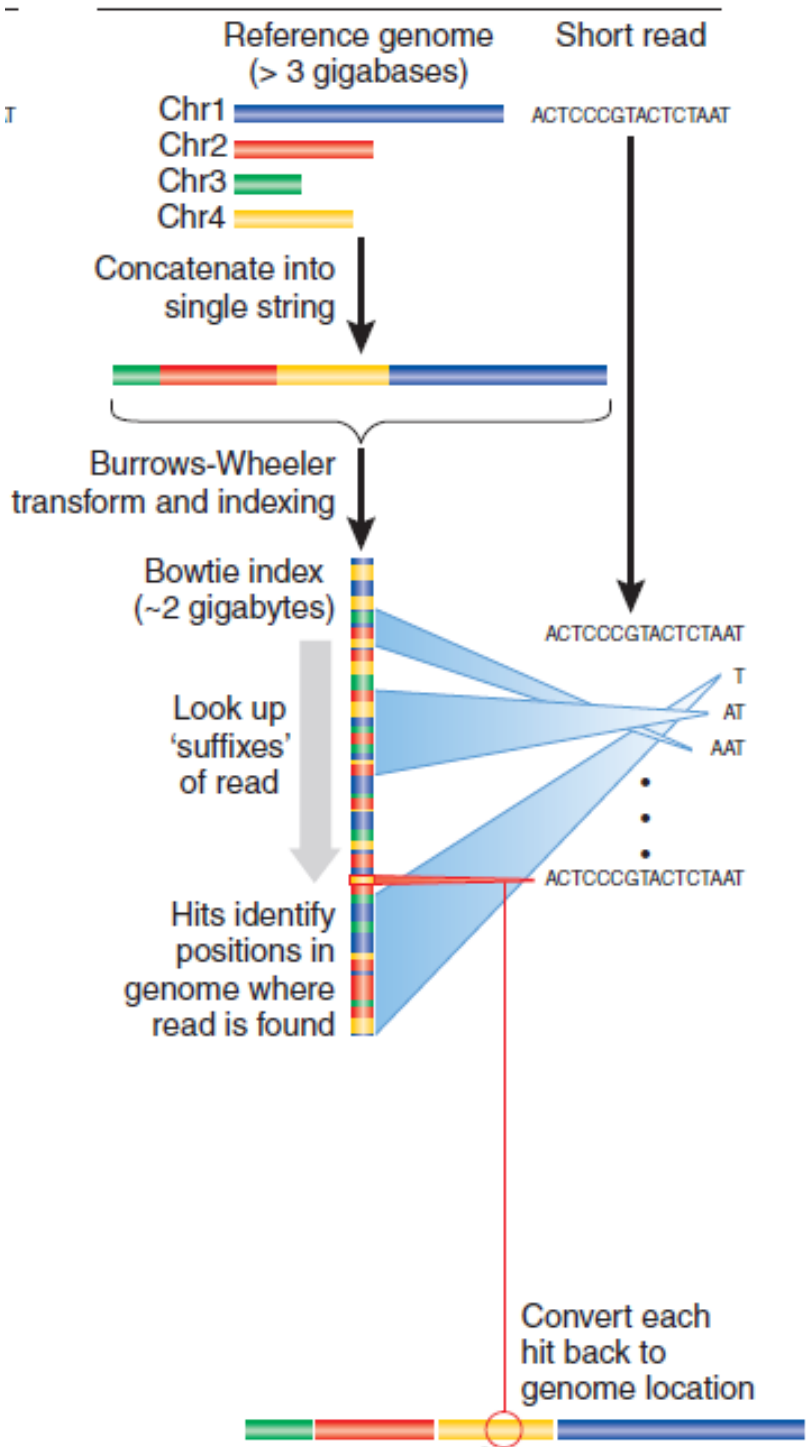


Burrows-Wheeler Transform

- To recreate T from BWT(T), repeatedly apply rule:
  - **T** = **BWT**[**LF**(i)] + **T**; i = **LF**(i)
  - Where **LF**(i) maps row i to row whose first character corresponds to i's last per LF Mapping

BWT Search

The LF mapping is also used in exact matching. Because the matrix is sorted lexicographically, rows beginning with a given sequence appear consecutively.



Burrows-Wheeler

- Store entire reference genome.
- Align tag base by base from the end.
- When tag is traversed, all active locations are reported.
- If no match is found, then back up and try a substitution.

Why Burrows-Wheeler?

BWT very compact:

Approximately ½ byte per base

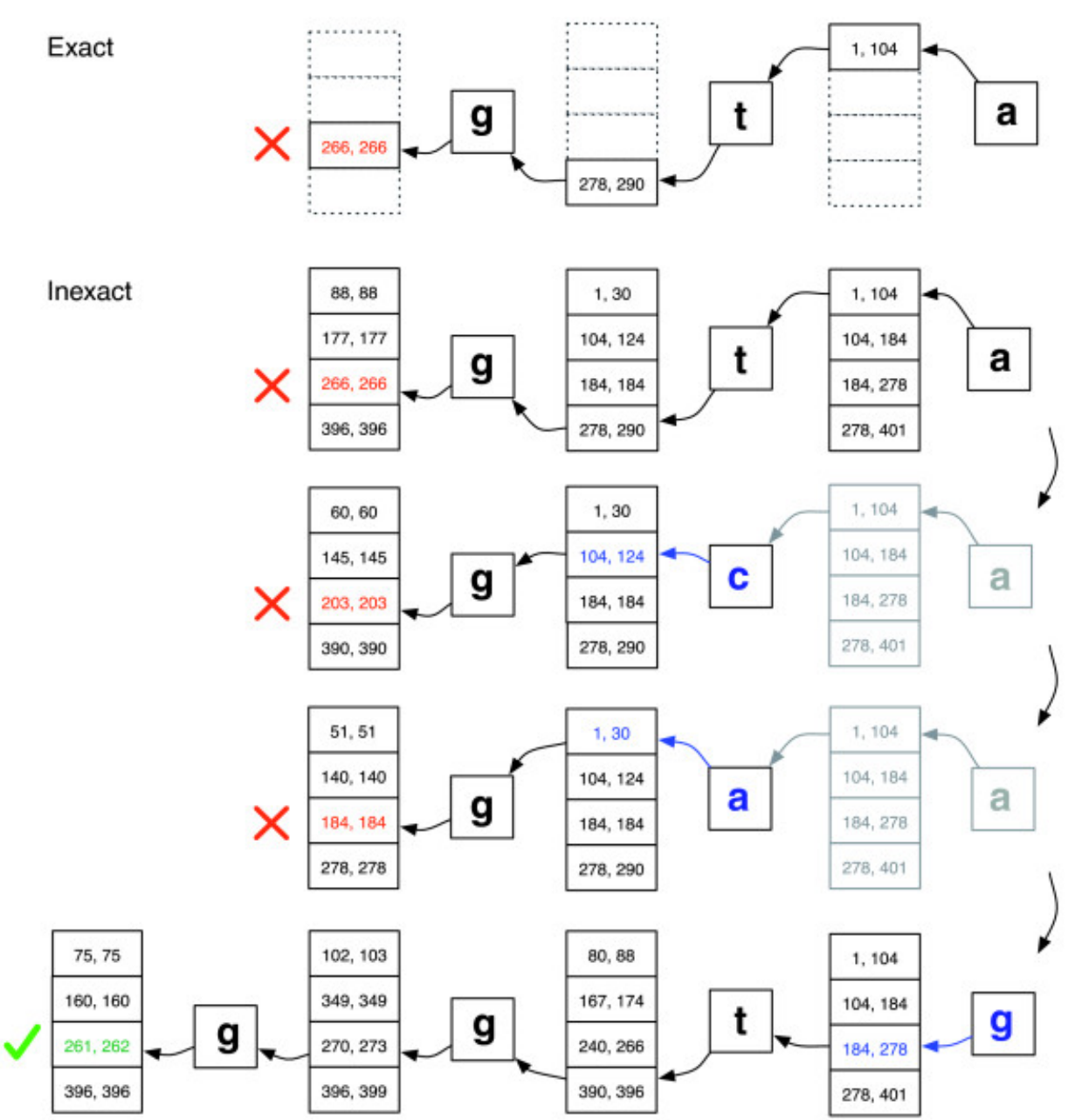
As large as the original text, plus a few "extras"

Can fit onto a standard computer with 2GB of memory

- Linear-time search algorithm

  proportional to length of query for exact matches

Inexact match



References

- (Bowtie) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome, Langmead et al, Genome Biology 2009, 10:R25
- SOAP: short oligonucleotide alignment, Ruiqiang Li et al. Bioinformatics (2008) 24: 713-4
- (BWA) Fast and Accurate Short Read Alignment with Burrows-Wheeler Transform, Li Heng and Richard Durbin, (2009) 25:1754–1760
- SOAP2: an improved ultrafast tool for short read alignment, Ruiqiang Li, (2009) 25: 1966–1967
- (MAQ) Mapping short DNA sequencing reads and calling variants using mapping quality scores. Li H, Ruan J, Durbin R. Genome Res. (2008) 18:1851-8.

Main advantage of BWT against suffix array

- BWT needs less memory than suffix array
- For human genome m = $3 * 10^9$:
  - Suffix array: $m\log_2(m)$ bits = 4m bytes = 12GB
  - BWT: m/4 bytes plus extras = 1 - 2 GB
    - m/4 bytes to store BWT (2 bits per char)
    - Suffix array and occurrence counts array take 5 m $\log_2$ m bits = 20 n bytes
    - In practice, SA and OCC only partially stored, most elements are computed on demand (takes time!)
    - Tradeoff between time and space

List of reads mappers: Bioinformatics. 2012 Dec 15;28(24):3169-77.

Mapper	Data	Seq.Plat.	Input	Output	Avail.	Version	Cit.	Citations / Years	Reference
BFAST	DNA	I,So,4, Hel	(C)FAST(A/Q)	SAM TSV	OS	0.7.0	94	37.11	Homer *et al.* (2009)
Bismark	Bisulfite	I	FASTA/Q	SAM	OS	0.7.3	7	6.21	Krueger and Andrews (2011)
Blat	DNA	N	FASTA	TSV BLAST	OS	34	2844	275.67	Kent (2002)
Bowtie	DNA	I,So,4,Sa,P	(C)FAST(A/Q)	SAM TSV	OS	0.12.7	1168	363.42	Langmead *et al.* (2009)
Bowtie2	DNA	I,4,Ion	FASTA/Q	SAM TSV	OS	2.0beta5		0.00	Langmead and Salzberg (2012)
BS Seeker	Bisulfite	I	FASTA/Q	SAM	OS		19	9.26	Chen *et al.* (2010)
BSMAP	Bisulfite	I	FASTA/Q	SAM TSV	OS	2.43	31	11.06	Xi and Li (2009)
BWA	DNA	I,So,4,Sa,P	FASTA/Q	SAM	OS	0.6.2	738	224.20	Li and Durbin (2009)
BWA-SW	DNA	I,So,4,Sa,P	FASTA/Q	SAM	OS	0.6.2	160	67.69	Li and Durbin (2010)
BWT-SW	DNA	N	FASTA	TSV	OS	20070916	45	10.42	Lam *et al.* (2008)
CloudBurst	DNA	N	FASTA	TSV	OS	1.1	146	46.97	Schatz (2009)
DynMap	DNA	N	FASTA	TSV	OS	0.0.20		0.00	Flouri *et al.* (2011)
ELAND	DNA	I	FASTA	TSV	Com	2	7	1.09	Unpublished[1]
Exonerate	DNA	N	FASTA	TSV	OS	2.2	255	34.69	Slater and Birney (2005)
GEM	DNA	I, So	FASTA/Q	SAM, Counts	Bin	1.x	4	1.35	Unpublished[2]
GenomeMapper	DNA	I	FASTA/Q	BED TSV	OS	0.4.3	31	11.66	Schneeberger *et al.* (2009)
GMAP	DNA	I,4,Sa,Hel,Ion,P	FASTA/Q	SAM, GFF	OS	2012-04-27	217	29.52	Wu and Watanabe (2005)
GNUMAP	DNA	I	FASTA/Q Illumina	SAM TSV	OS	3.0.2	15	5.73	Clement *et al.* (2010)
GSNAP	DNA	I,4,Sa,Hel,Ion,P	FASTA/Q	SAM	OS	2012-04-27	72	31.61	Wu and Nacu (2010)
MapReads	DNA	So	FASTA/Q	TSV	OS	2.4.1		0.00	Unpublished[3]
MapSplice	RNA	I	FASTA/Q	SAM BED	OS	1.15.2	50	28.17	Wang *et al.* (2010)
MAQ	DNA	I,So	(C)FAST(A/Q)	TSV	OS	0.7.1	957	251.66	Li *et al.* (2008a)
MicroRazerS	miRNA	N	FASTA	SAM TSV	OS	0.1	7	2.75	Emde *et al.* (2010)
MOM	DNA	I,4	FASTA	TSV	Bin	0.6	18	5.55	Eaves and Gao (2009)
MOSAIK	DNA	I,So,4,Sa,Hel,Ion,P	(C)FAST(A/Q)	BAM	OS	2.1	4	1.18	Unpublished[4]
mrFAST	miRNA	I	FASTA/Q	SAM	OS	2.1.0.4	158	58.34	Alkan *et al.* (2009)
mrsFAST	miRNA	I,So	FASTA/Q	SAM	OS	2.3.0	32	18.03	Hach *et al.* (2010)
Mummer 3	DNA	N	FASTA	TSV	OS	3.23	683	81.58	Kurtz *et al.* (2004)
Novoalign	DNA	I,So,4,Ion,P	(C)FAST(A/Q) Illumina	SAM TSV	Bin	V2.08.01	137	34.49	Unpublished[5]
PASS	DNA	I,So,4	(C)FAST(A/Q)	SAM GFF3 BLAST	Bin	1.62	45	13.67	Campagna *et al.* (2009)
Passion	RNA	I,4,Sa,P	FASTA/Q	BED	OS	1.2.0		0.00	Zhang *et al.* (2012)
PatMaN	miRNA	N	FASTA	TSV	OS	1.2.2	38	9.36	Prüfer *et al.* (2008)
PerM	DNA	I,So	(C)FAST(A/Q)	SAM TSV	OS	0.4.0	30	10.88	Chen *et al.* (2009)

List of reads mappers (continuation)

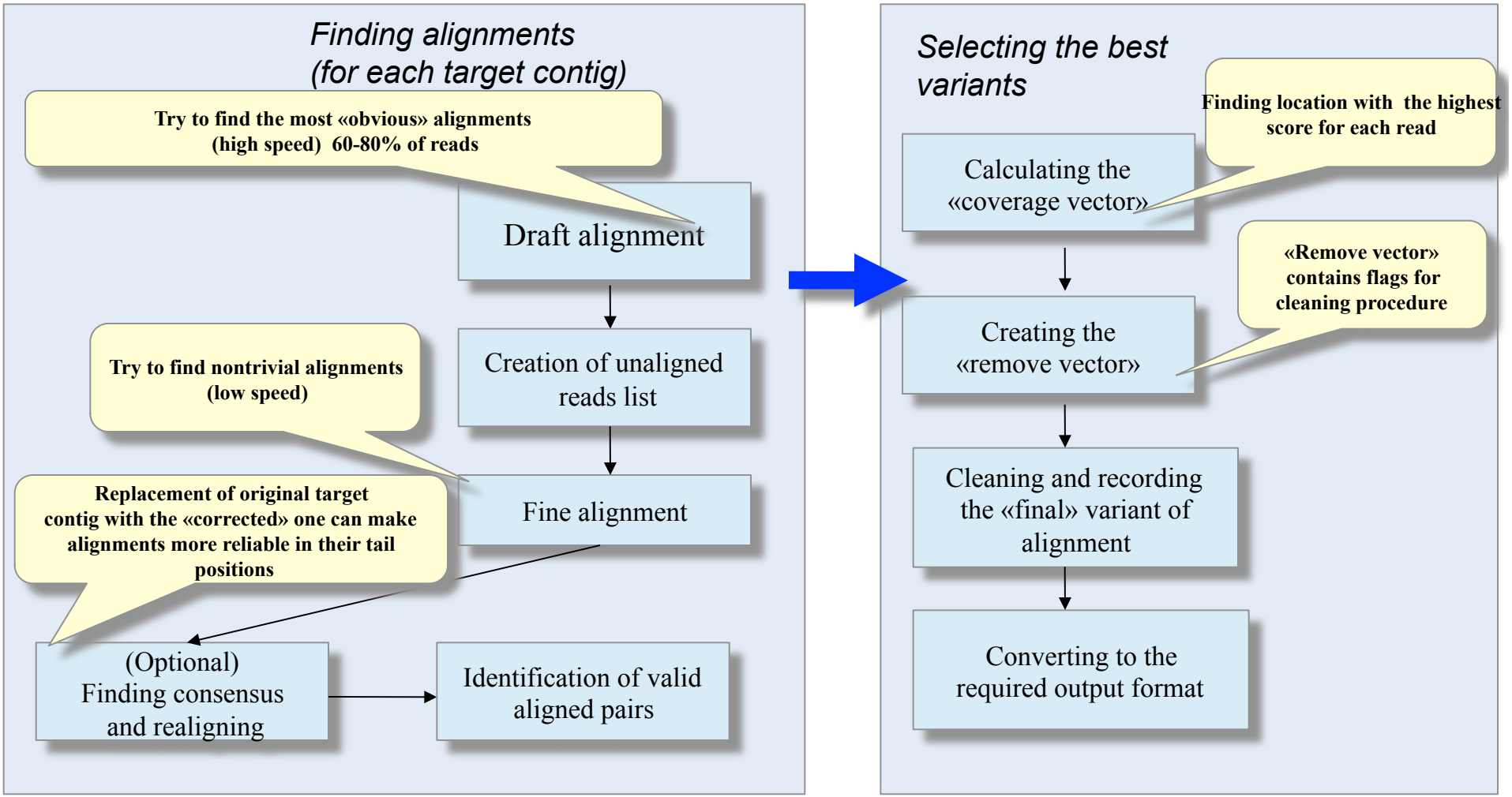
ProbeMatch	DNA	I,4,Sa	FASTA	ELAND	OS		6	1.92	Kim *et al.* (2009)
QPALMA	RNA	I,4	Specific	TSV	OS	0.9.2	75	21.11	De Bona *et al.* (2008)
RazerS	DNA	I,4	FASTQ	TSV ELAND	OS	1.1	58	20.17	Weese *et al.* (2009)
REAL	DNA	I	FASTA/Q	TSV	OS	0.0.28		0.00	Frousios *et al.* (2010)
RMAP	DNA	I,So,4	(C)FAST(A/Q)	BED	OS	2.05	162	38.27	Smith *et al.* (2008)
RNA-Mate	RNA	So	CFASTA	BED Counts	OS	1.1	28	10.04	Cloonan *et al.* (2009)
RUM	RNA	I,4	FASTA/Q	SAM TSV BED	OS	1.11	2	2.36	Grant *et al.* (2011)
SeqMap	DNA	I	FASTA	ELAND	OS	1.013	142	37.34	Jiang and Wong (2008)
SHRiMP	DNA	I,So,4,Hel	(C)FAST(A/Q)	TSV	OS	1.3.2	155	50.91	Rumble *et al.* (2009)
SHRiMP 2	DNA	I,So,4	FASTA/Q	SAM	OS	2.2.2	15	11.76	David *et al.* (2011)
Slider	DNA	I	Illumina	TSV	OS	0.6	39	10.98	Malhis *et al.* (2009)
Slider II	DNA	I	Illumina	TSV	OS	1.1	16	7.25	Malhis and Jones (2010)
Smalt	DNA	I,4,Sa,Ion,P	FASTA/Q	SAM	OS	0.6.1		0.00	Unpublished[6]
SOAP	DNA	I	FASTA/Q	TSV	OS	1.11	451	104.41	Li *et al.* (2008b)
SOAP2	DNA	I	FASTA/Q	SAM TSV	OS	2.21	294	99.38	Li *et al.* (2009b)
SOAPSplice	RNA	I,4	FASTA/Q	TSV	Bin	1.8	3	3.54	Huang *et al.* (2011a)
SOCS	DNA	So	(C)FAST(A/Q)	TSV	OS	2.1.1	49	14.15	Ondov *et al.* (2008)
SpliceMap	RNA	I	FASTA/Q	SAM BED	OS	3.3.5.2	63	29.80	Au *et al.* (2010)
SSAHA	DNA	N	FASTA/Q	TSV	OS	3.1	483	42.29	Ning *et al.* (2001)
SSAHA2	DNA	I,4,Sa	FASTA/Q	SAM	Bin	2.5.5	483	44.99	Ning *et al.* (2001)
Stampy	DNA	I	FASTA/Q	SAM TSV	Bin	1.0.16	26	16.19	Lunter and Goodson (2011)
Supersplat	RNA	N	FASTA	TSV	OS	1.0	21	9.93	Bryant Jr *et al.* (2010)
TopHat	RNA	I	FASTA/Q, GFF	BAM	OS	1.4.1	389	121.04	Trapnell *et al.* (2009)
VMATCH	DNA	N	FASTA	TSV	Bin		26	2.75	Unpublished[7]
WHAM	DNA	N	FASTQ	SAM	OS	0.1.4	3	3.33	Li *et al.* (2011)
X-Mate	DNA	I,So,4	(C)FAST(A/Q)	SAM BED Counts	OS	1	1	0.74	Wood *et al.* (2011)
ZOOM	DNA	I,So,4	(C)FAST(A/Q)	SAM BED GFF	Com	1.5	109	28.66	Lin *et al.* (2008)

Mapping reads with mutated sequences

%	#mapped	ReadsMap		#mapped	BWT	
mutations	reads	Sn	Sp	reads	Sn	Sp
1	**18363276**	**0.88783**	**0.92828**	**20428.64**	**0.91541**	**0.91408**
2	**18368502**	**0.75714**	**0.79191**	**17334.35**	**0.78026**	**0.77373**
3	**18361496**	**0.79248**	**0.82913**	**17974.39**	**0.81714**	**0.78807**
4	**18365644**	**0.64525**	**0.67502**	**17068.01**	**0.66489**	**0.59820**
5	**18361920**	**0.65808**	**0.68847**	**16426.47**	**0.67852**	**0.53796**
6	**18364062**	**0.63162**	**0.66118**	**15978.07**	**0.65195**	**0.42795**
7	**18369140**	**0.61925**	**0.64801**	**15987.15**	**0.63861**	**0.32685**
8	**18367384**	**0.59114**	**0.61875**	**16378.48**	**0.60893**	**0.23003**
9	**18373472**	**0.58140**	**0.60824**	**17666.77**	**0.60000**	**0.16000**
10	**18371406**	**0.54331**	**0.56774**	**18658.51**	**0.56072**	**0.10136**

ReadsMap

Workflow of alignment of genomic reads (no intron insertions) to the reference genome



Tests results on genome reads

	Reads #	**Aligned** (Percent)	Alignments Number	True alignments	Sp	Sn
BWA (no pair)	18 363 068	18 277 290 (0.99533)	18 277 290	17 836 240	**0.97587**	**0.97131**
BWA (pair)	18 363 068	18 359 440 (0.99980)	18 359 440	18 087 459	**0.98519**	**0.98499**
TopHat (no pair)	18 363 068	17 527 411 (0.95449)	19 039 852	17 4988 77	0.91907	0.95294
TopHat (pair)	18 363 068	18 076 620 (0.98440)	19 018 097	18 047 001	0.94894	0.98279
Bowtie (no pair)	18 363 068	18 186 084 (0.99036)	19 782 028	18 170 026	0.91851	0.98949
Bowtie (pair)	18 363 068	18 010 584 (0.98080)	19 337 086	17 997 376	0.93072	0.98009
ReadsMap_unspl (no pair)	18 363 068	18 363 057 (0.99999)	19 887 669	18 252 554	0.91778	0.99398
ReadsMap_unspl (pair)	18 363 068	18 363 036 (0.99999)	19 048 464	18 257 367	0.95847	0.99424
CleanReads **ReadsMap**_unspl (no pair)	18 363 068	18 363 058 (0.99999)	19 889 301	18 312 219	**0.92071**	**0.99723**
CleanReads **ReadsMap**_unspl (pair)	18 363 068	18 363 038 (0.99999)	19 047 654	18 315 257	**0.96155**	**0.99740**

Example of read alignment disrupted by intron close to the read end

ReadsMap: (generates right alignment)

```
[DD] Sequence:       3(      1), S:           1, L:       76 3
Block of alignment: 2
    1 E: 16275272      6 [-- CT] P: 16275272         1, L:       6, H:100.00, W:     60, S:0
    2 E: 16277748     70 [AC --] P: 16277748         7, L:      70, H:100.00, W:    700, S:0
          1        11  16275256  16275266  16275276  16275285  16277732  16277739
          nnnnnnnnnnnnnnnn(..)gtcagaaagtaactggCAAATT]ctatgtataaaattgt(..)taatgtaaacacttac[
          ................(..)..............|||||| ................(..).................
          ----------------(..)----------------CAAATT ----------------(..)----------------
          1         1         1         1         5         7         7         7

   16277748  16277758  16277768  16277778  16277788  16277798  16277808  16277818
          ACATTATGACGACTAGAAACAGCATACTCTCTGGCCGTCTGTCCAGATAGATCTTGAGAAGATACATCAAtgttttgctc
          ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||.........
          ACATTATGACGACTAGAAACAGCATACTCTCTGGCCGTCTGTCCAGATAGATCTTGAGAAGATACATCAA----------
```

Bowtie (Langmead et al., 2010) (generates random alignment of the left short segment)

```
[DD] Sequence:       3(      1), S:           1, L:       76 3
Block of alignment: 2
    1 E: 16277745      4 [ct CA] P: 16277745         1, L:       4, H: 50.00, W:    -22, S:8
    2 E: 16277749     69 [CA tg] P: 16277749         8, L:      69, H:100.00, W:    690, S:8
          1        11  16277729  16277739  16277747  16277752  16277762  16277772
          nnnnnnnnnnnnnnnn(..)ttttaatgtaaacact?[TACA]---[CATTATGACGACTAGAAACAGCATACTCTCTGG
          ................(..).................  0|0| ... ||||||||||||||||||||||||||||||
          ----------------(..)----------------  CAAA tta CATTATGACGACTAGAAACAGCATACTCTCTGG
          1         1         1         1         3        11        21        31

   16277782  16277792  16277802  16277812  16277820  16277830  51304553  51304563
          CCGTCTGTCCAGATAGATCTTGAGAAGATACATCAA]?tgttttgctcaagtag(..)nnnnnnnnnnnnnnnn
          ||||||||||||||||||||||||||||||||||  .................(..).................
          CCGTCTGTCCAGATAGATCTTGAGAAGATACATCAA  ----------------(..)----------------
         41        51        61        71        77        77        77        77
```

ReadsMap Intron Restoration example using reliably mapped reads

Intron restoration procedure in the case of short unaligned flanks.

A. Initial "draft" alignment. At the left end there is the short unaligned flank of 3 nucleotides length (marked by red color).

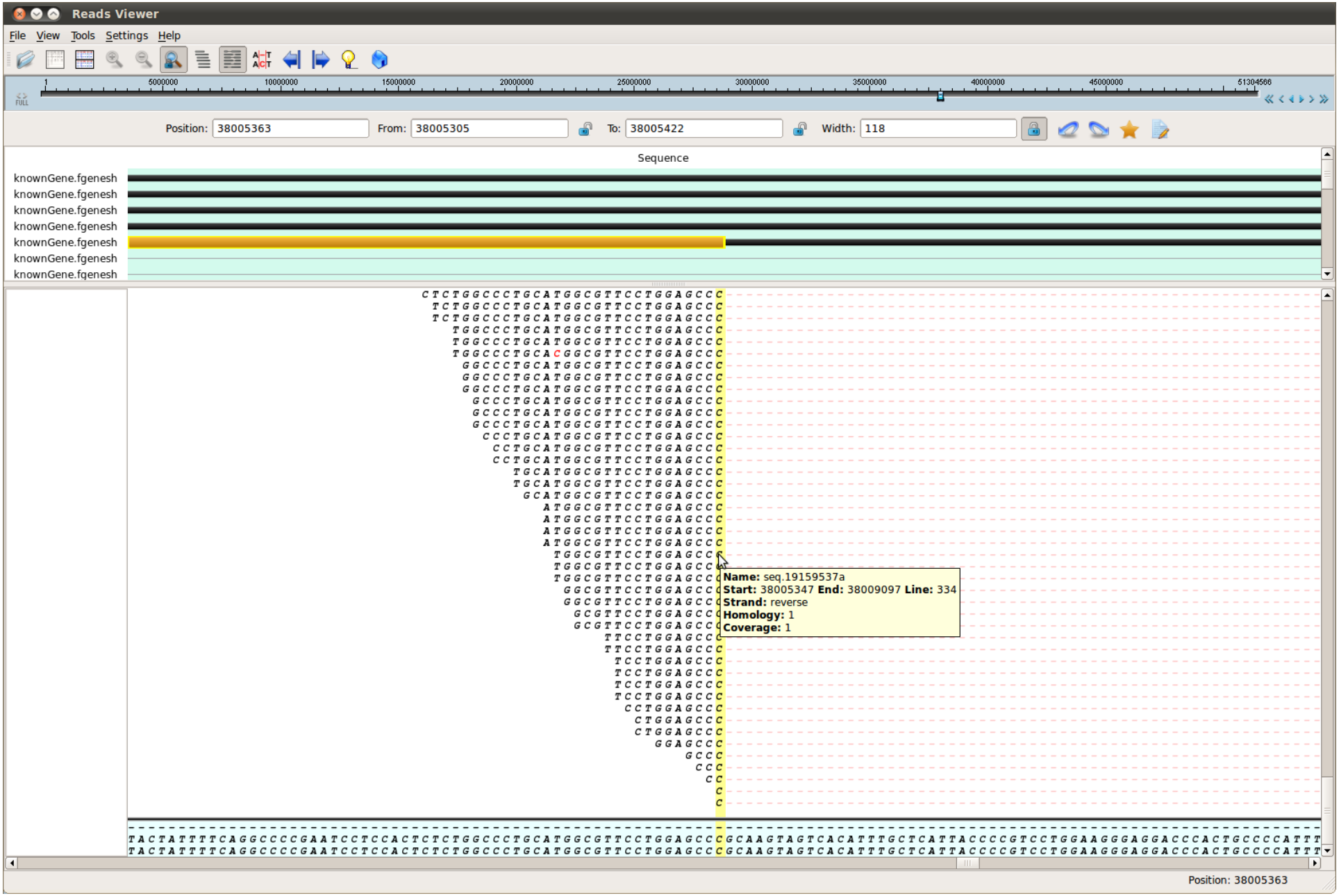
```
nnnnnn(..)ttgaatataaaagtatACCTTTCTATCACCACCCTTATTTATTTCTGGTTCTTGAGACATTTCctgcagatgcaaaaac(..)
......(..).................||||||||||||||||||||||||||||||||||||||||||||||.................(..)
------(..)-------------tcaACCTTTCTATCACCACCCTTATTTATTTCTGGTTCTTGAGACATTTC----------------(..)
```

B. Reliable(intron containing) alignment that «support» a potential intron. At the edges of blocks there are classic splicing sites (CT-AC in complement chain) and size of blocks is sufficient to postulate the «correctness» of the current alignment.

```
tt  CATTTCTTCTTCAAC]cttgaatgaaagtttg(..)gaatataaaagtatac[CTTTCTATCACCACCCTTATTTATTTCTGGTTCTT  ga
..  ||||||||||||||| ..............(..)..............  |||||||||||||||||||||||||||||||||  ..
--  CATTTCTTCTTCAAC ---------------(..)---------------  CTTTCTATCACCACCCTTATTTATTTCTGGTTCTT  --
```

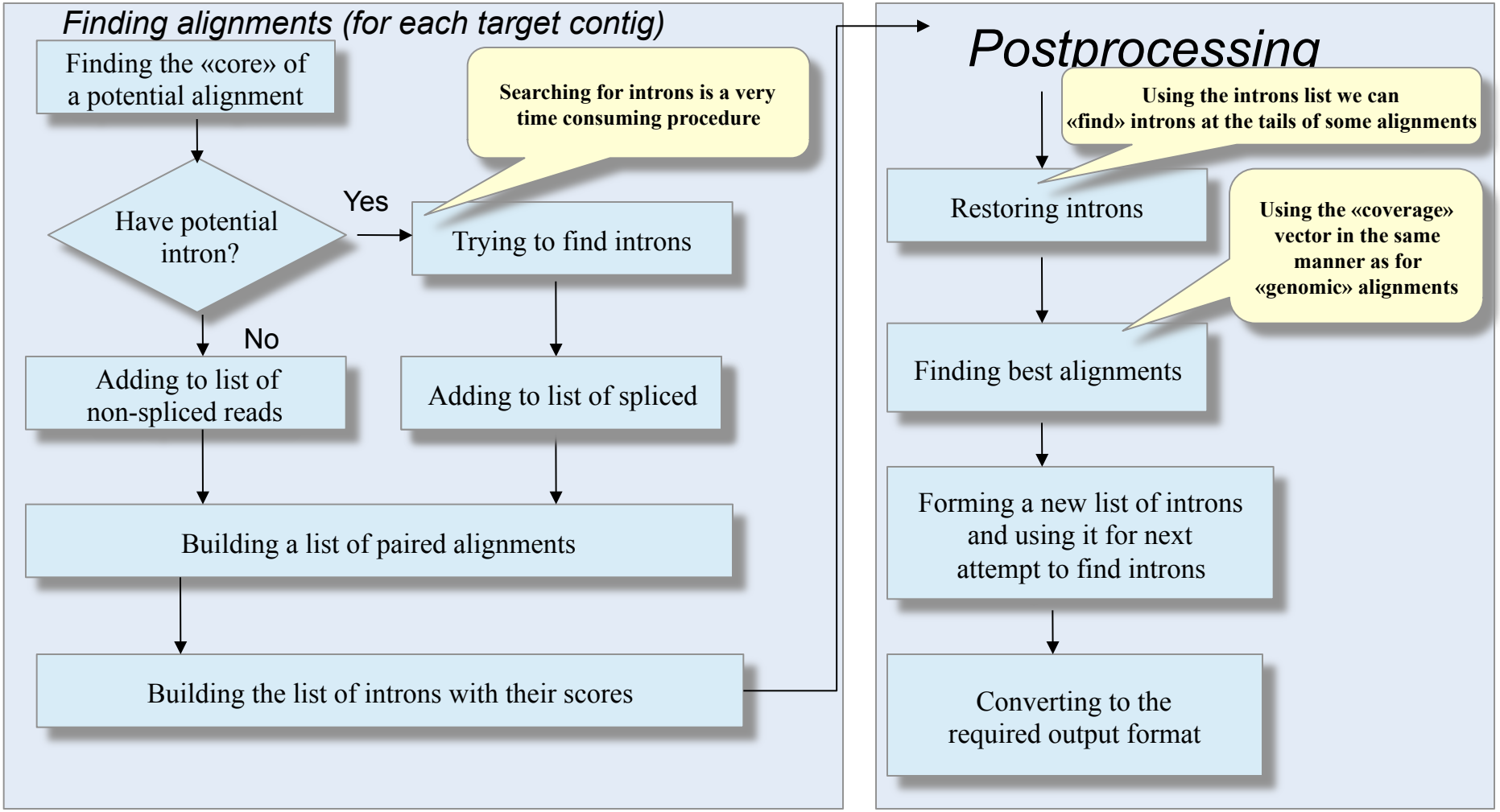
C. Result of intron restoration. **Based on «supporting» alignments**, not only 3 unaligned nucleotides (see A) but also 2 neighboring ones, that were originally the part of the main block (marked with color), were moved to the left exon. As a result the read is not just fully aligned, but the intron is also correctly located.

```
ctTCAAC]cttgaatgaaagtttg(..)gaatataaaagtatac[CTTTCTATCACCACCCTTATTTATTTCTGGTTCTTGAGACATTTCct
..|||||  ..............(..)..............  ||||||||||||||||||||||||||||||||||||||||||..
--TCAAC ---------------(..)---------------  CTTTCTATCACCACCCTTATTTATTTCTGGTTCTTGAGACATTTC--
```


Reads Viewer
File View Tools Settings Help
1 5000000 10000000 15000000 20000000 25000000 30000000 35000000 40000000 45000000 51304566
Position: 38005363 From: 38005305 To: 38005422 Width: 118
Sequence
knownGene.fgenesh
knownGene.fgenesh
knownGene.fgenesh
knownGene.fgenesh
knownGene.fgenesh
knownGene.fgenesh
knownGene.fgenesh
Name: seq.19159537a
Start: 38005347 End: 38009097 Line: 334
Strand: reverse
Homology: 1
Coverage: 1
Position: 38005363

ReadsMap

Workflow of alignment of RNASeq reads (with possible intron insertions)



Test sets for read mapping software

Genomic reads (generated from 22 Human chromosome)

Length	Reads Count	InDel	Parametrs
76bp	18 363 068	704 (0.002%) 1-4bp	insert size = 200 bp, standard deviation = 20 bp, coverage = 40

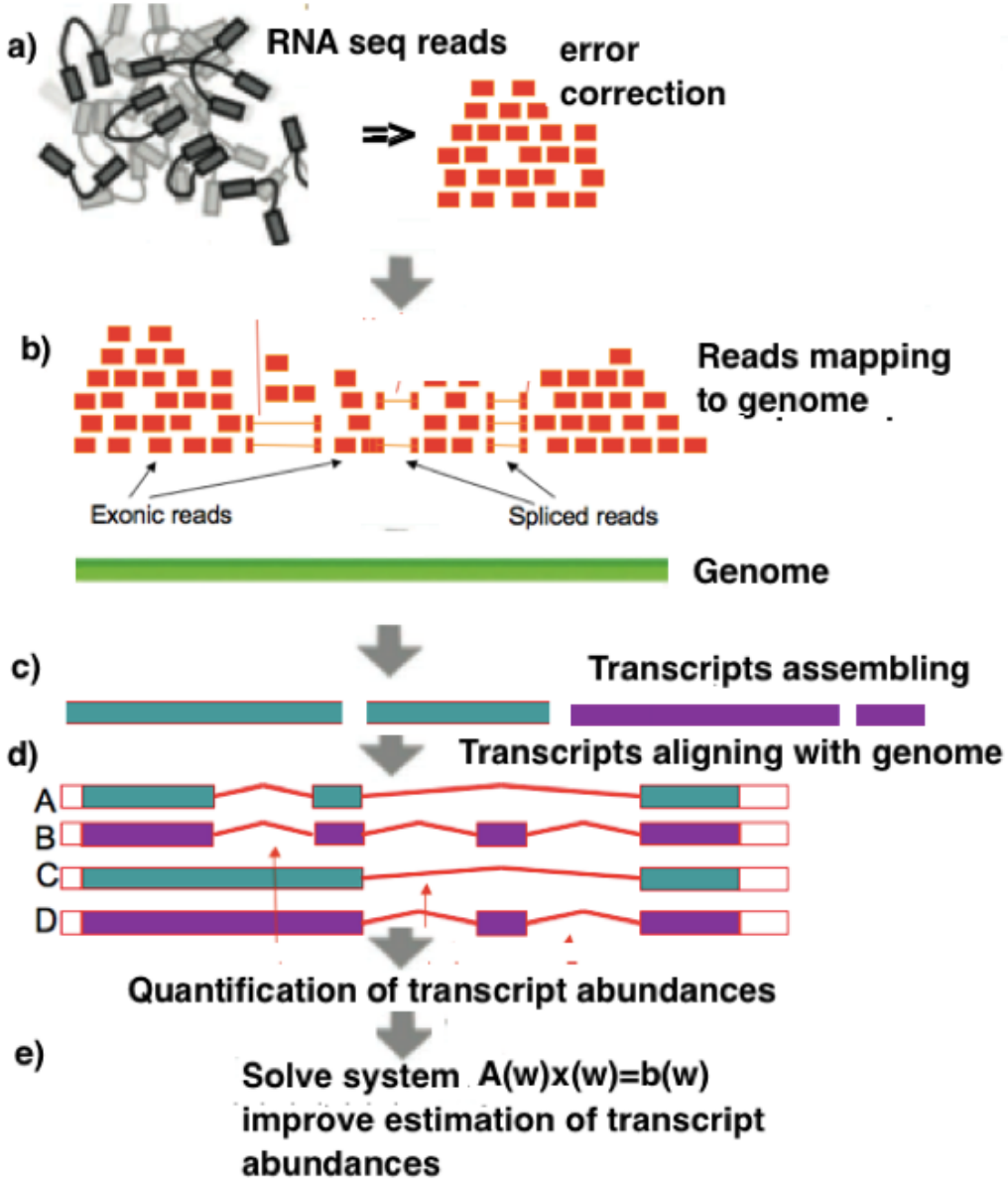
mRNA reads

Length	Reads Count	Introns	Parametrs
50bp	2 979 624	492 743 (16.5%)	insert size = 200 bp, standard deviation = 20 bp, coverage = 40
76bp	1 960 300	485 857 (24.8%)	insert size = 200 bp, standard deviation = 20 bp, coverage = 40
100bp	1 489 796	469 319 (33.3%)	insert size = 300 bp, standard deviation = 30 bp, coverage = 40

Spliced reads tests results

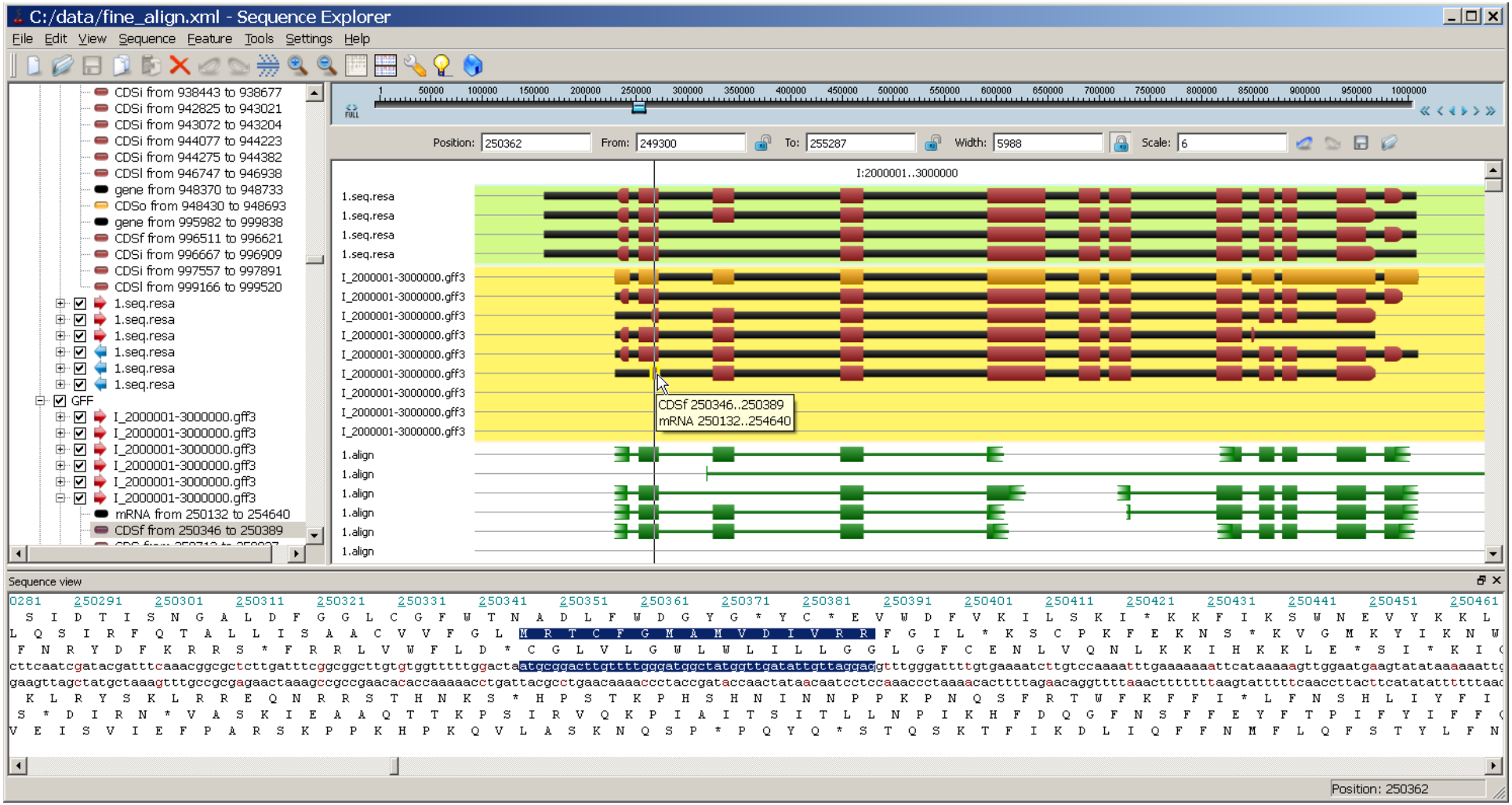
Read length	50bp		76bp		100bp	
	Sp	Sn	Sp	Sn	Sp	Sn
TopHat	**0.92411**	**0.99418**	0.95145	0.98644	0.95673	0.91890
PASS v 2.1.1	0.89005	0.91547	0.88750	0.90603	0.86458	0.87765
ReadsMap	0.93715	0.99172	0.96349	0.99404	0.96220	0.99327
CleanReads ***ReadsMap***	**0.93727**	**0.99309**	**0.96478**	**0.99537**	**0.96478**	**0.99537**

Transomics pipeline for Transcript identification and quantification



Sequence Explorer to analyze discovered alternative splice forms

identifyed using nextgen reads or est mapping to genome sequence



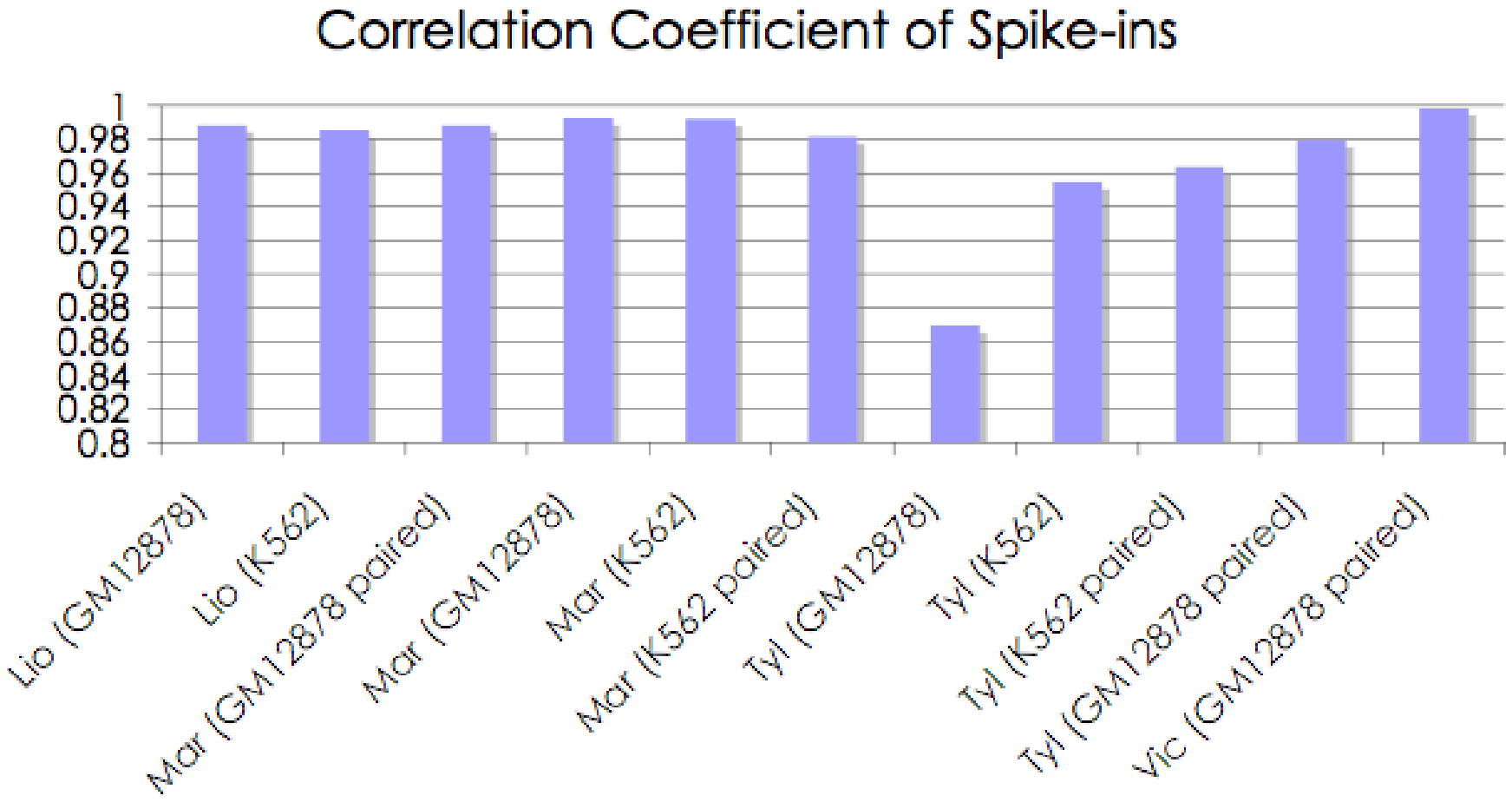
Compute a relative abundance of alternative transcripts generated

We can use a solution of a system of linear equations. Let we have a set of n transcripts from a gene locus **$T= (t_1,t_2,\ldots,t_n)$**.

Let these transcripts can generated altogether a variety of m reads **$R=(r_1,r_2,\ldots,r_m)$**. Each transcript can produce just some of these reads or all of them. Let matrix **$G = (g_{ij})$** will have **$g_{i,j} =1$** if transcript j can generate read **r_i** and **$g_{i,j} =0$** otherwise. The **i-th** column **$(g_{1i}\ g_{2i},\ldots,g_{mi})$** of this matrix shows which reads the transcript **i** can generate. If the quantities of j-th transcript would be **x_j**, then the number of reads of some type produced by n transcripts can be computed as a component of the vector **G x'**, where the vector **$x =(x_1,\ldots.,x_n).$**
If we have observed numbers of reads from R mapped to the gene locus under consideration **$b =(b_1, b_2, \ldots, b_k)$**, than we have a system of linear equations:

Gx' = b',

which need to be solved to determine **unknown quantities of transcripts x**.
This system of linear equations is overdetermined as there are more equations than unknowns (the number of reads is much bigger than the number of transcripts: m >> n). **The method of least squares** can be used to find an approximate solution.

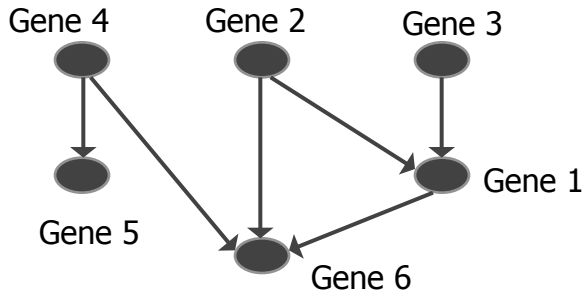


Relative accuracy of spike-in transcript quantification submitted by 11 participants of the RGASP assessment experiment (presented at the workshop by Dr. Kokocinski, The Sanger Institute, Cambridge, member of the assessor's group).

Reconstructing Genetic Regulatory Network

	Exp. 1	.....	Exp. P
Gene 1	0.78	.....	0.50
Gene 2	0.73	.....	0.09
Gene 3	0.99	.....	0.56
.....♪	.....	.....	.....
Gene N	0.28	.....	0.89

Microarray data



Genetic regulation network

RNASeq data nnotation and quantification of all genes and their isoforms across samples.

With microarray data we analyze predefined splicing isoforms , but it could not be used to identify previously uncharacterized events

Ongoing research projects in developing Computational tools for high-throughput analysis of biological data

Eukaryotic genome analysis tools

Annotation of new genomes

FGENESH++: **an automatic eukaryotic gene identification and annotation pipeline**

Bacterial genome analysis tools

FGENESB**: a complex pipeline for annotation of bacterial genomes: genes, operons, promoters and terminators identification**

Software for analysis of next generation sequencing data

- **ab initio genome assembling, reconstruction of sequence using a reference genome**
- **mutation profiling and SNP discovery**
- **assembling transcripts from RNASeq data**

Gene expression regulation

- **Promoter identification**
- **De novo functional motifs discovery**
- **Gene Expression data analysis**
- **Gene networks construction**
- **Databases of regulatory sequences**

High-throughput experimental technique created vast amounts of biological data

Digging out the "treasure" from massive biological data represents the primary challenge in bioinformatics, consequently placing unprecedented demands on big data storage, data manipulation and efficient analysis of this information.