Bioinformatics Analysis Tools for NGS Data

Sequence representation and data retrieval

Philipp Rescheneder, Moritz Smolka

April 27, 2016

Center for Integrative Bioinformatics Vienna
Max F. Perutz Laboratories

CIBIV
Center for Integrative Bioinformatics Vienna

Sequence representation: FASTA

- One of the most ”dangerously” simple formats

Sequence representation: FASTA

- One of the most ”dangerously” simple formats
- Seemingly trivial but it is also ”under-specified”, there are many ”custom” extensions

Sequence representation: FASTA

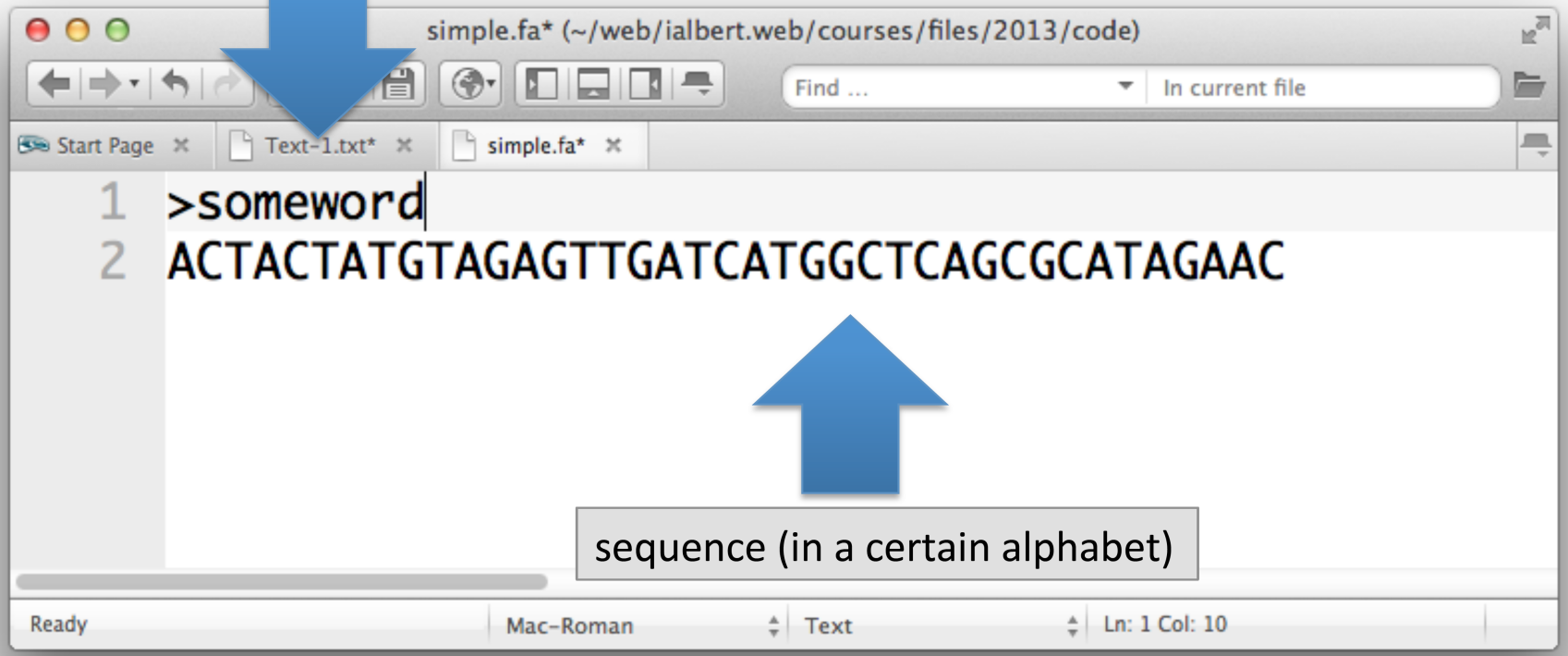
- One of the most ”dangerously” simple formats
- Seemingly trivial but it is also ”under-specified”, there are many ”custom” extensions
- Tools may make assumptions on the structure of a FASTA file

Sequence representation: FASTA

- One of the most ”dangerously” simple formats
- Seemingly trivial but it is also ”under-specified”, there are many ”custom” extensions
- Tools may make assumptions on the structure of a FASTA file
- Surprising number of problems can arise

FASTA format

identifier



sequence (in a certain alphabet)

The alphabet is similar to a specification: we need to know what are the valid characters to describe the sequence

Alphabets

- Nucleotide sequences: International Union of Pure and Applied Chemistry (IUPAC) codes

Table 1. The IUPAC nucleic acid notation

	Symbol	Meaning	Mnemonic
DNA Bases	G	Guanine	Guanine
	T	Thymine	Thymine
	A	Adenine	Adenine
	C	Cytosine	Cytosine
Ambiguity Characters	R	G + A	puRine
	Y	T + C	pYrimidine
	S	G + C	Strong interactions (3 H bonds)
	W	T + A	Weak interactions (2 H bonds)
	K	G + T	Keto
	M	A + C	aMino
	D	G + T + A	Not-C (D follows C in alphabet)
	H	T + A + C	Not-G (H follows G)
	B	G + T +C	Not-A (B follows A)
	V	G + A + C	Not-T or U (V follows U)
	N	G + A + T + C	aNy

Alphabets

- Nucleotide sequences: International Union of Pure and Applied Chemistry (IUPAC) codes
- Peptide sequence: amino acid one letter code

Table 1. The IUPAC nucleic acid notation

	Symbol	Meaning	Mnemonic
DNA Bases	G	Guanine	Guanine
	T	Thymine	Thymine
	A	Adenine	Adenine
	C	Cytosine	Cytosine
Ambiguity Characters	R	G + A	puRine
	Y	T + C	pYrimidine
	S	G + C	Strong interactions (3 H bonds)
	W	T + A	Weak interactions (2 H bonds)
	K	G + T	Keto
	M	A + C	aMino
	D	G + T + A	Not-C (D follows C in alphabet)
	H	T + A + C	Not-G (H follows G)
	B	G + T +C	Not-A (B follows A)
	V	G + A + C	Not-T or U (V follows U)
	N	G + A + T + C	aNy

SYMBOL		
1-Letter	**3-Letter**	**AMINO ACID**
Y	Tyr	Tyrosine
G	Gly	Glycine
F	Phe	Phenylalanine
M	Met	Methionine
A	Ala	Alanine
S	Ser	Serine
I	Ile	Isoleucine
L	Leu	Leucine
T	Thr	Threonine
V	Val	Valine
P	Pro	proline
K	Lys	Lysine
H	His	Histidine
Q	Gln	Glutamine
E	Glu	glutamic acid
Z	Glx	Glu and/or Gln
W	Trp	Tryptophan
R	Arg	Arginine
D	Asp	aspartic acid
N	Asn	asparagine
B	Asx	Asn and/or Asp
C	Cys	Cysteine
X	Xaa	Unknown or other

Multi record FASTA

It is not clear what the sequence above contains nucleic acids or aminoacids

(feels like a nucleic acids because of having so many ACTG both those are also valid amino acids)

More considerations

- Many tools will embed extra information into either the identifier or the ”free zone” of the description section
- See the FASTA format wiki page
- **Accession:** unique (often numerical) identifier for each sequence that is entered into the database
- **Locus** an identifier that represents a position in the genome, multiple accessions may point to the same locus
- Loci may have versions like: ABCD.1 ABCD.2

GenBank	gb|*accession*|*locus*
EMBL Data Library	emb|*accession*|*locus*
DDBJ, DNA Database of Japan	dbj|*accession*|*locus*
NBRF PIR	pir||*entry*
Protein Research Foundation	prf||*name*
SWISS-PROT	sp|*accession*|*entry name*
Brookhaven Protein Data Bank	pdb|*entry*|*chain*
Patents	pat|country|number
GenInfo Backbone Id	bbs|number
General database identifier	gnl|database|identifier
NCBI Reference Sequence	ref|accession|locus
Local Sequence identifier	lcl|identifier

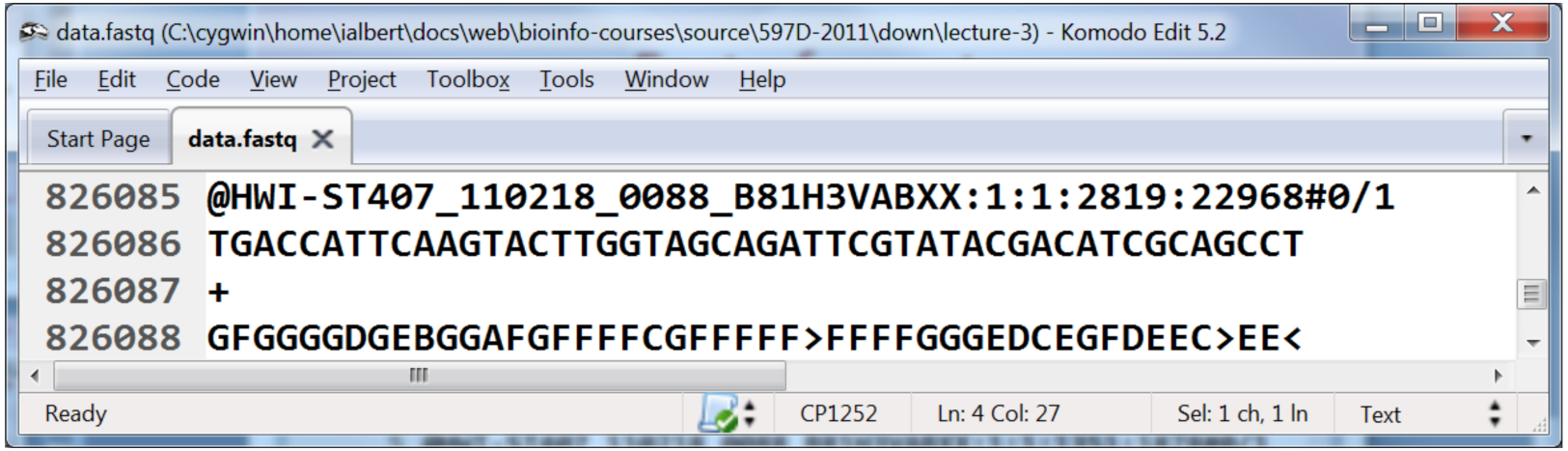
Understand your FASTA file

- First step of any sequence processing step
- How many sequences do we have
- Are sequences all on a single line or over multiple lines
- What is the identifier, what is embedded int the description
- We used almost exclusively for reference genomes

Extending the FASTA format

- The sequences are measurements
- There needs to be a way to associate quality measures to each base
- FASTQ: .fq, .fastq (FASTA with qualities)

Structure of a FASTQ file



Four lines per FASTQ record

1. @ indicates the sequence identifier
2. The sequence content of the read
3. + optionally repeat the sequence id (often left empty)
4. Sequence quality string

Paper: The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants - *Nucl. Acids Res. (2010) 38 (6): 1767-1771.*

Encodings

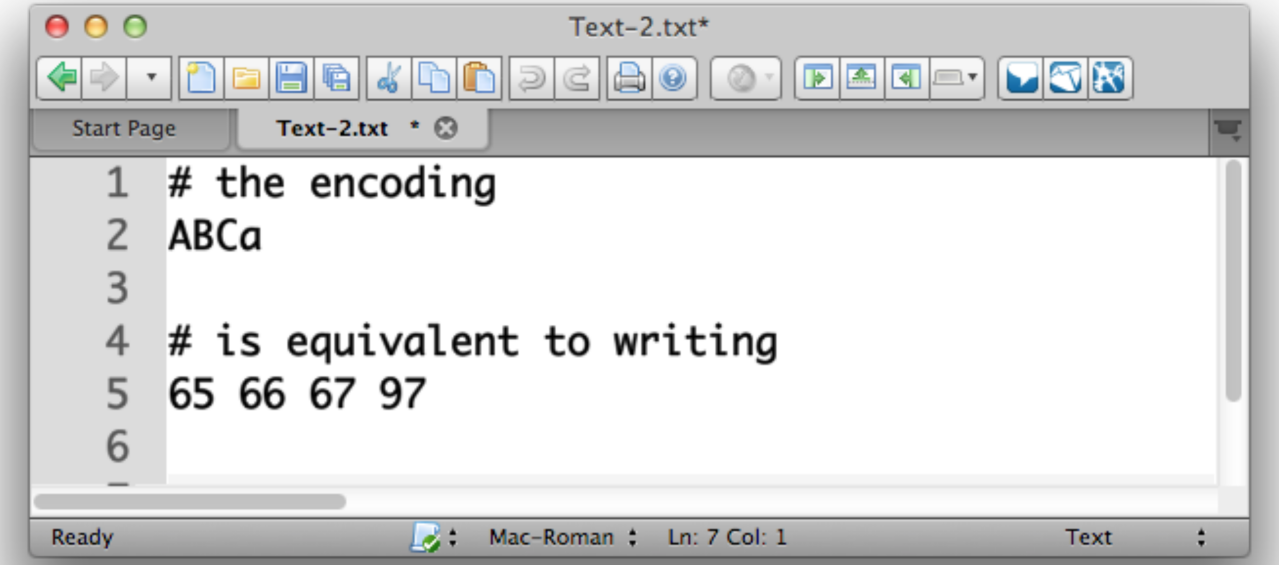
An encoding is a transformation from one representation to another

- The information is not changed
- Example: ASCII code

Encodings

An encoding is a transformation from one representation to another

- The information is not changed
- Example: ASCII code



One character → one byte space
ABCa = 4 bytes long
65 66 67 97 = 11 bytes long

Good: three characters are turned into one, saves space
Bad: not readable, hinders understanding

Quality Scores

- A quality score is a number that usually has limits, a low (say 0) to a high (say 40)
- A quality score represents an error probability
- It characterizes a single step of the process and NOT the entire experimental procedure
- Quality scores are used to represent base calling accuracy, alignment accuracy and other probabilities

PHRED Quality Scores

- The reported quality indicates the probability of an error

$$Q = -10 \log_{10}(e)$$

where e is the probability of a base call being wrong.

PHRED Quality Scores

- The reported quality indicates the probability of an error

$$Q = -10 \log_{10}(e)$$

where e is the probability of a base call being wrong.

  - Q10: 1 in 10 incorrectly called bases (90% accuracy)
  - Q20: 1 in 100 (99% accuracy)
  - Q30: 1 in 1000 (99.9 % accuracy)

There are multiple encodings

- Illumina used to switch around the encoding every once in a while
- Finally they settled on the Sanger encoding/Phred quality representation. Since 2011 or so.
- There are plenty of datasets/tools out there that may use different encodings!

Sanger Encoding

- Quality value range between 0 and 93
- Start the scale at character 33
- End the scale at character 33 + 93 = 126

Currently most instruments only produce qualities in the range of 0 to 40

Illumina 1.3 encoding

- Obsolete but still often observed in the wild
- Quality range between 0 to 62
- Start scale at character 64
- End scale at character 64 + 62 = 126

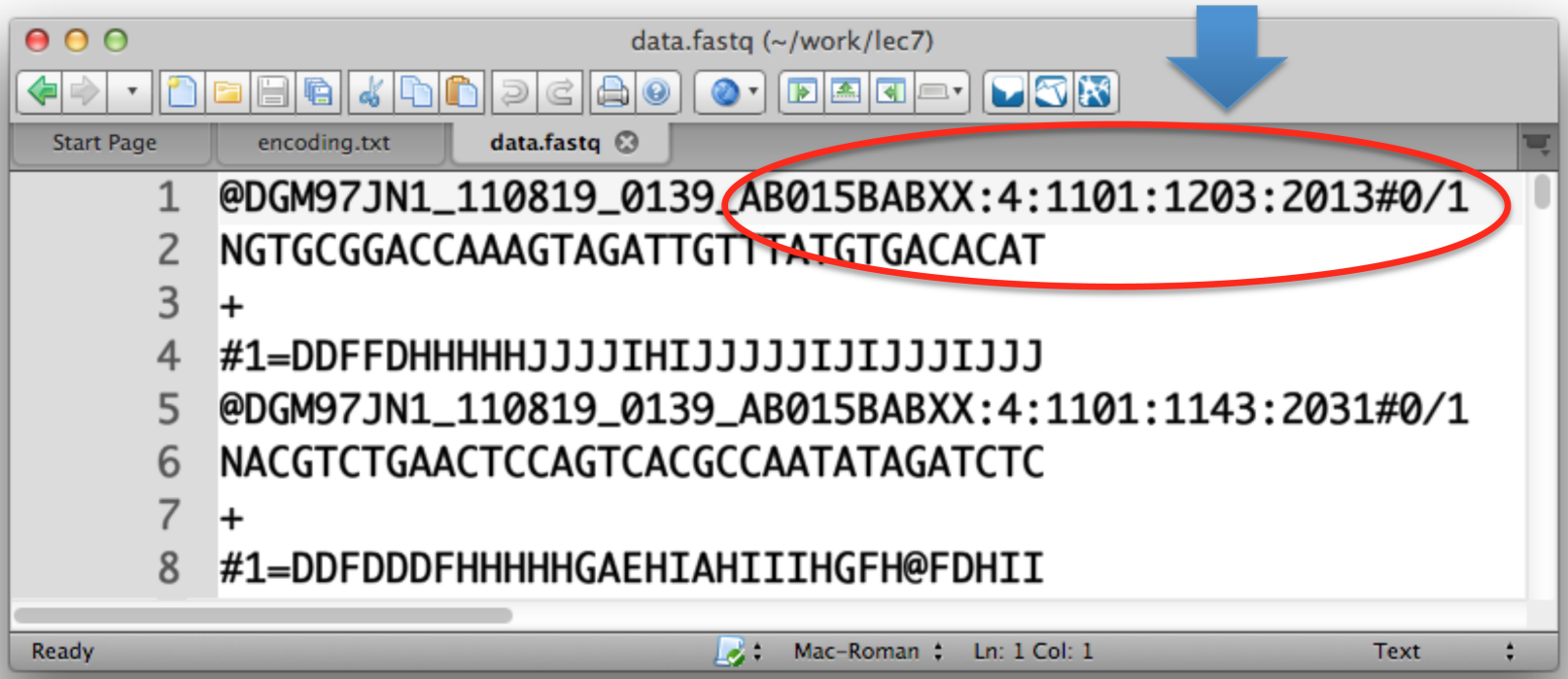
FASTQ encoding summary

```
 SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
 ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......................
 ................................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII......................
 ...................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ......................
 LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL.....................................................
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
 |                          |    |        |                              |                     |
33                         59   64       73                            104                   126
 0.........................26...31.......40
                           -5....0........9.............................40
                                 0........9.............................40
                                    3.....9.............................40
 0.2......................26...31.........41

S - Sanger        Phred+33,  raw reads typically (0, 40)
X - Solexa        Solexa+64, raw reads typically (-5, 40)
I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
J - Illumina 1.5+ Phred+64,  raw reads typically (3, 40)
    with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
    (Note: See discussion above).
L - Illumina 1.8+ Phred+33,  raw reads typically (0, 41)
```

FASTQ header

Illumina instrumentation
specific information: lane, tile, spot



De-facto standard for producing sequencing reads. The vast majority of current tools expect this format.

Storing data in SRA removes the extra header information in the FASTQ record! That is unfortunate! Some information is now lost and available only to the original authors!

FASTQ header

```
lec6 — ~/edu/lec6 — bash — 48×6

$ head -1 illumina-data.fq
@HWI-ST1342:96:H0NP9ADXX:2:1115:13393:59201

ialbert@grit ~/edu/lec6
$
```

1. Instrument name: **HWI-ST1342** (unique for every sequencer)
2. Run id: **96**
3. Flowcell id: **H0NP9ADXX** (unique for every flowcell)
4. Flowcell lane: **2**
5. Tile number within the flowcell: **1115**
6. X-coordinate of the cluster in the tile: **13393**
7. Y-coordinate of the cluster in the tile: **59201**

More fields are may also be present (not shown above):

1. Mate pair 1 or 2
2. Flag: Y or N

… control bits, index sequences, usually defined in the Illumina manuals

FASTQ paired end data

- There is no standard way to save paired end data in FASTQ files

FASTQ paired end data

- There is no standard way to save paired end data in FASTQ files
- Use two files
  - Both files must contain the same number of reads in the same order
  - reads_1.fq: Read1/1, Read2/1, Read3/1
  - reads_2.fq: Read1/2, Read2/2, Read3/2

FASTQ paired end data

- There is no standard way to save paired end data in FASTQ files
- Use two files
  - Both files must contain the same number of reads in the same order
  - reads_1.fq: Read1/1, Read2/1, Read3/1
  - reads_2.fq: Read1/2, Read2/2, Read3/2
- Use one file:
  - Mates must be next to each other
  - reads.fq: Read1/1, Read1/2, Read2/1, Read2/2, ...

FASTQ paired end data

- There is no standard way to save paired end data in FASTQ files
- Use two files
  - Both files must contain the same number of reads in the same order
  - reads_1.fq: Read1/1, Read2/1, Read3/1
  - reads_2.fq: Read1/2, Read2/2, Read3/2
- Use one file:
  - Mates must be next to each other
  - reads.fq: Read1/1, Read1/2, Read2/1, Read2/2, ...
- Most programs don't check the read names to find a matching pair

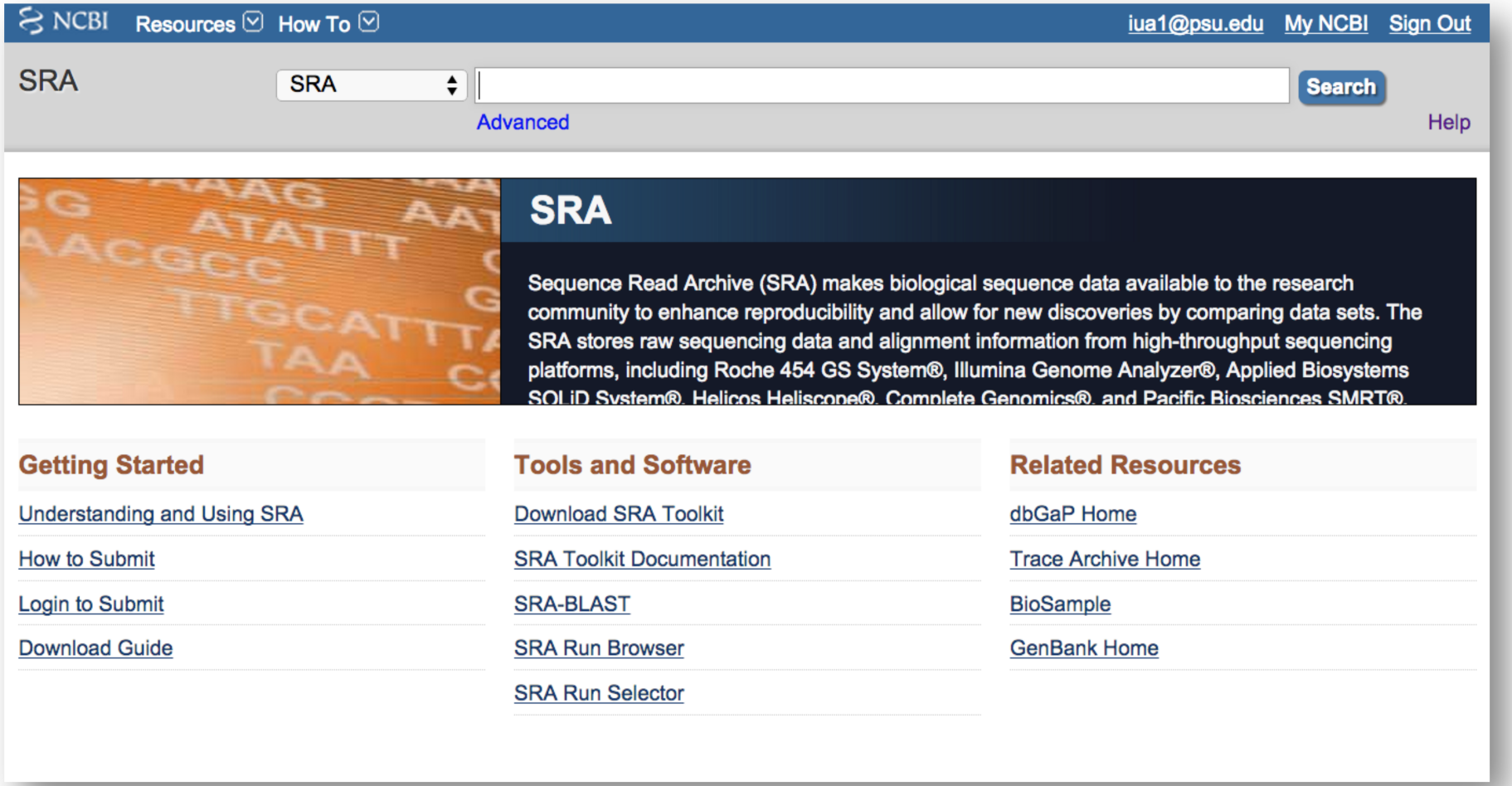
FASTQ paired end data

- There is no standard way to save paired end data in FASTQ files
- Use two files
  - Both files must contain the same number of reads in the same order
  - reads_1.fq: Read1/1, Read2/1, Read3/1
  - reads_2.fq: Read1/2, Read2/2, Read3/2
- Use one file:
  - Mates must be next to each other
  - reads.fq: Read1/1, Read1/2, Read2/1, Read2/2, ...
- Most programs don't check the read names to find a matching pair
- Simple to convert. You just have to know what the program you are using expects

FASTQ paired end data

- There is no standard way to save paired end data in FASTQ files
- Use two files
  - Both files must contain the same number of reads in the same order
  - reads_1.fq: Read1/1, Read2/1, Read3/1
  - reads_2.fq: Read1/2, Read2/2, Read3/2
- Use one file:
  - Mates must be next to each other
  - reads.fq: Read1/1, Read1/2, Read2/1, Read2/2, ...
- Most programs don't check the read names to find a matching pair
- Simple to convert. You just have to know what the program you are using expects
- When working with paired FASTQ files, do simple sanity checks (e.g. count the number of reads in both files)

Data retrieval: Short read archive



It is (partially) documented and “sort of logical” – but only “sort of”

SRA naming conventions

NCBI BioProject: **PRJN…** (aka SRA study **SRP…**)

- the overall description of a single research initiative; a project will typically relate to multiple samples and datasets

NCBI BioSample: **SAMN…** (aka SRA Sample **SRS…**)

- a description of biological source material; each physically unique specimen should be registered as a single BioSample with a unique set of attributes

SRA Experiment: **SRX…**

- a unique sequencing library for a specific sample

SRA Run: **SRR…**

This contains the data

- a manifest of data file(s) linked to a given sequencing library (experiment)

Full list of prefixes

Accession Prefix	Accession Name	Definition
SRA	SRA submission accession	The submission accession represents a virtual container that holds the objects represented by the other five accessions and is used to track the submission in the archive.
SRP	SRA study accession	A Study is an object that contains the project metadata describing a sequencing study or project. Imported from BioProject.
SRX	SRA experiment accession	An Experiment is an object that contains the metadata describing the library, platform selection, and processing parameters involved in a particular sequencing experiment.
SRR	SRA run accession	A Run is an object that contains actual sequencing data for a particular sequencing experiment. Experiments may contain many Runs depending on the number of sequencing instrument runs that were needed.
SRS	SRA sample accession	A Sample is an object that contains the metadata describing the physical sample upon which a sequencing experiment was performed. Imported from BioSample.
SRZ	SRA analysis accession	An analysis is an object that contains a sequence data analysis BAM file and the metadata describing the sequence analysis.

Visit BioProject for the data

BioProject | BioProject | Limits | Advanced

Display Settings: | Send to:

Zaire ebolavirus

Accession: PRJNA257197 ID: 257197

Zaire ebolavirus Genome sequencing

Zaire ebolavirus sample sequencing from the 2014 outbreak in Sierra Leone, West Africa.

Project Data Type: Genome sequencing

Attributes: Scope: Multiisolate; Material: Genome; Capture: Whole; Method type: Sequencing

Relevance: Medical

Project Data:

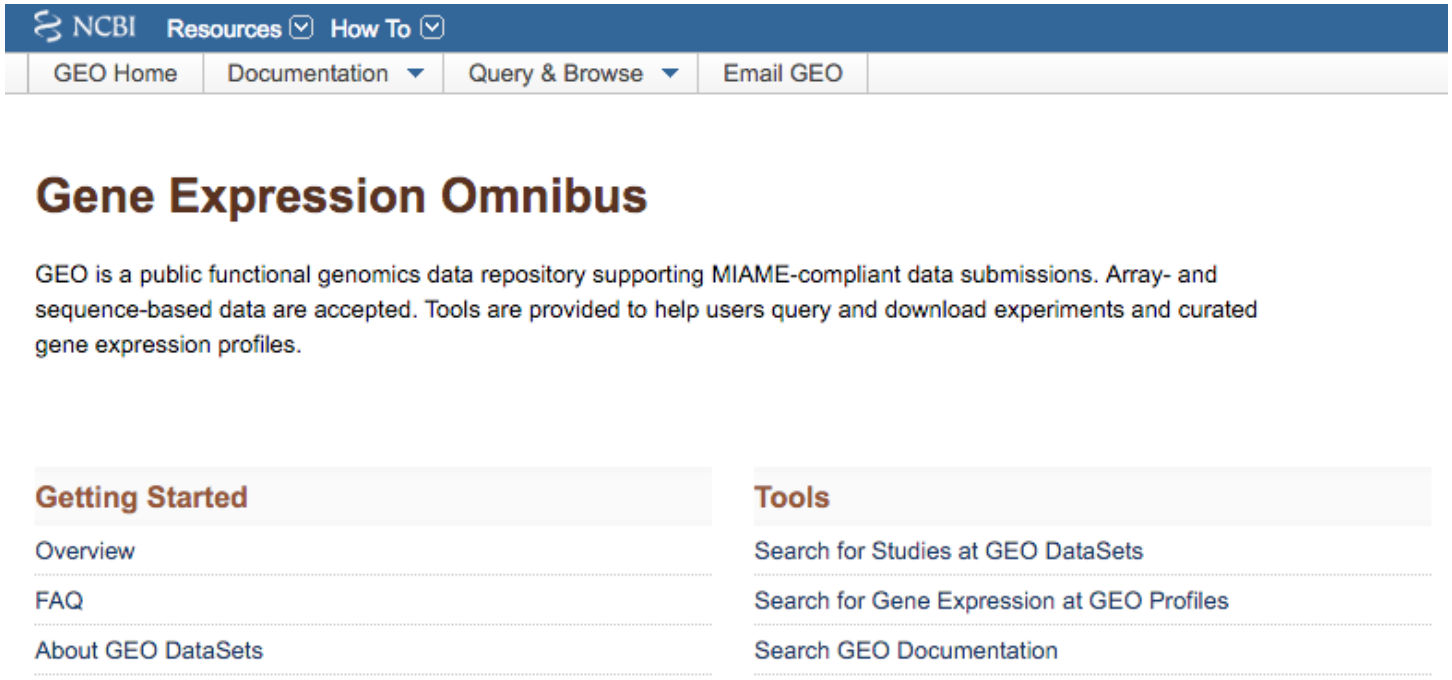
Resource Name	Number of Links
SEQUENCE DATA	
Nucleotide (Genomic RNA)	99
SRA Experiments	195
Protein Sequences	891
OTHER DATASETS	
BioSample	99

Gene expression Omnibus

- GEO was originally designed for microarray data, later augmented for high throughput sequencing
- The Gene Expression Omnibus also stores results from functional genomic experiments.
- Additional data is stored at GEO (e.g. read counts from RNA-Seq)
- But the raw data links back to SRA

Gene expression Omnibus

- GEO was originally designed for microarray data, later augmented for high throughput sequencing
- The Gene Expression Omnibus also stores results from functional genomic experiments.
- Additional data is stored at GEO (e.g. read counts from RNA-Seq)
- But the raw data links back to SRA



GEO example experiment

NCBI

GEO Gene Expression Omnibus

HOME | SEARCH | SITE MAP | GEO Publications | FAQ | MIAME | Email GEO

NCBI > GEO > **Accession Display** Not logged in | Login

Scope: Self Format: HTML Amount: Quick GEO accession: GSE70149 GO

Series GSE70149 Query DataSets for GSE70149

Status	Public on Jan 19, 2016
Title	A-type lamins bind both hetero- and euchromatin, the latter being regulated by lamina-associated polypeptide 2alpha
Organism	Mus musculus
Experiment type	Expression profiling by high throughput sequencing Genome binding/occupancy profiling by high throughput sequencing
Summary	This SuperSeries is composed of the SubSeries listed below.
Overall design	Refer to individual Series
Citation missing	*Has this study been published? Please login to update or notify GEO.*
Submission date	Jun 22, 2015
Last update date	Apr 21, 2016
Contact name	Philipp Rescheneder
E-mail	philipp.rescheneder@univie.ac.at
Organization name	MFPL
Street address	Dr. Bohr Gasse 9
City	Vienna
ZIP/Postal code	1030
Country	Austria

Platforms (1) GPL13112 Illumina HiSeq 2000 (Mus musculus)

Samples (28) GSM1717489 Input WT 12cyc DNA
More... GSM1717490 Input KO 12cyc DNA
GSM1717491 LAP2alpha WT 12cyc ChIPSeq

This SuperSeries is composed of the following SubSeries:

GSE70147 A-type lamins bind both hetero- and euchromatin, the latter being regulated by lamina-associated polypeptide 2alpha [ChIP-Seq]

GSE70148 A-type lamins bind both hetero- and euchromatin, the latter being regulated by lamina-associated polypeptide 2alpha [gene expression]

Relations

BioProject PRJNA287718

GEO sample

Library strategy	ChIP-Seq
Library source	genomic
Library selection	ChIP
Instrument model	Illumina HiSeq 2000

Data processing	Adapters were clipped with cutadapt Reads were mapped to the mouse genome with bowtie2 version 2.1.0 using default parameters. Reads with a mapping quality below 20 were discarded. Reads stemming from PCR duplicates were removed and all ChIP read files were downsampled to match the read count of their respective input sample using picard-tools Regions of enrichment (peaks) were identified EDD (parameters "-b 11 –g 5 –fdr 0.1) version 1.0.2 and SICER version 1.1 (parameters: window size of 1,000bp, a gap size of 3,000bp and a false discovery rate of 0.01) Genome_build: mm9 Supplementary_files_format_and_content: BED files containing peaks called by EDD or SICER; BigWig files containing log ratios between histone modification ChIP-Seq read counts and respective Input read counts for 500bp bins

Submission date	Jun 22, 2015
Last update date	Jan 20, 2016
Contact name	Philipp Rescheneder
E-mail	philipp.rescheneder@univie.ac.at
Organization name	MFPL
Street address	Dr. Bohr Gasse 9
City	Vienna
ZIP/Postal code	1030
Country	Austria

Platform ID	GPL13112	
Series (2)	GSE70147	A-type lamins bind both hetero- and euchromatin, the latter being regulated by lamina-associated polypeptide 2alpha [ChIP-Seq]
	GSE70149	A-type lamins bind both hetero- and euchromatin, the latter being regulated by lamina-associated polypeptide 2alpha

Relations

BioSample	SAMN03785446

Supplementary file	Size	Download	File type/resource
GSM1717491_17468_17468_CCGTCC_H7E4VADXX_2_20140306B_20140306_bowtie2_filtered_dupremoved_downsampled-W1000-G3000-FDR0.01-island.bed.gz	77.5 Kb	(ftp)(http)	BED
GSM1717491_17468_edd-b11-g5-fdr0.1_peaks.bed.gz	3.7 Kb	(ftp)(http)	BED

Raw data provided as supplementary file

Processed data provided as supplementary file

Getting data from SRA

- In sra format
- Special format to increase compression rate
- You will need to install a software called **sra-toolkit**
  - `github.com/ncbi/sra-tools/wiki/Downloads`

- Download manually and unzip with fastq-dump

```
$ fastq-dump SRR501544.sra
```

- Get data directly with fastq-dump

```
$ fastq-dump SRR501544
```

- When working with paired end data using the "–split-3" option is important