V9 – Genomics data

Program for today:

(1) Read mapping

(2) SNP calling

(3) SNP frequencies in 1000 Genomes data -> consider overlapping genes

(4) Isoforms of genes (alternative splicing)

(5) Non-canonical translation -> not all translated sequences start with AUG

(6) Removing sequence redundancy

You will hear much more about sequences and how to deal with them in the lectures by Prof. Sven Rahman and Prof. Fabian Müller.

Today, in lecture #9, we will deal with several topics around genomic sequences.

Points (1) and (2) deal with the two most common tasks in sequence analysis: mapping of reads and identification of single nucleotide polymorphisms (SNPs).

Sometimes, we need to take special care when preparing a dataset for a statistical analysis.

E.g. we will mention the issue of overlapping genes in point (3) and the issue of removing sequence redundancy in point (6).

In point (5), we will touch on a point that you may consider for granted: translation starts „always“ with a AUG codon that is translated into a methionin (M) amino acid. This is what you read in molecular biology textbooks. It turns out that this is not always the case.

In point (4), we briefly comment on the importance of mRNA and protein isoforms that result from alternative splicing.

(1) Read mapping: range of usage

The accurate **alignment** of reads generated by NGS machines to a reference genome is a crucial part in many **application workflows**, such as

- genome resequencing (in contrast to de novo assembly),
- DNA methylation,
- RNA-Seq (transcriptomics),
- ChIP sequencing (e.g. histone marks, TFBS occupancies),
- SNP detection,
- detection of genomic structural variants, and
- metagenomics (sequencing mixtures of organisms).

Hatem et al.
BMC Bioinformatics (2013) 14:184

Unless we talk about de novo assembly of a genomic sequence, the first task in an NGS project is usually the alignment of sequencing reads to an existing reference genome.

Listed here are some workflows where read mapping is a crucial part.

Read mapping tools

Numerous tools have been developed for this challenging task:

MAQ, RMAP, GSNAP,
Bowtie, Bowtie2,
BWA, SOAP2, Mosaik, FANGS, SHRIMP, BFAST,
MapReads, SOCS, PASS, mrFAST, mrsFAST, ZOOM,
Slider, SliderII, **RazerS**, RazerS3, Novoalign and
GPU-based tools such as SARUMAN and SOAP3.

Hatem et al.
BMC Bioinformatics (2013) 14:184

These are some of the well-known software tools used for mapping of NGS reads.

Read mapping techniques: (1) Hash tables

For most of the existing tools, the mapping process starts by building an **index** either for the reference genome or for the reads.

Then, the index is used to find the corresponding genomic positions for each read.

There are **two main types of techniques** for this: Hash tables + BWT

(1) The hash based methods either hash the reads or the genome.

The main idea for both types is to build a **hash table** for **subsequences** of the reads/genome.

The **key** of each entry is a subsequence

while the **value** is a list of positions

where the subsequence can be found.

Key	Hashed index	Genomic location
"GCTAGC"	Key1	Chr1 123412 … …
"TTTAGC"	KeyN	Chr6 988472

Hatem et al.
BMC Bioinformatics (2013) 14:184

V9 Processing of Biological Data WS 2021/22 4

In principle, one could simply scan the genomic sequence for each read sequence. However, this would be quite inefficient.

Therefore, the existing tools typically construct an index either for the reference genome or for the reads. This index is then used during the string search.

The first type of indexing techniques use a **hash table**, see the table shown on the bottom right.

In this example, the genomic sequence is indexed. Different 6-letter words are each given a hash index and where they are located in the genome.

Read mapping techniques: (2) Burrows Wheeler transform

The **BWT** of the string T = "abracadabra$" is "ard$rcaaaabb.
It is represented by the matrix M where each row is a rotation of the text, and the rows have been sorted lexicographically.
The transform corresponds to the last column labeled L.

I	F	L
1	$ abracadabr	a
2	a $abracadab	r
3	a bra$abraca	d
4	a bracadabra	$
5	a cadabra$ab	r
6	a dabra$abra	c
7	b ra$abracad	a
8	b racadabra$	a
9	c adabra$abr	a
10	d abra$abrac	a
11	r a$abracada	b
12	r acadabra$a	b

Modern alignments use an extension of BWT named **FM index** after Ferragina & Manzina

www.wikipedia.org

V9 Processing of Biological Data WS 2021/22 5

The second technique does not index the string itself, but uses its so-called **Burrows Wheeler transform (BWT)**.

According to Wikipedia, the Burrows–Wheeler transform is an algorithm to prepare data for use with data compression techniques such as bzip2. It was invented by Michael Burrows and David Wheeler in 1994.

The algorithm can be implemented efficiently using a suffix array thus reaching linear time complexity.

An **FM-index** is a compressed full-text substring index based on the Burrows–Wheeler transform, with some similarities to the suffix array. It was created by Paolo Ferragina and Giovanni Manzini.

Read mapping techniques: (2) Burrows Wheeler transform

C[c] is a table that, for each character c in the alphabet, contains the number of occurrences of lexically smaller characters in the text.

C[c] of "ard$rcaaaabb"

c	$	a	b	c	d	r
C[c]	0 (no character is smaller than $)	1 (1 time $)	6 (5 times a plus 1 time $)	8	9	10

The function Occ(c, k) is the number of occurrences of character c in the prefix L[1..k].

Occ(c, k) of "ard$rcaaaabb"

	a	r	d	$	r	c	a	a	a	a	b	b
	1	2	3	4	5	6	7	8	9	10	11	12
$	0	0	0	1	1	1	1	1	1	1	1	1
a	1	1	1	1	1	1	2	3	4	5	5	5
b	0	0	0	0	0	0	0	0	0	0	1	2
c	0	0	0	0	0	1	1	1	1	1	1	1
d	0	0	1	1	1	1	1	1	1	1	1	1
r	0	1	1	1	2	2	2	2	2	2	2	2

These are two auxiliary tables used to construct the FM index.

Read mapping techniques: (2) Burrows Wheeler transform

The FM-index itself is a compression of the string L together with C and Occ in some form, as well as information that maps a selection of indices in L to positions in the original string T.

FM index is used e.g. by the tools Bowtie and BWA

Soap uses a different variant of BWT.

www.wikipedia.org

According to

http://pages.di.unipi.it/ferragina/Libraries/fmindexV2/index.html

The FM-index combines compression and indexing by encapsulating in a single compressed file both the original file plus some *indexing information*. The space occupancy of the FM-index is close to the one required by the best known compressors, like bzip2. But additionally to a compressor, the FM-index is able to efficiently support substring search operations, and the decompression of portions of the original file. Every such operation is executed on the FM-index by looking *only at a small portion of the compressed file*, thus requiring few *milliseconds* on a commodity PC over files of several megabytes.

Read alignment: features

Crucial default options:

- Maximum number of mismatches in the **seed** (default 2). The seed is "the first few tens of base pairs of a read." The seed part of a read is expected to contain less erroneous characters.
- Maximum number of **mismatches** in the read (2 to 10)
- **Seed length** (28 – 32).
- **Quality threshold**: It is equal to 70 for MAQ and Bowtie while it depends on the read length and the genome size for Novoalign.
- Splicing: This option is enabled for GSNAP.
- **Gapped alignment**: It is enabled for Bowtie2, GSNAP, BWA, Novoalign and MAQ while it is disabled for SOAP2.
- Minimum and maximum **insert sizes** for paired-end mapping: The insert size represents the distance between the two ends (0 to 500).

Hatem et al.
BMC Bioinformatics (2013) 14:184

One complication in read alignment is that we are not only looking at positions that align perfectly or exactly. Often, the two sequence may differ „a little bit" due to either the normal biological variation between an individual and the reference genome or due to technical sequencing errors.

The genetic difference between individual humans today is minuscule – on average about 0.1% of all positions (https://humanorigins.si.edu/evidence/genetics).

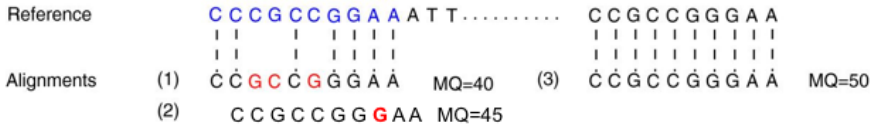
Based on this, we instruct the alignment algorithm to search for almost perfectly matching positions of read and reference genome and allow for a small given number of **mismatches**.

From Hatem et al paper: „*Seeding* represents the first few tens of base pairs of a read. The **seed** part of a read is expected to contain less erroneous characters due to the specifics of the NGS technologies. Therefore, the seeding property is mostly used to maximize performance and accuracy."

Read alignment: evaluation criteria

The sequence in blue is the original genomic position where the simulated read was extracted from. After applying sequencing errors, the read does not exactly match to the original location (3 mismatches marked in red).

3 possible alignment locations for the read with their mapping quality score (MQ).



Naïve criterion: only consider the alignment (1) as the correct alignment.

Hatem et al.
BMC Bioinformatics (2013) 14:184

Link to Hatem et al paper: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-184

Alignment (3) is a perfect match of the altered string to an alternative position in the genome. In this position, the match achieves the highest mapping score MQ = 50. MQ stands for „mapping quality“.

Alignment (2) is a position shifted by one base pair where it has only 1 mismatch G-A and a slightly better score (MQ = 45) than in the original position (MQ = 40).

The naïve criteria would judge the tool as incorrectly mapping the read if the tool returned either alignment (2) or (3) while in fact it picked a more accurate matching.

Mapping quality scores (MQ)

Mapping Quality Scores quantify the probability that a read is misplaced. They were introduced by Heng Li and Richard Durbin and are usually reported on a Phred scale.

For a particular short sequence read, consider its best alignment in the genome. *SUM_BASE_Q(best)* : sum of base quality scores at mismatched bases.
Also, consider all other possible alignments for the read. *SUM_BASE_Q(i)* : sum of base quality scores at mismatched bases for alignment *i*.

Then, the **mapping quality** is defined as:

$$Mapping_{Quality} = -10 \times log_{10}\left(1.0 - \frac{10^{-SUM\_BASE\_Q(best)}}{\sum_i 10^{-SUM\_BASE\_Q(i)}}\right)$$

The quantity $10^{-SUM\_BASE\_Q(i)}$ tries to approximate the probability of generating a particular read when alignment i is used as template. E.g., if there is a single mismatch with base quality 20, we approximate the probability of sampling the read as ~0.01; with two mismatches with base quality 20, the approximation becomes ~0.0001.

Li H, Ruan J, Durbin R. (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research* **18**:1851-8.

Link to the Li & Durbin paper: https://pubmed.ncbi.nlm.nih.gov/18714091/

Phred values were introduced by Ewing & Green (https://genome.cshlp.org/content/8/3/186.full) as quality values for base calling:

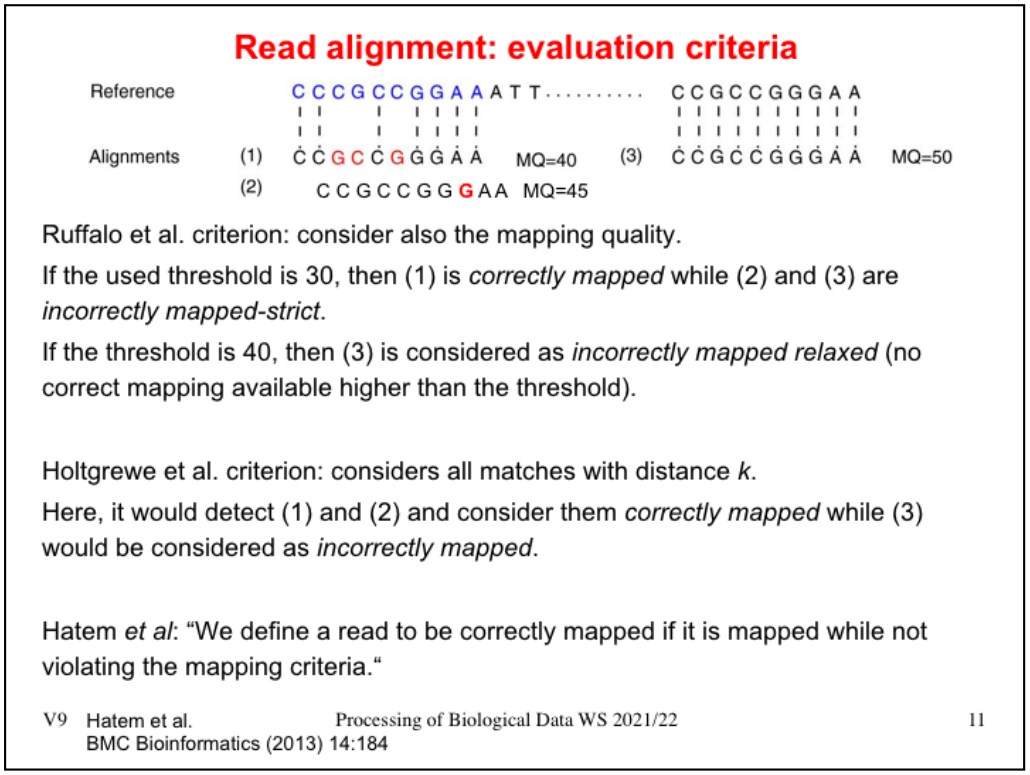
$q = -10 \times \log_{10}(p)$

where *p* is the estimated error probability for that base-call. Thus a base-call having a probability of 1/1000 of being incorrect is assigned a quality value of 30.

Text of this slide was copied from

https://genome.sph.umich.edu/wiki/Mapping_Quality_Scores

For paired end reads, we calculate SUM_BASE_Q as the sum of base quality scores at mismatched bases for both reads.



Ruffalo et al. (https://academic.oup.com/bioinformatics/article/27/20/2790/201940) classify the accuracy of the mapping(s) of a read as follows.

Correctly mapped read (CM): the read is mapped to the correct location in the genome and its quality score is greater than or equal to the threshold.

Incorrectly mapped read—strict (IM-S): the read is mapped to an incorrect location in the genome and its quality score is greater than or equal to the threshold.

Incorrectly mapped read—relaxed (IM-R): the read is mapped to an incorrect location in the genome, its quality score is greater than or equal to the threshold and there is no correct alignment for that read with quality score higher than the threshold.

Read alignment: throughput for simulated data

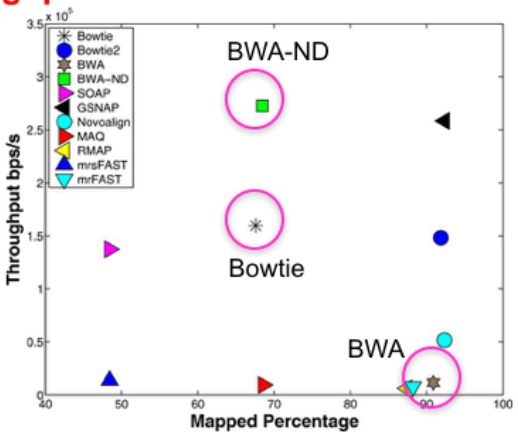
Task: map 1 million reads of length 125 extracted from the Human genome using `wgsim` with 0.09% SNP mutation rate, 0.01% indel mutation rate, and 2% uniform sequencing error rate.

Each tool was used with its own default options.

Bowtie only maps 68% of the reads, but achieves high throughput.

BWA maps 91% of the reads, but 15 x lower throughput.

However, when used with the same options as Bowtie, BWA achieves even a higher performance.

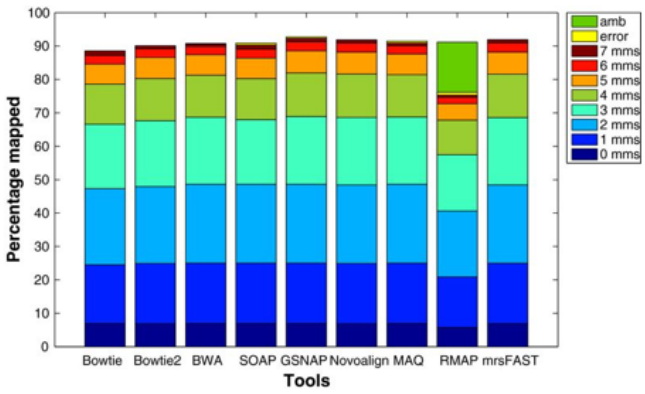


BWA-ND refers to BWA's results while using Bowtie's default options which are 2 mismatches in the seed, 3 mismatches in the whole read, and no gapped alignment.

Shown are the mapping results using the default options of each tool. The tools try to use the options that yield a good performance while maintaining a good output quality.

For instance, Bowtie achieves a throughput of around $1.6 \cdot 10^5$ bps/s at the expense of mapping only 67.58% of the reads. On the other hand, BWA maps 91% of the reads at the expense of having only a throughput of $0.1 \cdot 10^5$ bps/s. Additionally, SOAP and mrsFAST look like they provide the smallest mapping. However, they are only allowing 2 mismatches while other tools such as mrFAST and GSNAP are allowing more than 5 mismatches. Therefore, using only the default options to build our conclusions would be misleading. Indeed, further experiments show that BWA obtains a high throughput when allowed to use the same options as Bowtie. Moreover, BWA achieves a higher throughput than Bowtie in other experiments. Therefore, it is important to use the same options to truly understand how the tools behave.

Read alignment: number of allowed mismatches



Mapping 1 million reads of length 125 extracted using `wgsim` from the Human genome while allowing up to 7 mismatches and a quality threshold of 140.

-> most tools (except RMAP) achieve similar mapping rates.

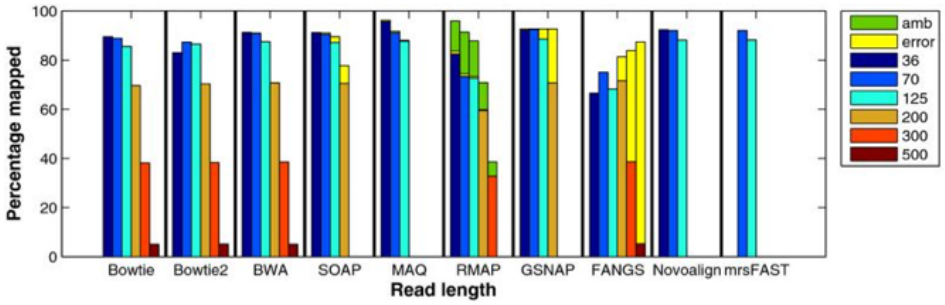
Hatem et al.
BMC Bioinformatics (2013) 14:184

For synthetic data generated with the software wgsim, quality thresholds of 60, 80, 100, 120, and 140 should correspond to 3, 4, 5, 6, and 7 mismatches. Here, all tools were allowed a maximum of 7 mismatches while using a quality threshold of 140. The figure shows that the tools map the reads with the same maximum number of mismatches while having similar mapping rates.

The differences in the mapping rates shown in the previous slide are due to the pruning of the search space done by the default options for some of the tools. In addition, other tools incorrectly mapped some of the reads causing an increase in the mapping percentage.

From the throughput point of view, the tools behave differently. For instance, Bowtie, MAQ, RMAP, and mrsFAST are able to maintain almost the same throughput while the throughput increases for SOAP2 and GSNAP and decreases for BWA. The degradation in BWA's performance is due to exceeding the default number of mismatches leading to excessive backtracking to find mismatch locations.

Read alignment: effect of read length



The effect of changing the read length from 36 to 500. The reads were extracted from the Human genome.

Longer reads tend to have **more mismatches**. For a fixed number of mismatches, the read length decreases the percentage of mapped reads.

Hatem et al.
BMC Bioinformatics (2013) 14:184

Longer reads tend to have more mismatches beside requiring more time to be fully mapped. In general, for a fixed number of mismatches, increasing the read length decreases the percentage of mapped reads. Therefore, the aim of this experiment is to understand the read length effect.

The figure shows that the mapping percentage decreases with the increase in the read length while the *error* percentage increases.

Bowtie, Bowtie2, and BWA were the only short sequence mapping tools that managed to map long reads. In particular, the max read length was 128 for MAQ, 300 for RMAP, and 200 for GSNAP, 199 for mrsFAST, while SOAP2 took more than 24 hours to map the reads with length 300 and hence not reported.

From the throughput point of view, tools do not maintain the same behavior. For instance, the throughput of Bowtie and SOAP2 decreases for long read lengths. This is due to the backtracking property and the split strategy used by Bowtie and SOAP2, respectively, to find inexact matches.

Read alignment: SNP calling with different mappers

Tools	accurately detected SNPs
Bowtie	1171
Bowtie2	2035
BWA	2067
SOAP2	1941
Novoalign	941
GSNAP	2602

Here, the tools were used to map an mRNA dataset of 23 million reads extracted from the Spretus mouse strain.

Then Partek was used to detect SNPs against mouse genome version mm9.

A quality threshold of 70 was used for Bowtie and Novoalign while the remaining tools were allowed 5 mismatches.

GSNAP detected the largest number of accurate SNPs while Novoalign detected the smallest.

This example illustrates that using a different mapping tool can greatly affect the number of obtained results.

Read alignment: conclusion

Mapping of short sequences is still subject of active development.

Genome indexing tools performed better than read indexing tools.

In general, there is no *best tool* among all of the tools; each tool was *the-best* in certain conditions.

Hatem et al.
BMC Bioinformatics (2013) 14:184

(2) Variant calling benchmark

-> Accurately detecting SNPs is critical e.g. for **medical diagnostics**.

Genome in a Bottle (GIAB) consortium:

public-private-academic consortium to develop the technical infrastructure (reference standards, reference methods, and reference data) to enable translation of whole human genome sequencing to clinical practice.

GIAB generated a set of highly confident variant calls for one individual in the 1000 Genome project:

they integrated 14 variant data sets from 5 NGS technologies, 7 read mappers and 3 variant calling methods, and manually cleaned up discordant data sets.

This highly accurate set of SNP and indel genotype calls can be used as **gold standard** variant genotype data set for systematic comparisons of variant callers.

Hwang et al., Scientific Reports 5, 17875 (2015)

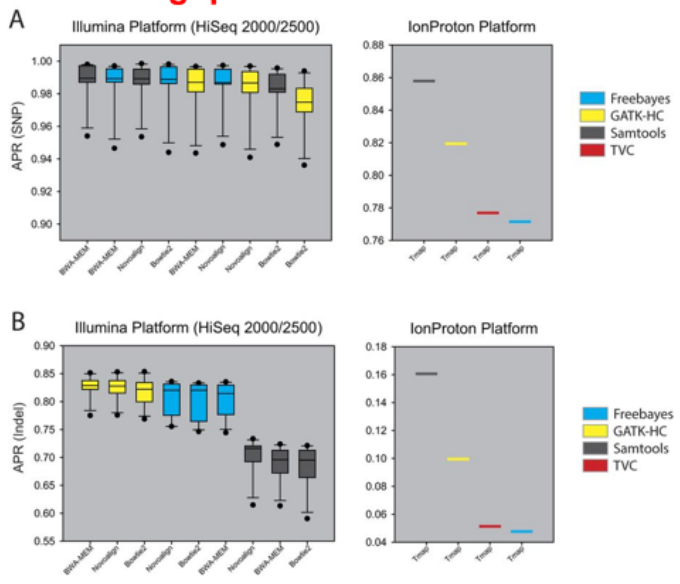
Website of the GIAB consortium: https://www.nist.gov/programs-projects/genome-bottle

GIAB publication: https://www.nature.com/articles/sdata201625

Variant calling: performance

Performance of variant calling pipelines measured by APR for **(A)** SNP and **(B)** indels.

APR: Area under a precision-recall curve.



Coverage 47 – 299 x

< 10 x

Hwang et al., Scientific Reports 5, 17875 (2015)

Hwang et al paper: https://www.nature.com/articles/srep17875

To compare the overall performance among thirteen pipelines, the authors compared the distributions of APR scores of multiple data sets for each pipeline on SNPs and indels.

The Ion Proton data set has much lower exome coverage (<10 ×) than those of Illumina data sets (43.6 × –298.5 ×).

For SNP variant calls, BWA-MEM-Samtools pipeline showed the best performance and Freebayes showed good performance across all aligners for both Illumina platforms.

For Ion Proton data, Samtools outperformed all other callers, including TVC, which is the Ion Proton's own variant calling method. Interestingly, the best variant caller of each data set varies. This observation of variation in best performed pipelines across data sets clearly demonstrates a data-specific effect of benchmarking results. Therefore, benchmarking performance of each variant calling pipeline needs to be based on multiple data sets to avoid misleading conclusions. The tested variant pipelines showed larger performance difference in calling indels. For indel calls, GATK-HC with any aligner outperformed Freebayes and Samtools on both Illumina platforms, while Samtools performed best on Ion Proton data. Although TVC is the official variant caller for Ion Proton data, it performed no better than other

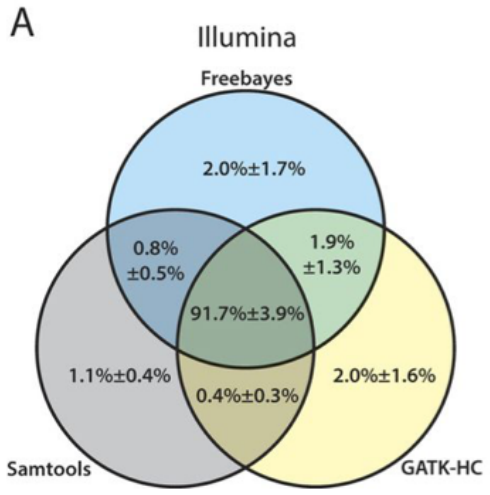
callers on both SNPs and indels.

Variant calling: consistency

Mean percentage with standard deviation of confidence variant calls with quality ≥ 20 for Illumina data sets.

-> Generally good agreement (92% overlap of results).

For low coverage IonProton data, the overlap is only 15%.



Hwang et al., Scientific Reports 5, 17875 (2015)

The authors then assessed the concordance (overlap) among the four variant callers for each NGS platform.

For Illumina data sets, they observed ~92% of concordance among the variant calls by three variant callers (see GATK-HC ∩ Samtools ∩ Freebayes) based on the average score of data sets. Concordance levels among variant calling pipelines varied across the data sets (82~97% overlap of called variants). These results indicate that not only the variant calling pipelines but also the data sets affect concordance of the identified variants. Therefore, caution is advised in interpreting concordance levels based on a single data set.

For Ion Proton data set, four callers showed 15.5% of overlap for the same quality score threshold (see GATK-HC ∩ Samtools ∩ Freebayes ∩ TVC). This low overlap among called variants is likely to originate from the high false positive rates for calling indel variants by Freebayes and Samtools.

Variant calling: recommendation

The authors recommend the use of BWA-MEM and Samtools pipeline for SNP calls and BWA-MEM and GATK-HC pipeline for indel calls.

Low coverage data is not suitable for reliable SNP calling.

Indels are detected at lower accuracy than SNPs.

Hwang et al., Scientific Reports 5, 17875 (2015)

Concluding remarks by the authors.

(3) SNPs in 1000 Genomes project

The 1000 Genomes Project ran between 2008 and 2015 and created the largest public catalogue of human variation and genotype data up to date.

The goal of the 1000 Genomes Project was to find most genetic variants with frequencies of at least 1% in the populations studied.

http://www.internationalgenome.org/

In the third application, we wanted to characterize the locations of SNPs found in genomes. We used the largest public data source available, the 1000 Genomes project which in fact sequenced the genomes of around 2500 individuals from the countries marked on the map.

Data set

We used only the European super-population with 503 individuals and focused on **autosomes** (chromosomes 1 – 22). Genes on sex chromosomes X and Y were ignored.

We kept autosomal SNPs with a minor allele frequency larger than zero → SNP exists

allele : variant form of a given gene

major allele : most common variant

minor allele: second-most common variant

We removed:

- genes starting with "SNO" (small nuclear RNAs) or "MIR" (microRNAs)
- genes with CDS start equal to the CDS end

Neininger K, Marschall T, Helms V (2019).
PLoS ONE 14(4): e0214816

We focused on the European super-population with ca. 500 individuals.

The reason was that we also analyzed in parallel the 500 parent genomes from the „Genomes of the Netherland“ project. The results for both data sets were very similar (data not shown).

We felt that data for the European cohort from the 1000G project would be more compatible with the GoNL data.

Also, we omitted the sex chromosomes X and Y because they appear to behave differently from autosomes.

E.g. the International SNP Map Working Group (https://www.nature.com/articles/35057149) found that the sex chromosomes have a lower diversity than autosomes. They suggested that the lower rate of polymorphism on the X chromosome may be explained by a lower effective population size, a lower mutation rate or by strong selection acting on the sex chromosomes in males.

Also, we filtered for genes annotated to have more than one allele, excluded SNO and MIR genes, or erroneous genes.

Problem: there exist many overlapping genes

Shown is overlap between 3 human genes: *MUTH, FLJ13949*, and *TESK2*.

Dark boxes : coding sequence.

Light boxes : untranslated regions.

Table 1. Frequency of Different Types of Overlaps Between Protein-Coding Genes in Human and Mouse Genomes

	Human		Mouse	
	Overlapping genes	Genes with overlapping exons	Overlapping genes	Genes with overlapping exons
Total	774	542	578	455
Embedded	126 (16.28%)	15 (2.77%)	53 (9.17%)	7 (1.54%)
Tail to tail	414 (53.49%)	360 (66.42%)	314 (54.32%)	280 (61.54%)
Head to head	234 (30.23%)	167 (30.81%)	211 (36.51%)	168 (36.92%)
Involving coding sequence		299 (55.17%)		232 (50.99%)
Coding–coding overlap		57 (10.52%)		31 (96.81%)

Veeramachaneni et al.
Genome Res. (2004) 14: 280-286

Link to Veeramachaneni et al. paper:
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC327103/

According to Veeramachaneni et al., it is believed that 3.2 billion bp of the human genome harbor ~35,000 protein-coding genes. On average, one could expect one gene per 300,000 nucleotides (nt).

Although the distribution of the genes in the human genome is not random, it is rather surprising that a large number of genes overlap in the mammalian genomes.

Veeramachaneni et al. identified >774 gene pairs sharing a locus in the human genome and 542 in the mouse genome.

Overlapping genes

One could speculate that overlapping genes would be more conserved between species than non-overlapping genes because a mutation in the overlapping region would cause changes in both genes.

Then, one would expect that evolutionary selection against these mutations is stronger.

However, Veeramachaneni *et al*. found that this is not the case.

Overlapping human and mouse genes were similarly conserved as non-overlapping genes.

Veeramachaneni et al.
Genome Res. (2004) 14: 280-286

The origin of overlapping genes is not clear. Interestingly, the mutation rates in the overlap regions are similar to non-overlap regions.

How to deal with overlapping genes

In the case of overlapping genes, it is problematic to define the **genomic regions** because they have a different meaning for the 2 overlapping genes.

Therefore, we distinguished 2 cases:

(1) Overlaps where one gene is located **inside another gene**.

Such genes inside other genes were excluded from the SNP analysis.

(2) **staggered overlaps** (genes overlap partially).

We collected all genes with staggered overlap. From each “bundle", only one gene was selected randomly to avoid overlapping genes.

In total, about 5% of all genes were removed due to overlaps.

Neininger K, Marschall T, Helms V (2019).
PLoS ONE 14(4): e0214816

We wanted to analyze the location of SNPs with respect to certain genomic regions, see slide 27.

However, a SNP in an overlapping region may belong to different regions with respect to either of the two genes.

To avoid confusion, we excluded shorter genes that are located inside longer genes and we randomly selected one of the genes showing staggered overlaps.

Since we had „enough“ data for our analysis, we rather prefered to analyze a „purified“ data set.

Refseq

The Reference Sequence (RefSeq) collection at NCBI provides a comprehensive, integrated, non-redundant, well-annotated set of sequences, including genomic DNA, transcripts, and proteins.

RefSeq transcript and protein records are generated in different ways:

- Computation Eukaryotic Genome Annotation Pipeline
  Prokaryotic Genome Annotation Pipeline
- Manual curation
- Propagation from annotated genomes that are submitted to members of the International Nucleotide Sequence Database Collaboration (INSDC)

Research question:

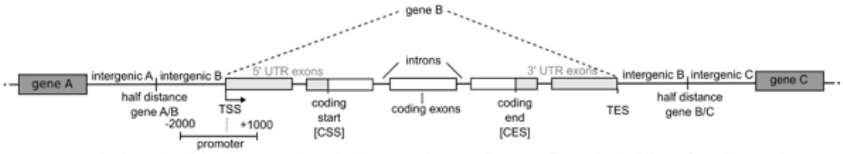
Are the **Single Nucleotide Polymorphism (SNP) frequencies** in different genomic regions similar to eachother or not?

https://www.ncbi.nlm.nih.gov/refseq/about/

V9 Processing of Biological Data WS 2021/22 26

The RefSeq annotation from NCBI is a very comprehensive, sophisticated and reliable annotation source for the location of genes and exons.

Definition of genomic regions



Every **gene** is located between two **intergenic regions**. Our definition for these is:

First intergenic region : interval between the transcription start site (TSS) of the considered gene and the mid-upstream position between this TSS and the transcription end site (TES) of the closest upstream gene.

Second intergenic region : defined analogously according to the TSS of the closest downstream gene.

Intragenic region of a gene : part between its TSS and its TES.

Gene **promoter** : region from 2000 bp upstream to 1000 bp downstream of the TSS.

Exons : intervals between the exon start positions and exon end positions (taken from UCSC genome browser).

5' UTRs : exonic segments between the TSS and the CSS

3' UTRs : exonic regions between the CES and the TES.

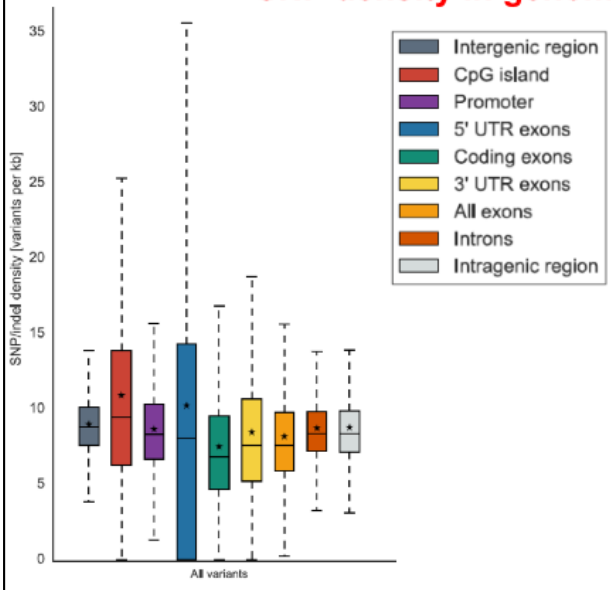
Introns : regions between the exonic gene parts.

Neininger K, Marschall T, Helms V (2019). PLoS ONE 14(4): e0214816

V9 Processing of Biological Data WS 2021/22 27

Based on the Refseq annotations, we analyzed the frequency of transition and transversion SNPs as well as indels in nine types (regions) of coding and non-coding genomic elements in the human genome.

SNP density in genomic regions



Neininger K, Marschall T, Helms V (2019). PLoS ONE 14(4): e0214816

Number of SNP variants per kb for different genomic regions.

→ lowest SNP density in coding exons (**green**)

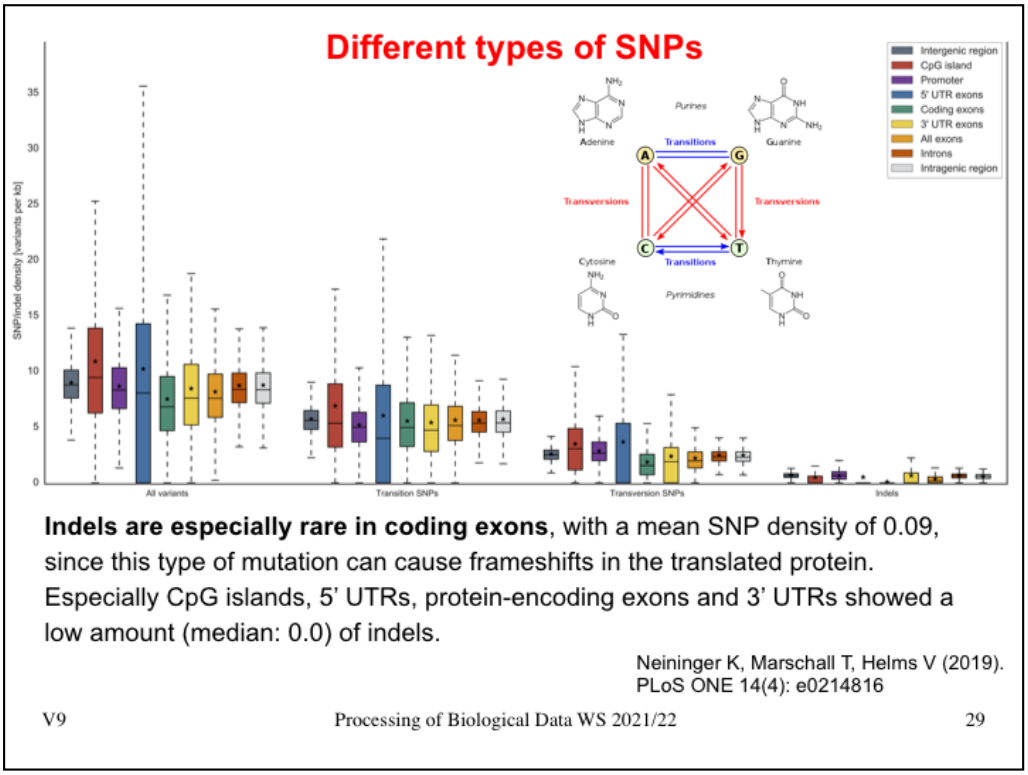
→ highest SNP density in CpG islands (**red**, due to frequent deamination of methylated cytosines into thymines)

Second-highest SNP density in intergenic regions (**grey**, low evolutionary pressure)

Considering 1000G data, median SNP densities were ~ 8–9 SNPs per kb for each genomic element and all variant types.

Protein-coding regions are conserved with a median SNP density of about 7 SNPs/kb for all SNP types. The boxplot for the 5' UTR contains some outliers with a maximum SNP density of up to about 35 SNPs per kb for 1000G data. This effect is due to the short 5' UTR length of 230 bp on average (median 180 bp).

Our findings of smaller SNP densities in genetically important gene regions such as coding exons or 5' UTRs are compatible with purifying selection to preserve their functionality.



Transitions refer to point mutations that change a purine nucleotide to another purine (A ↔ G), or a pyrimidine nucleotide to another pyrimidine (C ↔ T). Approximately two out of three single nucleotide polymorphisms (SNPs) are transitions. **Transversions** interchange a purine with a pyrimidine and are less frequent. This was also observed by us.

Indels might have more severe effects on transcription factor binding sites than base exchanges. Hence, the low frequency of indels in CpG islands might be related to a strict conservation of functional sequences within this genomic (regulatory) element especially in CpG islands in the promoter regions of the mammalian genes.

(4) Isoforms of genes

Gene isoforms are mRNAs that are produced from the same locus but are different in their

- transcription start sites (TSSs),
- protein coding DNA sequences (CDSs) and/or
- untranslated regions (UTRs),

All these processes may potentially alter gene function.

Alternative splicing (AS) of mRNA can generate a wide range of mature RNA transcripts.

It is estimated that AS of pre-mRNA occurs in 95% of multi-exon human genes.

There is abundant evidence for the expression of **multiple transcripts** in cells.

However, it is less clear whether these transcripts are expressed more or less equally across tissues or whether it would be biologically relevant to designate one transcript per gene as **dominant** and the rest as **alternative**.

www.wikipedia.org

Alternative splicing is a promiment mechanism to enlarge the complexity of gene regulation.

About 95% of all genes with more than one exon are alternatively spliced.

One important question is now whether (1) all these isoforms will be expressed in one tissue, (2) only some of them, or (3) only one of them.

Detect isoforms in proteomic data

Ezkurdia *et al*. re-analyzed 8 HT proteomics MS data sets.

At least 2 peptides were detected for 12 716 (63.9%) of the protein-coding genes but alternative protein isoforms only for 246 genes (1.2%).

→ the vast majority of genes had peptide evidence for just **one protein isoform**.

The isoform with the highest number of peptides was the **main proteomics isoform**.

A unique main proteomics isoform was identified for 5011 genes.

Ezkurdia et al J Proteome Res. (2015) 14: 1880–1887.

Probably this issue has not been completely settled yet.

For the moment, it is safe to assume that there will exist one major protein isoform in each tissue.

(5) Alternative translation: example TrpV6 channel protein

MUSCLE multiple sequence alignment of the translated 5′-UTR of TRPV6

Identical aa residues (compared with the human sequence) are *shaded*;

annotated N termini with the first Met^{+1} are in ***red***;

* : stop codon in frame

− : gap

Fecher-Trost et al. J. Biol. Chem. (2013) 288: 16629

The mammalian sequences upstream of the first AUG codon are conserved, but the one from rabbit contains an in-frame stop codon. In contrast, sequences from the other organisms contain several stop codons upstream of the annotated AUG and are not conserved. Sequence identity is highest among the 40 amino acids upstream of the first Met residue (position +1). This suggests that translation in mammals may start at a non-AUG

Now we come to something that you may not have expected.

Sometimes, protein translation may start at a non-AUG codon. This is called „alternative translation“.

Shown here is an alignment of the calcium channel protein TrpV6 from different species.

The red colored sequence region on the right is annotated in databases as the protein-coding region.

It is surprising to find that the sequence upstream of the translation start site is highly conserved and extends 40 amino acids upstream.

Alternative translation of human TRPV6

Nucleotide alignment of 5′-UTR TRPV6 sequences including the AUG triplet encoding the first methionine (*red*, +1) of the human protein.

Red, putative initiation sites;

underlined, STOP-codon in frame.

Experiments in the Flockerzi group (Medical department, Homburg) showed that translation starts at Thr^{-40} .

Fecher-Trost et al. J. Biol. Chem. (2013) 288: 16629

The group of Prof. Veit Flockerzi from Homburg discovered some years ago that the TrpV6 protein is 40 amino acids longer than they and the rest of the world previously thought.

In principle, this could have drastic consequences. Fortunately, for them, it turned out that the biological properties of the TrpV6 channel that they characterized for decades using a cloned version (that was 40 amino acids too short) were practically the same as those of the full-length protein.

HT discovery of alternative translation: ribosome profiling

Protocol resembles ChIP-Seq.

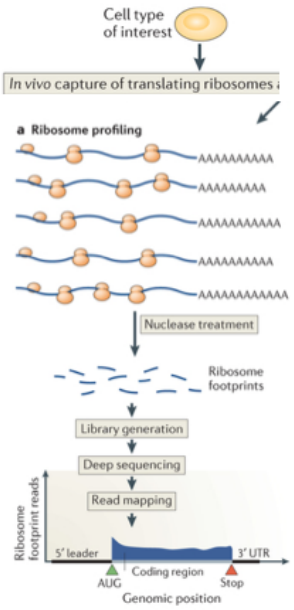
Translation is halted by applying ribosome inhibitors.

Isolate ribosome-bound mRNAs by size.

Then treat sample with a nonspecific nuclease.

This results in protected mRNA fragments termed **'footprints'**.

These ribosome footprints are isolated and converted to a library for deep sequencing.



Brar, Weissman, Nature Rev Mol Cell Biol 16, 651–664 (2015)

This slide explains the ribosome profiling protocol that was invented in the lab of Jonathan Weissman at Stanford.

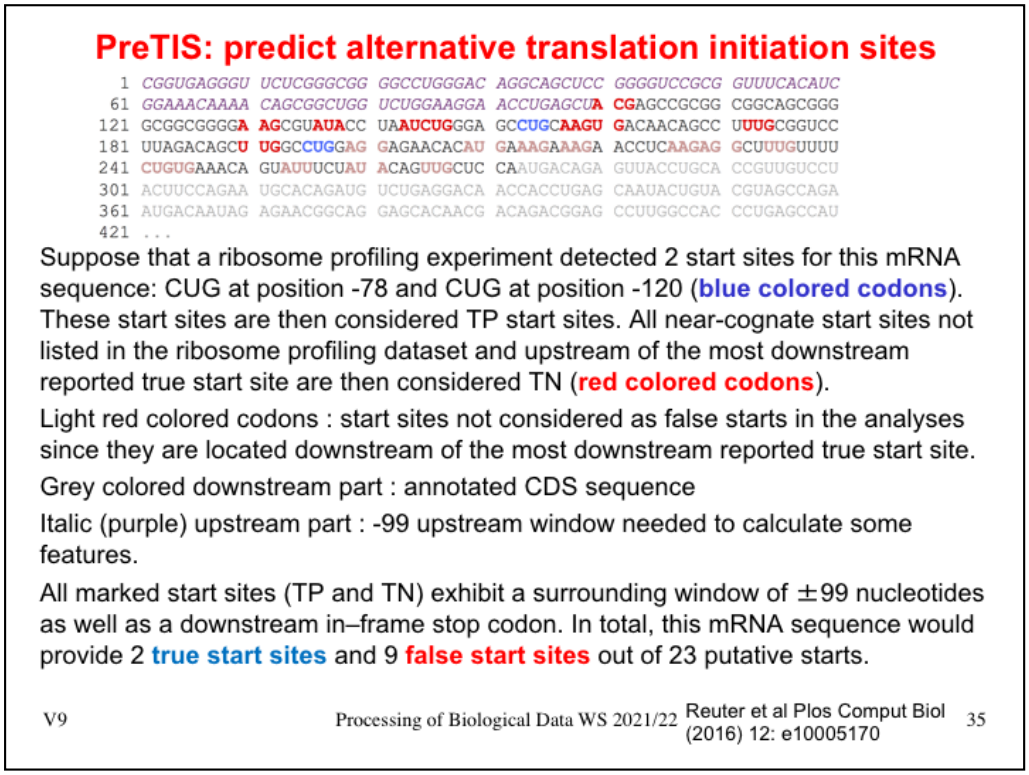
In an operational cell, ribosomes will constantly bind to mRNA messenger molecules and translate them into protein sequences.

To monitor the occupancy of ribosomes, one applies small chemical molecules that act as ribosome inhibitors and stall the further processing of mRNAs.

One can imagine that the conformations of the ribosomes get „frozen“ in a particular state, such as stopping a video clip.

This situation is shown in the figure below the writing „a Ribosome profiling“.

The rest of the protocol is very similar to the ChIP-seq protocol.



Which positions are considered as potential alternative translation initiation sites (aTIS)?

In this example, AUG at position 273-275 (colored light grey) would be the annotated translation start site in the database. All subsequent sequence is translated into protein.

But there are many potential alternative start sites upstream of this AUG that differ by one nuclotide. They are termed „near-cognate“ sites.

The first one is ACG at position 100-102 (colored red).

Let us assume that ribosome profiling detected two true start sites: CUG at position 153-155 (which means 120 positions upstream of the canonical start site) and CUG at position 195-197 (78 upstream).

These are colored blue. It is actually not easy to detect experimentally if both of them are used or if only the first one is used. We will ignore this complication.

All other alternative sites upstream of the first aTIS and between the two are assumed to be true negatives because they are apparently not used.

For the other aTIS candidates downstream of the second true positive aTIS site, we cannot make a statement whether they are also used or not because they are „overshadowed“ by the two aTIS sites upstream of them.

Data sets used for ML classifier

Cell line	Description	Genes	Start codons	TPs	TNs	Used for
HEK293	Human embryonic kidney cells	3,566	AUG and near-cognate	4,482	49,520	Human prediction model
HEK293	Human embryonic kidney cells	391	AUG	332	447	Validation set
Mouse ES	Mouse embryonic stem cells	1,632	AUG and near-cognate	3,009	19,864	Mouse prediction model

Three different datasets were used in this study to establish a human and mouse prediction model and to cross-validate the regression models. numbers indicate the filtered start sites used in the prediction approach.

doi:10.1371/journal.pcbi.1005170.t001

We only included curated mRNA sequences with available mRNA RefSeq identifier (starting with NM_).

Raw data is very unbalanced (number of TPs and TNs very different)

→ need to balance data sets (select random TN data points)

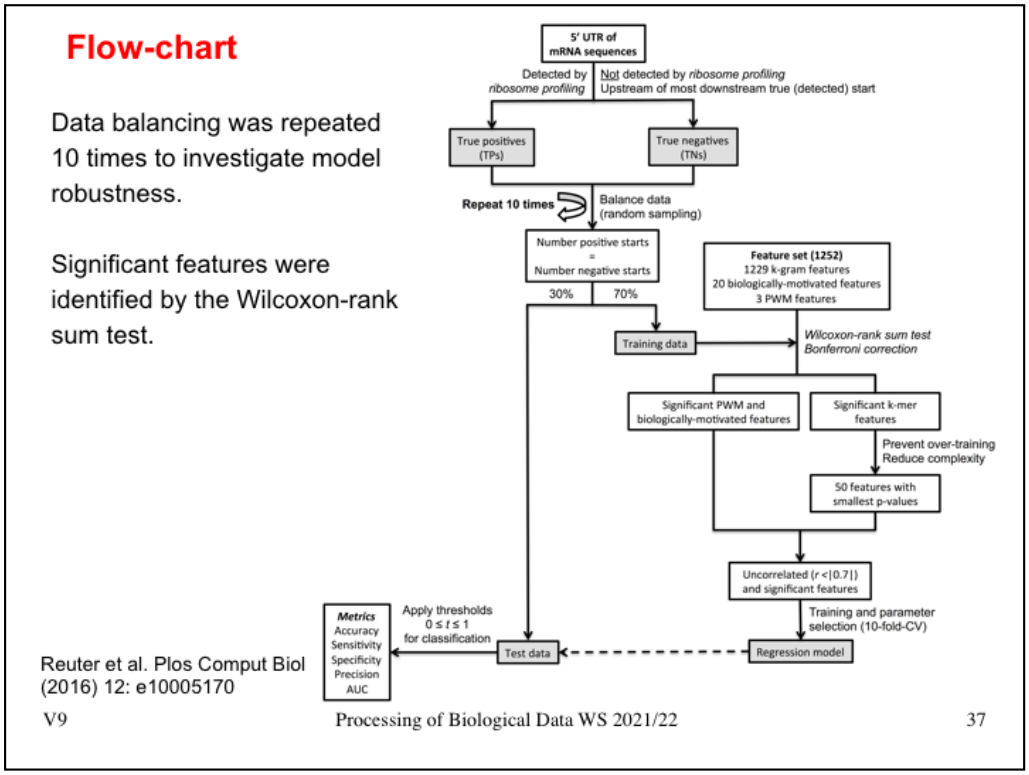
Reuter et al Plos Comput Biol (2016) 12: e10005170

These were the available suitable ribosome profiling data sets in 2015 when we conducted this project.

The number of TNs is 7-12 fold larger than TPs. Therefore, we downsampled the TNs by randomly selecting the same number of data points.

If one would not do this, a „successful“ classifier could alway predict „negative“ and would achieve around 90% accuracy on an imbalanced test set, simply because there are about 10 fold more negatives in the full data set.

If one would balance the test set (50:50), then this classifier would fail completely.



This is the flowchart used to train a classifier that predicts which candidate alternative start sites are used and which ones are not.

As true positives, we used the mRNA sequences detected in ribosome profiling to be bound to ribosomes.

As true negatives, we used all remaining mRNA sequences that were not detected.

Note that both steps include assumptions. There may be different reasons why mRNAs appear to be bound although they are in fact not translated, and the opposite.

Then, we compute a large number of features for the elements of both sets. These will be explained on the next slide in more detail.

We select the 50 (out of 1252) features showing the largest differences between both datasets in order to avoid over-training, and also check for correlation between them.

Features used

Mean value and standard deviation of the 44 features that were used in the best human model.

PWM : probability weight matrix

$$\mathrm{PWM}_{(nt,i)} = log\left(\frac{PFM_{(nt,i)}}{bg_{nt}}\right)$$

Entries of position–frequency–matrix (PFM) : sum of occurrences of a nucleotide at position *i* divided by the total number of sequences contained in *S*.

Reuter et al Plos Comput Biol (2016) 12: e10005170

	Feature	True starts	False starts	P-value
1.	**5' UTR length**	414.41±270.48	675.41±545.35	$< 10^{-310}$
2.	**5' UTR conservation**	0.4±0.16	0.33±0.16	8.2×10^{-190}
3.	**PWM positive**	2.75±1.5	-0.14±2.82	5.5×10^{-173}
4.	K-mer: upstream AUG	0.22±0.57	0.59±0.9	5.1×10^{-144}
5.	**5' UTR: percentage A**	0.18±0.05	0.2±0.05	9.6×10^{-100}
6.	**Kozak sequence context**	2.67±1.07	2.3±1.11	9.2×10^{-95}
7.	**Translational efficiency of flanking sequence**	83.75±20.11	77.12±21.4	1.1×10^{-83}
8.	K-mer: position -12 is C	0.13±0.34	0.3±0.46	2.7×10^{-77}
9.	K-mer: upstream Asparagine	1.25±1.37	1.61±1.61	4.0×10^{-43}
10.	K-mer: downstream AUG	1.14±1.15	0.92±1.1	9.2×10^{-41}
11.	K-mer: upstream A	17.24±7.43	18.81±7.89	4.0×10^{-40}
12.	K-mer: in-frame upstream Alanine	3.69±2.6	3.16±2.29	4.0×10^{-37}
13.	K-mer: upstream Alanine	10.27±4.5	9.38±4.6	6.2×10^{-37}
14.	**5' UTR: percentage G**	0.32±0.06	0.31±0.05	7.1×10^{-37}
15.	**Codon conservation**	0.23±0.42	0.12±0.32	3.2×10^{-36}
16.	K-mer: position -3 is A	0.31±0.46	0.2±0.4	3.4×10^{-35}
17.	K-mer: upstream CCG	2.98±2.43	2.56±2.31	7.1×10^{-34}
18.	K-mer: downstream CCA	2.04±1.54	1.75±1.45	1.1×10^{-32}
19.	K-mer: position -12 is A	0.3±0.46	0.19±0.4	4.0×10^{-32}
20.	K-mer: in-frame upstream Methionine	0.07±0.29	0.2±0.48	3.3×10^{-31}
21.	K-mer: upstream Arginine	12.15±4.34	11.33±4.64	1.5×10^{-29}
22.	K-mer: upstream Histidine	1.7±1.52	1.97±1.65	2.2×10^{-27}
23.	K-mer: GCC	6.4±3.87	5.77±3.75	1.1×10^{-25}
24.	K-mer: position 4 is G	0.37±0.48	0.28±0.45	2.3×10^{-25}
25.	K-mer: upstream Threonine	3.56±2.08	3.91±2.19	4.9×10^{-25}
26.	K-mer: upstream CGG	3.14±2.51	2.77±2.41	3.2×10^{-24}
27.	K-mer: upstream C	30.4±8.98	28.96±9.04	1.0×10^{-23}
28.	K-mer: position -2 is G	0.23±0.42	0.32±0.47	1.2×10^{-23}
29.	K-mer: upstream Stop	2.3±1.71	2.66±2.0	1.4×10^{-23}
30.	K-mer: UAG	1.34±1.2	1.57±1.35	5.6×10^{-23}
31.	K-mer: upstream CAU	0.58±0.85	0.73±0.95	3.4×10^{-22}
32.	K-mer: upstream Serine	9.44±3.29	8.93±3.14	5.7×10^{-22}
33.	K-mer: downstream Glutamine	3.57±2.01	3.26±1.88	2.4×10^{-21}
34.	K-mer: AGG	4.29±2.51	4.7±2.69	2.1×10^{-20}
35.	K-mer: AGC	4.4±2.43	4.02±2.19	2.1×10^{-20}
36.	K-mer: downstream ACC	1.45±1.26	1.27±1.17	2.0×10^{-19}
37.	K-mer: UAA	1.22±1.42	1.51±1.76	6.2×10^{-19}
38.	K-mer: downstream Proline	9.3±5.63	8.56±5.47	3.5×10^{-18}
39.	K-mer: upstream CAA	0.75±0.92	0.91±1.06	1.3×10^{-17}
40.	K-mer: in-frame upstream Histidine	0.54±0.77	0.67±0.87	1.7×10^{-17}
41.	K-mer: upstream GAU	0.63±0.85	0.77±0.96	2.1×10^{-16}
42.	K-mer: in-frame upstream GCC	1.21±1.4	1.02±1.22	6.7×10^{-16}
43.	K-mer: in-frame upstream GCG	1.14±1.42	0.97±1.27	6.2×10^{-14}
44.	**PWM negative**	1.94±1.34	1.59±1.09	1.6×10^{-08}

Mean value and standard deviation of the 44 features that were used in the best human model (biologically-motivated and PWM features are shown in bold). All 4,482 true and 49,520 false start sites were considered for this analysis. All listed features showed significant differences between true and false start sites (P–values < 1.6×10^{-8}). Note that due to numerical reasons, very small p–values (< 10^{-316}) are represented as 0.0 in python programming language (*scipy version 0.17.0*). The PWM–scores are based on the test data (compare to Fig 4).

doi:10.1371/journal.pcbi.1005170.t003

These are possible features by which true translation start sites and false start sites may potentially differ.

Obvious criteria are the length of the 5‘UTR region and its conservation.

If the considered codon is actually a false start and real translation starts in front of it, the annotated UTR may be too long. This matches the observation that the 5‘UTR in front of false starts is much longer (675 nt) than in front of true starts (414 nt).

If a UTR regions is highly conserved, this also suggests that it may in fact be translated.

The k-mer counts are raw counts in a 99 nt upstream or downstream window from the central codon.

Evaluation

	Accuracy	Specificity	Sensitivity	Precision	AUC	Threshold
			HEK293			
Linear SVR	0.80±0.01	0.80±0.01	0.81±0.01	0.80±0.01	0.80±0.01	0.62±0.01
RBF SVR	0.82±0.01	0.81±0.01	0.83±0.02	0.82±0.01	0.82±0.01	0.55±0.02
Polynomial SVR	0.80±0.01	0.80±0.01	0.81±0.02	0.80±0.01	0.80±0.01	0.59±0.02
Linear Regression	0.80±0.01	0.80±0.01	0.81±0.01	0.80±0.01	0.80±0.01	0.55±0.01
			Mouse ES			
Linear SVR	0.75±0.01	0.75±0.01	0.76±0.01	0.75±0.01	0.76±0.01	0.65±0.03
RBF SVR	0.76±0.01	0.76±0.01	0.76±0.02	0.76±0.01	0.76±0.01	0.58±0.03
Polynomial SVR	0.75±0.02	0.75±0.01	0.76±0.02	0.75±0.02	0.75±0.02	0.62±0.03
Linear Regression	0.76±0.01	0.75±0.01	0.76±0.01	0.75±0.01	0.76±0.01	0.55±0.01

The prediction was repeated 10 times to evaluate the model robustness. Shown are the average performance measures.

doi:10.1371/journal.pcbi.1005170.t002

All human models perform very similarly with accuracies of about 80%
while the average performance of the mouse model is lower with average accuracies of about 76%,

Reuter et al. Plos Comput Biol (2016) 12: e10005170

Support vector regression gave only slightly better results than standard linear regression. Hence, we used the robust linear regression for the Webserver version of PreTIS.

The accuracies for the mouse data set were slightly lower.

Is model transferable to other species?

Performance of the best human HEK293 model applied to the mouse ES dataset

→ model is reasonably transferable,

suggests universal translation code

	Unbalanced datasets			
	Mouse ES		**Mouse ES**	
Threshold	$t = 0.54$		$t = 0.52$	
	TP	TN	TP	TN
Predicted positive	2,161	4,569	2,273	5,072
Predicted negative	848	15,295	736	14,792
Total	3,009	19,864	3,009	19,864
Accuracy	0.76		0.75	
Sensitivity	0.72		0.76	
Specificity	0.77		0.74	
Precision	0.32		0.31	

	Balanced datasets			
	Mouse ES		**Mouse ES**	
Threshold	$t = 0.54$		$t = 0.52$	
	TP	TN	TP	TN
Predicted positive	2,161	689	2,273	763
Predicted negative	848	2,320	736	2,246
Total	3,009	3,009	3,009	3,009
Accuracy	0.74		0.75	
Sensitivity	0.72		0.76	
Specificity	0.77		0.75	
Precision	0.76		0.75	

doi:10.1371/journal.pcbi.1005170.t004

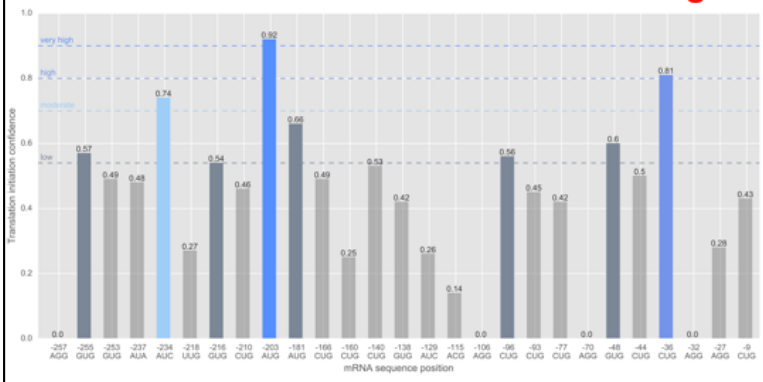
Reuter et al. Plos Comput Biol (2016) 12: e10005170

Interestingly, when applying the trained human model to the mouse embryonic stem cell data set from ribosome profiling, the results were almost as good as with the model trained on mouse data.

On the one hand, this suggests that the mouse data set is maybe not so good.

On the other hand, this suggests that the translation code in human and mouse is quite similar.

Alternative start codons of human gene GIMAP5



AUG at position -203 is a hot candidate with a very high confidence value of 0.92 of being a true start site.

Predicted start sites were subdivided into 4 confidence groups and highlighted by different colors and dashed lines:

- very high (best candidates with $c \geq 0.9$),
- high ($0.8 \leq c < 0.9$),
- moderate ($0.7 \leq c < 0.8$) and
- low ($t = 0.54 \leq c < 0.7$) initiation confidence c.

Reuter et al Plos Comput Biol (2016) 12: e10005170

As an example for the application of PreTIS, we show here the predictions for the human gene GIMAP5.

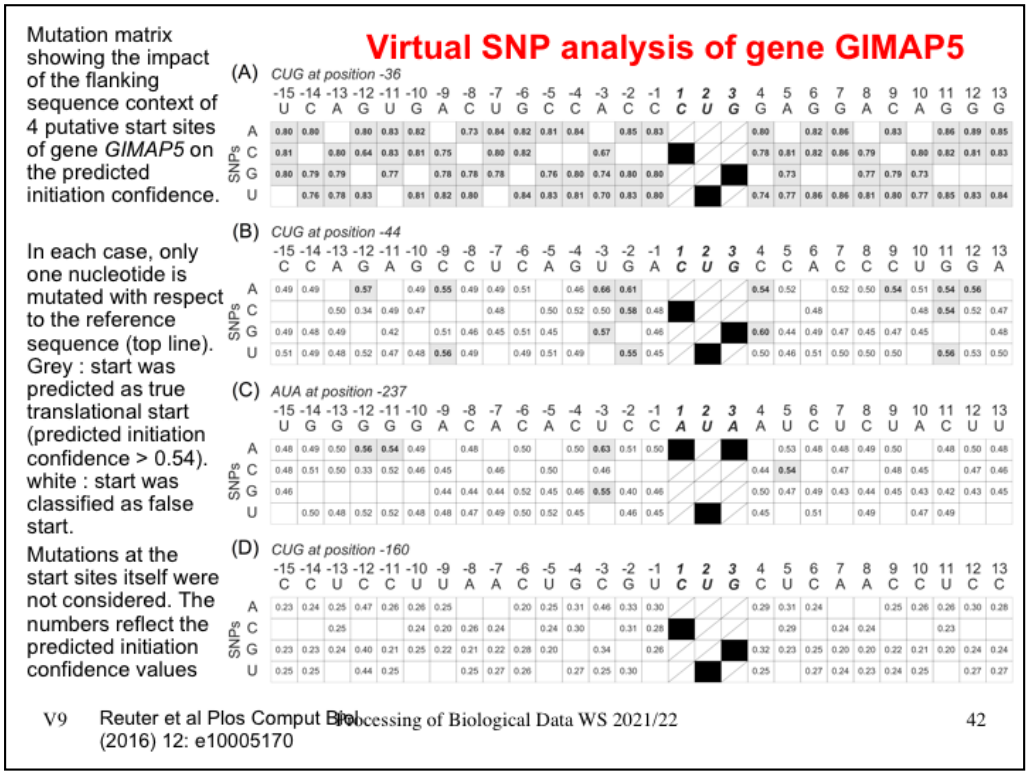
The annotated start site is a AUG codon at position 0 (right of the shown codons).

Listed are all alternative start sites upstream of the annotated translation start: AUGs and codons differing by one nt.

For each putative alternative start site, we show the „translation initiation confidence“ predicted by PreTIS‘s linear regression model.

The predictions are color coded according to the confidence score.

AUG at position -203 is assigned the highest score.



Here, we tested how the PreTIS prediction changes if one nt is exchanged to an alternative nucleotide that could result e.g. from a SNP.

In the upper line, CUG at position -36, has a PreTIS score of 0.81 (see previous slide).

The largest decrease to 0.64 would result from replacing G in position -12 by C.

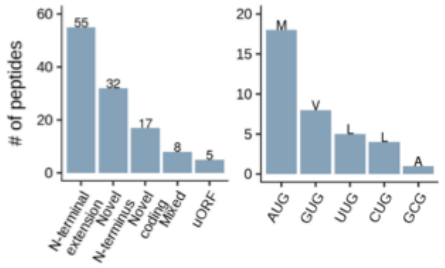
On the other hand, G at position +7 appears unfavorable. Replacing this by any other nt increases the score to 0.86.

The second line considers CUG at position -44 with PreTIS score of 0.50, which stands for low confidence.

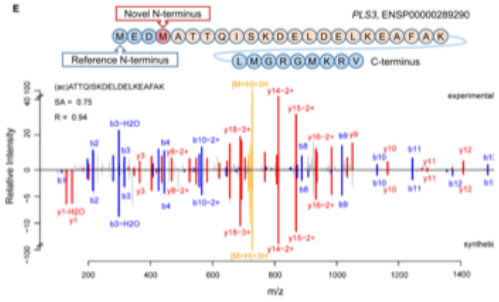
Some mutations, e.g. U->A at position -3 increase the score to a much better value of 0.66.

In the last line, CUG at position -160 has a low score of 0.25. Most mutations show practically no change, except for C->A in position -12 (0.47) and C-> A in position -3 (0.46).

MS of 29 human tissues



Non-canonical coding regions detected using proteomics data (left) and different alternative start codons identified by acetylated N-terminal peptides (right). The majority of cases are N-terminal extensions of annotated genes. All but one of the detected alternative translation start sites correspond to point mutations of the first base of the classical AUG codon.



Validation of a novel translation start site for the protein PLS3. (Top) Novel translation site position within the amino acid sequence context, (bottom) mirror plot of the tandem mass spectra of the endogenous N-terminally acetylated peptide (peaks pointing upwards) and the corresponding synthetic peptide spectrum (peaks pointing downwards).

Wang et al. identified in 29 human tissues 117 aTIS peptides mapping to 89 genes and 99 alternative translation start sites.

Fifty-five of these aTIS peptides represent 5′ N-terminal extensions of the original gene, 32 peptides represent novel (acetylated) N-termini downstream of the canonical start site, 17 represent frame-shifts potentially leading to an entirely new sequence, five peptides likely represent upstream ORFs (uORF) with a stop codon before the canonical start site and 8 peptides with mixed annotation.

One can validate the existence of aTIS peptides by comparing their spectra to synthetic peptides.

Panel E shows an example for a peptide (ac)ATTQISKDELDELKEAFAK derived from the actin-binding protein plastin-3 (PLS3).

Note the lacking y1 peaks (very left) for the experimentally detected (shorter) peptide.

(6) Removing sequence redundancy

Let's assume we want to know whether the **amino acid composition** of certain protein sequences differs in one genomic region from the other regions.

For example, we want to know whether **transmembrane (TM) segments** of membrane proteins are more hydrophobic than the rest of the protein sequence

To check this, we could simply analyze all protein sequences from NCBI, predict the TM segments in them and compare the amino acid compositions.

However, this search would likely be **biased** by

- what proteins have been sequenced and which ones not, and
- by duplicated sequencing experiments.

→ It is very important to **remove sequence redundancy** before such analyses!

This can be done by software tools such as CDhit or BlastClust

For many bioinformatics analyses, we need to process the considered data set and remove redundant sequences.

Here, we briefly explain for which applications this is important and mention how this can be done with tools such as CDhit or BlastClust.

BlastClust

blastclust -i infile -o outfile -p F -L .9 -b T -S 95

The sequences in "infile" will be clustered and the results will be written to "outfile".

The input sequences are identified as nucleotide (-p F); "-p T", or protein.

To register a pairwise match two sequences will need to be 95% identical (-S 95) over an area covering 90% of the length (-L .9) of each sequence (-b T) .

Another popular package is CD-HIT, see http://weizhongli-lab.org/cd-hit/

https://www.ncbi.nlm.nih.gov/Web/Newsltr/Spring04/blastlab.html

The Link given at the bottom of the slide links to a page explaining how BlastClust works.

But BlastClust is apparently no longer included in the latest release of the Blast program.

Take home messages

- Usually one **removes sequence redundancy** when correlating sequence features with properties of proteins etc.
- Check for **overlapping genes**
- Which translated variant is relevant? May want to try PreTIS

Today, we addressed several points that may be relevant if you analyze genomic data sets.