Genome Informatics for Evolutionary Immunology (GI4EI)

Final Report

By Shivam Kedia under Dr. Rintu Kutum

Introduction

Introduction

Next Generation Sequencing

Next Generation Sequencing (NGS) is a term that refers to a group of modern DNA sequencing technologies that are capable of sequencing multiple DNA fragments in parallel. These techniques have revolutionised the field of genomics, making it possible to sequence entire genomes quickly and at a lower cost than traditional sequencing methods. In this report, I provide an overview of the different NGS technologies and their applications.

There are several NGS platforms available, including Illumina, Ion Torrent, and PacBio.

Illumina is the most widely used platform, and it is based on a technique called Sequencing-by-Synthesis. It involves breaking DNA into small fragments and attaching them to a glass slide. Then, nucleotides which are labelled by their flourescence are added to the fragments one at a time. As each nucleotide is added, an image is taken. The colour of the fluorescence indicates which nucleotide was added. This process is repeated many times until the entire DNA sequence is reconstructed.

NGS technologies have many applications in the field of genomics, including whole genome sequencing, transcriptomics, epigenomics, and metagenomics.

- **Whole Genome Sequencing** involves sequencing an entire genome, which allows researchers to identify genetic variants and mutations that may be associated with diseases.
- **Transcriptomics** involves sequencing the RNA transcripts produced by a cell, which can provide insight into gene expression patterns.
- **Epigenomics** involves sequencing modifications to the DNA that do not involve changes to the underlying DNA sequence, such as methylation.
- **Metagenomics** involves sequencing the DNA from an entire microbial community, which can provide insights into the composition and function of the community.

Next Generation Sequencing technologies have revolutionised the field of genomics by allowing for rapid, accurate, and cost-effective sequencing of DNA. There are

several NGS platforms available, each with its own strengths and weaknesses. These technologies have many applications in the field of genomics, including those discussed above. As these technologies continue to improve, they will play an increasingly important role in various domains of life sciences sucha as medicine, agriculture, and environmental science.

RNA Sequencing

RNA Sequencing

Introduction

RNA sequencing (RNA-seq) is a powerful technology that allows researchers to investigate the expression of genes at a molecular level that first came about with the advent of Next-Generation Sequencing technology in the mid 2000s. It helps measure whether RNA is present in a biological sample, and the quantity of RNA present in the transcriptome. RNA-seq has become an increasingly popular tool in genomics research, enabling researchers to explore gene expression, alternative splicing, and RNA editing, among other processes.

Transcription

The transcriptome refers to the set of all RNA transcripts present in a cell sample. These include both coding (those that can be translated into a protein) and non-coding (those that cannot) RNA molecules. Transcripts are produced in a process of copying a segment of DNA into RNA known as ***transcription***.

Translation

Messenger RNA carries genetic information for protein synthesis through the process of ***translation***.

Genetic information is transported by the mRNA molecule in the form of a nucleotide sequence. These sequences are divided into codons by triplets of nucleotides that correspond to specific amino acids.

Transfer RNA (tRNA) matches anticodons on tRNA with mRNA codons and carrying specific amino acids.

Translation involves the following steps: mRNA divides into codons, ribosomes bind to mRNA, and read it in a 5' to 3' direction. Ribosomes create peptide bonds, forming a polypeptide that grows as it moves along the mRNA. Protein synthesis ends at a stop codon. Post-translation, polypeptides may undergo folding, modifications, and targeting to cellular compartments to become functional proteins.

Steps in RNA-Seq

RNA sequencing involves several steps, including sample preparation, library construction, sequencing, and data analysis. Here is a brief overview of the steps involved in RNA-seq:

1. **Sample Preparation**: RNA extraction is the first step in RNA-seq. Total RNA is isolated from the biological sample, such as cells or tissues, using a commercial kit.
1. **RNA Quality Control**: The quality and quantity of RNA samples are assessed using several methods, including spectrophotometry, gel electrophoresis, and capillary electrophoresis.
1. **Library Construction**: RNA-seq libraries are prepared by converting RNA into cDNA (complementary DNA) using reverse transcription. Then, adaptors are added to the cDNA fragments, followed by amplification and purification. The library preparation method can vary depending on the sequencing platform used.
1. **Sequencing**: The RNA-seq libraries are loaded onto the sequencer, and sequencing is performed according to the manufacturer's protocol. The sequencing method can vary depending on the platform used, such as Illumina or PacBio.
1. **Data Analysis**: The raw sequencing data is processed to remove low-quality reads, adapter sequences, and contaminating sequences. Then, the remaining reads are aligned to a reference genome or assembled *de novo* to generate a transcriptome. The transcript abundance is then estimated using bioinformatics tools, such as RSEM, Cufflinks, or DESeq2.

Transcriptome Assembly

The process of attributing genomic features to raw data containing sequences read from a biological sample (through RNA-seq) is called Transcriptome Assembly. The following two methods are used for this:

De Novo

This method doesn't need a reference genome to build the transcriptome. It is useful when the genome is unknown, incomplete, or significantly changed compared to the reference. However, using short reads for assembly has challenges. Firstly, it's important to decide which reads should be combined into longer sequences, known

as *contigs*. Secondly, it must handle errors and other issues that can happen during sequencing. Lastly, it should be computationally efficient.

De Bruijn graphs, used by assemblers like Trinity, Oases, Bridger, and rnaSPAdes, break reads into smaller sequences and organize them efficiently.

The main algorithm used for assembly has shifted from overlap graphs to de Bruijn graphs. Overlap graphs were used with older sequencing methods, but they don't work well with the large number of reads generated by RNA sequencing.

Using paired-end and long-read sequencing alongside short reads can help overcome some of the limitations. Metrics like median contig length, number of contigs, and N50 are used to evaluate the quality of a *de novo* assembly.

Genome guided

This approach uses similar methods as DNA alignment, but it adds the complexity of aligning reads that cover non-continuous parts of the reference genome. These non-continuous reads come from sequencing spliced transcripts.

Alignment algorithms typically have two steps: first, aligning short portions of the read (seeding the genome), and second, using dynamic programming to find the best alignment, sometimes with the help of known annotations.

Software tools like Bowtie, TopHat, STAR, HISAT2, and GMAP use genome-guided alignment. The output of genome-guided alignment can be used by tools like Cufflinks or StringTie to reconstruct continuous transcript sequences.

The quality of a genome-guided assembly can be assessed using de novo assembly metrics and by comparing it to known transcripts, splice junction, genome or protein sequences using a couple different methods.

It is important to note that the quality of assembly can differ based on the metric used for evaluation. Assembly tools that work well for one sample or experiment may not work as effectively for another. Combining different approaches might be the most dependable strategy for achieving reliable results.

Applications of RNA-Seq

RNA-Seq has the ability to uncover new information about diseases, find markers for diagnosing and treating illnesses, identify potential pathways for developing drugs, and make genetic diagnoses. These results can be personalised for specific groups or individual patients, which could lead to more effective ways to prevent, diagnose, and treat diseases. However, the costs in terms of money and time, as well as the

need for a team of experts including bioinformaticians, doctors, researchers, and technicians, pose challenges in fully understanding and interpreting the large amount of data generated by this analysis.

Appendix

Weekly Reports

Tuesday - @January 31, 2023

- Learnt to access AC04-208 Lab computers
- Forklift, SSH
- Miniconda on local system
- `Fastqc` - Installed succesfully

```
conda create GI4EI
conda activate GI4EI
..
conda install -c bioconda fastqc
```

- `Multiqc` - Installed after adding channels

```
conda config --add channels defaults
conda config --add channels bioconda
conda config --add channels conda-forge
```

Thursday - @February 2, 2023

- Looked at RNAseq pipeline on https://nf-co.re/rnaseq
- How nextflow works - used to make pipelines portable
- Tried to install STAR (rnaseq) on M1 mac - ran into issues using `miniconda`
- https://github.com/alexdobin/STAR : Claims STAR is only installable on `x86_64` based machines

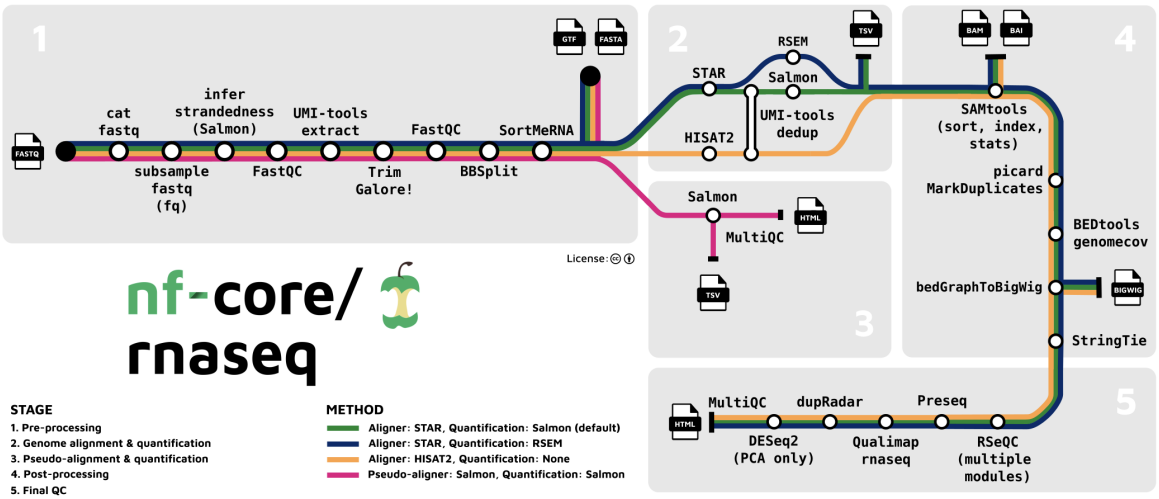
> 💡 Suggestion using `Anaconda` instead of `miniconda`

- Didn't work with `anaconda` either, the solution would be to use Homebrew, which installs it for the user, not limited to one environment

Tuesday: @February 7, 2023

Observations over weekend

- Cannot completely migrate `conda` envs using `conda env export environment.yml`, dependencies may not be resolved across different operating systems
- New machine available, `ssh` details recd



- Installed the following packages on new linux system using command: (refer https://nf-co.re/rnaseq)

```
conda install <package_name>
```

▼ Packages:

Stage 1:

- `fastq`
- `salmon`
- `fastqc`
- `umi_tools` **- causes conflicts, not installed in same env**
- `trim-galore`
- `bbmap`
- `sortmerna`

Stage 2:

- `star`
- `rsem`

Stage 3:

- `hisat2`
- `multiqc`

Stage 4:

- `picard`
- `samtools`
- `ucsc-bedgraphtobigwig`
- `bedtools`
- `stringtie`

Stage 5:

- `rseqc` - causes certain packages to downgrade, installed in different env
- `preseq` - causes certain packages to downgrade, installed in different env
- `qualimap`

```
conda install -c "bioconda/label/cf201901" qualimap
```

- `bioconductor-dupradar`
- `bioconductor-deseq2`

In order to avoid difficulties, packages have been installed in the following way:

Env Name	Packages
rnaseq-stage5	`rsqec`, `preseq`
umi-tools	`umi_tools`
GI4EI-test	all others

List available envs using `conda info --envs`

Activate envs using `conda activate <env_name>`

Thursday : @February 9, 2023

▼ The `fastq` file format

Files containing reads. https://en.wikipedia.org/wiki/FASTQ_format

```
@MN00537:51:000H2K25G:1:11101:2213:1092 1:N:0:9
CTCCAGTCCTTACTCCCATATCTAACCTCTTACCCCTACNTCATAGGTANACATTTTAATGAAT
+
FFFFFFFFFFFFAFFFFFFFF=FFFFAFFFFFFF/AFFF#FFFFFFFFF#FFFFFFFF
```

What it means

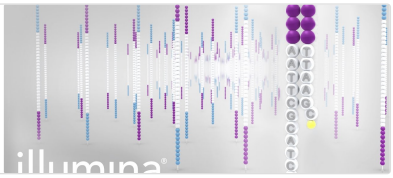
```
@SequenceID <Project Name>:<Seq_Lane>:<Location:of:seq> <pair lane>
The Read Itself
+ is placeholder
Phred Quality Score
```

▼ Illumina Sequencing by Synthesis

Illumina Sequencing by Synthesis

Explore the Illumina workflow, including sequencing by synthesis (SBS) technology, in 3-dimensional detail. Go from sample preparation, to cluster generation, to sequencing on a system flow cell with the proprietary SBS process, through to data analysis on the BaseSpace® Sequence Hub.

https://www.youtube.com/watch?v=fCd6B5HRaZ8&pp=ygUgaWxsdW1pbmEgc2VxdWVuY2luZyBieSBzeW50aGVzc2k=



Tuesday : @February 14, 2023

https://en.wikipedia.org/wiki/FASTQ_format

The `FASTQ` Format

It is a text based format for storing biological sequences and their quality scores. Usually nucleotide sequences are used. Sequence letter and quality scores are encoded with single `ASCII` characters.

It is a 4 line format, each line is a field:

1. Begins with @ character followed by Sequence ID, (optional) description
2. Raw Sequence
3. + character as placeholder, (optional) same Seq ID
4. Quality Scores for each letter in the sequence, reported in ASCII, range from 0x21 (!) to 0x7e (~).
   Refer to:

https://support.illumina.com/help/BaseSpace_OLH_009008/Content/Source/Informatics/BS/QualityScoreEncoding_swBS.ht

ASCII	Decimal	Phred Quality	Hex
!	33	0	0x21
“	34	1	0x22
}	125	92	0x7d
~	126	93	0x7e

Illumina Identifiers

Reads generated from Illumina products use a systematic identifier as sequence ID.

`@HWUSI-EAS100R:6:73:941:1973#0/1`

`@Instrument Name : Flowcell Lane : Tile No. : X-coord of cluster : Y-coordinate  #Index for multiplexed sample /Member of pair`

Illumina pipelines since v1.4 use #NNNNNN for multiplex ID, where the Ns are the multiplex tag

For further Illumina details refer: https://en.wikipedia.org/wiki/FASTQ_format

NCBI reads

`fastq` files from the NCBI read archive often have a description in the first and third Fields, they can hold read lengths and original identifiers.

Variations in `Fastq` Files

▼ **Quality**

Two equations used:

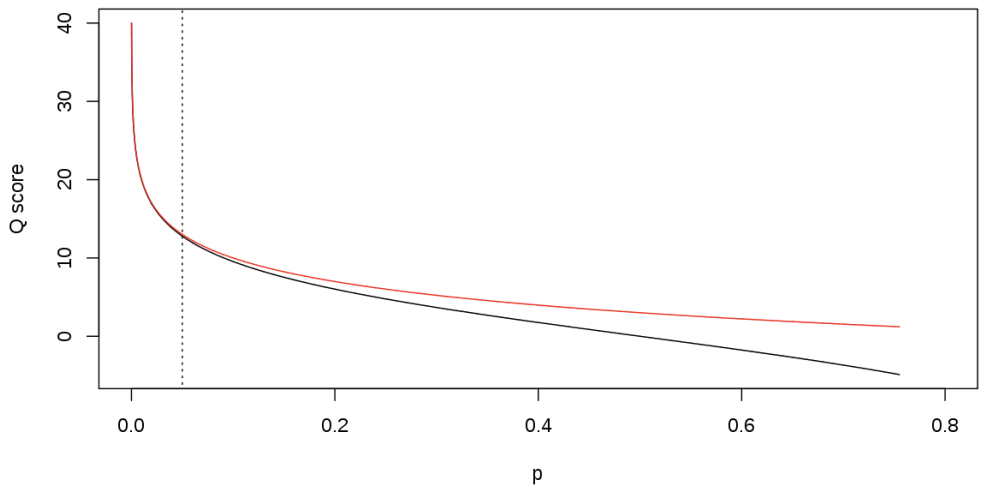
1. Phred Quality Score

$$Q_{Sanger} = -10\ log_{10} p$$

where ***p*** is the probability that the base is incorrect.

2. Odds

$$Q_{Solexa} = -10\ log_{10} \frac{p}{1-p}$$



▼ **Encoding**

Thursday : @February 23, 2023

- Wrote a skeleton python script:
  - input `fastq` file as command line argument
  - process arguments to validate input
  - read 4 lines from a file (equivalent to the data of one read)
  - convert the quality scores to human readable numbers

Tuesday : @February 28, 2023

- Completed previous python script to make it more robust
- Get all reads, one at a time
- Convert the read scores to integer
- Generate a matrix consisting scores corresponding to reads (in the order they occur in the file)
- Debug program
  - Was throwing a StopIteration exception, learnt how to catch that

Tuesday : @March 14, 2023

- Worked on some python code:
  - Given a position *k,* read all the bases in a `fastq` file at that position, and plot a bar graph of the number of different bases occurring at that position
  - Code on github
  - Tried to plot density plots and histogram, but these don't work for discrete data like base callss

Thursday : @March 23, 2023

- Obtained a sample `fastqc` file (received during the Bioinformatics class)
- copied it to Dr. Rintu's ANT PC using `scp`
- ran `fastqc` inside of a `conda` environment on the file obtained

Further Work (Coding)

- Referred to `seaborn` and `matplotlib` libraries for data visualisation using python
- Used `matplotlib` to generate bar plot of fastq file, containing the no. of nucleotide bases for a given position in a fastq file
- Further, wrote code to create a box_whisper plot identical to that seen in a FastQC report to report the per base sequence quality
  - Read documentation for various plots
  - First tried using `catplot` from seaborn and various other libraries
  - Used `numpy` to convert data read from `fastq` file to data types required for easier processing, involved use of various numpy matrix's functions such as Transpose
  - Narrowed down on `box` plot type from `matplotlib`
  - Learnt about various attributes in the plots such as:
    - Subfigures
    - Span
    - Plot parameters
    - Boxplot-specific properties
    - `patch_artist` and custom colouring a plot

- Coloured sections of a plot to make it identical to that in the fastqc report: linewidth, boxcolours, border colours, background colors, etc.

Smart Microscopes

Shivam Kedia

Pitfalls of Traditional Microscopy Techniques

- Requires trained humans - increases time spent by people handling equipment
- Statistical Confidence of the data generated meets scientific standards by a bare margin
- Resulting Images can be biased by the expectation of the human

Possible Advantages of Smart Microscopy

- Mechanised handling of equipment - resulting in higher throughput
  less human involvement -> more work done in less time

- Higher Quality Data - Smart Microscopes can auto-adjust based on the study subjects and lead to higher quality images - there is less human error - directly a result of higher throughput due to lack of human involvement

Developments in Genome Informatics

Need for Advanced Genome Informatics

- New Sequencing technologies have been churning out large datasets at unprecedented rates
- This data has the potential to give insight into much more sophisticated genetic studies -> creates a need for advanced genomic data analysis techniques

Potential Areas of Development

- So far, reference genomes have been used to work with genomic data
- This has certain drawbacks:
  1. Inefficient for Statistical Analysis (since biased towards references used)
  2. Does not help understand genomic data that have very large variation with respect to the reference genome
- Newer models have shown promising results, lot of scope for more powerful models

Techniques with Higher Quality output can be worked upon

- Present techniques are limiting the quality and quantity of data
- Trade-off between multiple parameters
- Newer strategies look promising

Importance of Refining Genomic Analysis Techniques

- Study of genomes is largely contingent upon the way experimental data is processed and analysed - this is why more sophisticated genome informatics is required
- Advanced analytic techniques can help further research into uncharted territory which was limited by the quality of data available after analysis and the cap on the size of raw datasets that could be processed using traditional models

Need for Principles for Data Stewardship

- Variety of Scientific Data standards available
- Causes development of standard-specific tools
- Leads to poor automated recognition of data generated by studies and their consequent reuse
- There is a need for principles that standardise the way data is stored and handled

- FAIR - Finding, Accessing, Integrating, Reusing

- The way data is deposited and stored should be according to the above 4 principles to make it more accessible

- Databases should be annotated properly to enable them to be searched generally, because specific technologies meant to parse data cannot handle the increasing diversity of what's becoming available

- Research output should also be handled according to FAIR principles to increase its availability and consequent usefulness to the community

FAIR Principles

Scientific Data Handling

Shivam Kedia

24th Dec, 2022.

https://www.nature.com/articles/sdata201618

Need for Data Handling Guidelines

- Research is a contribution to posterity, lot's of resources are spent on it.
- The way in which data coming out of research is handled causes difficulties in maximally utilising research output.
- Data must be handled in a 'good' way to make it useful for later, but we don't yet completely know what a 'good' way is.
- That's where FAIR principles come in.

What is Data Stewardship

- Proper collection, annotation and long-term storage of data.
- Research outputs are valuable assets, so it is important to make sure they are useful in the long-term.
- They should be discoverable and usable for further research.

Why FAIR

- There are numerous standards prescribed for the storage of Data.
- Both machines and humans use available databases and tools to find relevant data.
- The impact of a publication is it’s ability to be found, accessed, applied and reused.

Presence of Various Data Storage Standards and Formats

- Special-Purpose Databases exist to hold specific kinds of data.
- Not all types of data can be stored and (later used) using such special purpose repositories.
- These types of data that don’t necessarily belong to a popular special-purpose category are equally important.
- There exist more general-purpose databases to allow the storage of such kinds of data.
- There is no universally prescribed format to sift through such databases.

What are FAIR principles

- Minimal set of community-agreed guidelines
- Allow easier discovery, access, integration and re-use of research output while giving due credit
- Focus on making the work FAIR for both machines and people

FAIRer Data

For Humans

- Humans are better are at processing contextual cues, and can intuitively make sense of the intent of a digital object.

For Machines

- Humans are slow, and their search scope is limited.
- Machines need to capable of making decisions when different types data, formats, topics and access protocols are encountered.
- They need to first, make sure the data is contextually applicable, and second, the machine knows how to use or ‘parse’ it.
- The presence of a multitude of standards and their specific-parses makes it difficult for machines.

How to FAIR

- These are domain-independent guidelines, and generalised principles that when applied to scholarly output can help maximise their benefit.
- Implementation:
  - F - Thorough Metadata, Indexing
  - A - Availability of multiple formats
  - I - Presence of data in interchangeable formats for use by multiple machine-users
  - R - Unique, permanent links to research objects, to easily identify and reuse data while giving credit

FAIR Principles in AI and Biopharma

Applications of FAIR and shortcomings

Shivam Kedia

Considerations towards making data FAIR

Source:
https://frontlinegenomics.com/a-guide-to-the-fair-principles-in-biopharma/

Making Existing Data FAIR

1. Go through existing data - Examine its structure, source and identification methodologies.
2. Describe dataset using clear, specific vocabulary that is machine actionable
3. ‘Link’ these descriptive vocabularies to the dataset
4. Assign a license and metadata
5. Publish data

Metadata

Data about the Data

- Describes features of the dataset:
  context, quality, condition and characteristics
- Making FAIR Metadata is important. It helps even when the data it describes is not FAIR.
- If metadata is FAIR, the dataset inherits the advantages of FAIR and the data is easier to make sense of based on the metadata

Persistent Identifiers

Making sure that data is Findable

- These are two part references to a digital-object.
- The first part contains a string which is the unique identifier to an object.
- The second part is a service that links the unique identifier to its actual location on the internet.
- E.g : DOIs, URLs, PURLs,

Authorisation and Authentication

Procedures to Access data

- Metadata helps make the data findable.
- Persistent identifiers tell us where the data can be found.
- In order to access data, there are protocols followed. Not all data is open-to-all.
- Authentication ensures the data is being accessed by only those allowed (authorised) to access it.
- Authorisation determines the extent to which any data set is accessible to a given user/agent.

Making Data Interoperable

Speeding up the discovery and use of research data

- Well-known formats usable across systems should be used along with rich, standardised metadata.
- Controlled Vocabularies: Organised arrangements of words to optimise searching
- Ontologies: Organise information in Structural frameworks to extract relevant data
- Using README files: text files introducing a repository, clearly articulating it's objective, use and context

FAIR in Biopharma

Source:
https://frontlinegenomics.com/a-guide-to-the-fair-principles-in-biopharma/

Examples of FAIR in Biopharma

More at FAIR Toolkit: https://fairtoolkit.pistoiaalliance.org/category/use-cases/

- ISA - a metadata tracking framework
- Open PHACTS - Platform for drug discovery information
- wwPDB - Information about 3D structures of proteins & nucleic acids
- UniProt - Annotated data for protein sequences

Challenges in FAIR Biopharma

- **Unstructured Data** - data is generated in a variety of formats and that has to be organised, poor tagging leads to a lack of structure
- **Trapped Historical Data** - Unsupported old technologies, Inexplicability of existing data due to the absence of those who created it
- **Ontology Management** - Same items can be called by multiple different names
- **Cultural barriers** - A new approach towards data-management has to be taken, with the focus being on data-sharing
- **Investment** - Need to transition existing datasets and train people, lack of a predetermined path to standardise metadata and ontologies, people in the industry need to change their mindset from owning and using to reusing data

Challenges in scaling FAIR data

- Need for discipline-specific guidelines
- Lack of FAIR-guided organisation due to lack of data-related expertise -> training needed
- Assessing and quantifying the data to provide metrics is a rigid system
- FAIR data can be encouraged by allowed these metrics to different degrees

FAIRness criteria recommended by the FAIR Data Maturity Model Working Group - called the FAIR metrics can be used by organisations to self-assess their degree of FAIRness and adapt these metrics to the specific needs of the organisation.

Implementing FAIR in Biopharma

First Steps

- Strategies should focus on Data centric information systems to extract maximum value from their Data.
- Not all Data can be 100% FAIR and it is important to realise that FAIRification is an expensive, time-consuming process
- Technical Elements need to apply FAIR: Master Data Management, IT partners, Data Producers, Data Security
- How to make FAIR gains:
  - Provide incentives
  - Pre-competitive collaboration - companies work together to address the data-management problem
  - Clear Standards such as the FAIR metrics
  - Demonstrate value of investment

Collated Advice for starting FAIR journey in Biopharma:

- Use most common URIs
- Use controlled vocabularies
- Re-use existing ontologies
- Don’t make over-engineered ontologies
- Design with interoperability in mind
- Connect with existing FAIR databases

FAIR in AI

Source:
https://frontlinegenomics.com/fair-data-applied-to-ai-machine-learning/

Why are FAIR principles focussing on both humans and machines?

- Humans can easily interpret the context of a digital object
- Machines are needed to operate at the scale at which data is available nowadays - They are fast

How FAIR can help in AI

- A computational agent makes decisions about its actions based on its environment.
- AIs are such agents that can explore data autonomously irrespective of its type or access protocols
- Machine-actionable data is such that it provides enough information for such autonomous exploration by AIs to take appropriate actions like a human would.
- Poor data management and stewardship is a barrier to its use by AI. Thus the implementation of FAIR principles can help by making data machine-actionable and improve its quality.

Challenges & Solutions for FAIR in AI

- **Data Search:** AI advancements depend on good training datasets. More information about datasets is needed (structure of data, multimodality, sparseness and types of models trained using the datasets).

  Metadata, provenance and annotations are all essential for FAIR data in AI. Making sources and workflows visible can help associate learned information from a dataset.

- **Data Accessibility :** Algorithms need to be trained across dispersed repositories. These datasets have to used from a filesystem which limits the interpretation of data from different repositories by AI.

  Automated collection of metadata, provenance and annotations will help reduce researcher burden. The FAIRness of data collected will increase if these characteristics are machine-actionable.

- **Lack of Standardisation :** Metadata, ontologies, vocabularies play an important role in the way data is found and accessed and consequently used to train algorithms.
  A lack of standardisation here can cause hidden biases. There is no FAIR way yet to obtain standardised metadata.

  To solve this, the data producers need to be linked with data scientists and management experts to collaborate and recommend best practises, metadata standards and ontologies. Researcher engagement with AI experts is critical application of FAIR principles throughout projects.

Scalable Workflows

Snakemake and CWL

Shivam Kedia

Snakemake

https://academic.oup.com/bioinformatics/article/28/19/2520/290322

- Uses a python based engine to create something analogous to makefiles
- Helps in building workflows
- Uses rules defined to specify a workflow, rules consist of inputs, outputs and code to run (these can be shell scripts)
- Automatically infers names of input/output files which are specified as wildcards
- Uses a directed acyclic graph to establish the sequence of jobs and the dependencies between rules
- Independent tasks can be parallelised, and this helps the snakemake engine scale based on different machine types ranging from single-core machines to large compute clusters

Common Workflow Language

https://www.commonwl.org/index.html

- Group consisting of multiple vendors and individuals interested in portable workflows.
- Common Workflow language defines open standards to connect tools and create workflows
- Aim to create tools and workflows that can scale across systems (similar to snakemake in this aspect)