Sequenciação de Nova Geração

- NGS: formatos de ficheiro
- NGS: Variant Calling Analysis

1

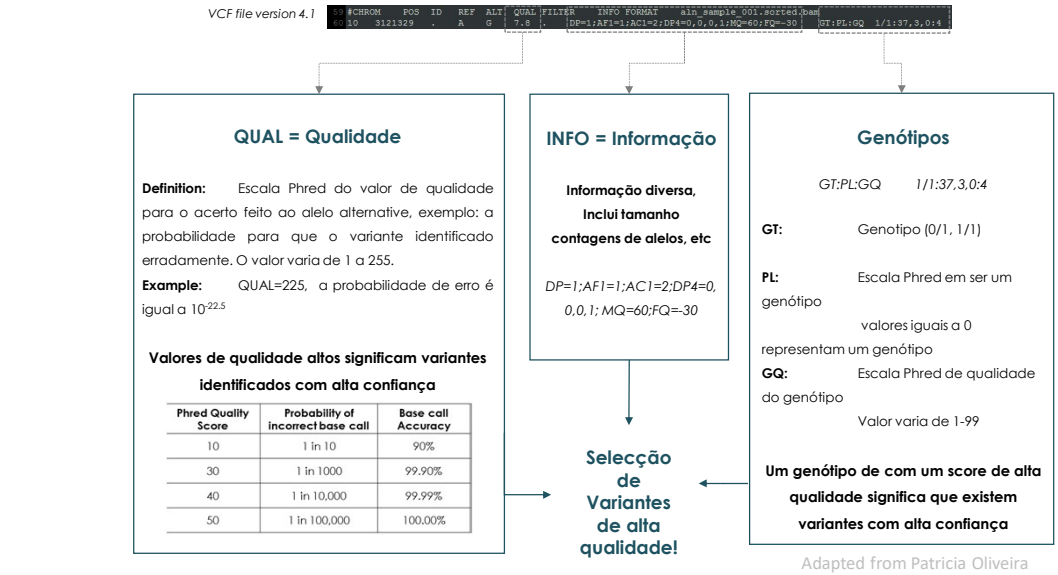
NGS: Formato VCF

- Variant Call Format (VCF) é um formato no qual é possível guardar SNP, inserções, deleções, variantes de numero de cópias e variantes estruturais.
- Este formato foi definido pelo consórcio 1000 Genomes: http://www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-41 e o grupo de investigação GA4GH: http://ga4gh.org/#/fileformats-team
- Cada entrada tem 8 colunas delimitadas por tabs:
  - 1. CHROM - cromossoma
  - 2. POS – posição de referência
  - 3. ID – Identificadores. Na existência de mais que um identificador é utilizado o símbolo ";" para os separar.
  - 4. REF – Nucleótido(s) de referência: Cada base deve conter um A,C,G,T,N (case insensitive) ou mais.
  - 5. ALT – Nucleótido(s) alternativos: Lista separada por virgulas dos nucleótidos identificados (A,C,G,T,N,*) em pelo menos uma amostra.
  - 6. QUAL - Qualidade: Valor de escala Phred de qualidade de acerto do ALT. i.e. ?10log10 prob(variante em ALT ser errado).
  - 7. FILTER – Estado do filtro: PASS se passou a todos os filtros do sequenciador, i.e. o variante existe nesta posição.
  - 8. INFO – Informação adicional: Os campos INFO contêm informação guardada numa lista separada por o símbolo ";"
  - Chaves com valores opcionais no formato: <key>=<data>[,data].

Dataset para a aula:
https://www.dropbox.com/s/2qfdhk69a2pajc5/Aula_VCFs.zip

2

NGS: Filtragem de Qualidade de variantes detetados



Adapted from Patricia Oliveira

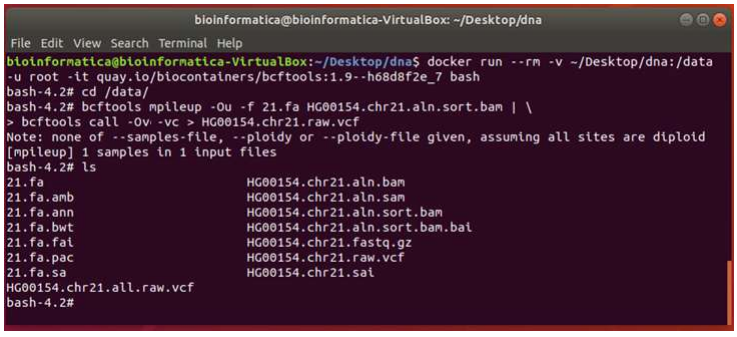
3

NGS: Identificar os variantes

```
# Verificar os alinhamentos
$ samtools tview HG00154.chr21.aln.sort.bam 21.fa -p 21:9417961
```

```
# Identificar os variantes e extrair para um ficheiro vcf

$ bcftools mpileup -Ou -f 21.fa HG00154.chr21.aln.sort.bam | \
bcftools call -Ov -vc > HG00154.chr21.raw.vcf
```



Execute:
docker pull quay.io/biocontainers/bcftools:1.9--h68d8f2e_7

4

NGS: Manipular ficheiros VCF

- Leitor de VCFs para Python: **PyVCF** (https://pyvcf.readthedocs.io/en/latest)

```
# Leitura do ficheiro vcf
>>>import vcf
>>>vcf_reader = vcf.Reader(open('~/HG00154.chr21.raw.vcf', 'r'))
>>>for record in vcf_reader:
...   print(record)
```

```
File  Edit  View  Search  Terminal  Help
Record(CHROM=21, POS=10753628, REF=G, ALT=[A])
Record(CHROM=21, POS=10753688, REF=T, ALT=[C])
Record(CHROM=21, POS=10753714, REF=T, ALT=[C])
Record(CHROM=21, POS=10753745, REF=C, ALT=[T])
Record(CHROM=21, POS=10753752, REF=T, ALT=[C])
Record(CHROM=21, POS=10753777, REF=G, ALT=[T])
Record(CHROM=21, POS=10753787, REF=G, ALT=[C])
Record(CHROM=21, POS=10753830, REF=T, ALT=[A])
Record(CHROM=21, POS=10753857, REF=G, ALT=[C])
Record(CHROM=21, POS=10753858, REF=A, ALT=[C])
Record(CHROM=21, POS=10753896, REF=G, ALT=[A])
Record(CHROM=21, POS=10753938, REF=C, ALT=[T])
Record(CHROM=21, POS=10753948, REF=T, ALT=[G])
Record(CHROM=21, POS=10753960, REF=C, ALT=[T])
Record(CHROM=21, POS=10753969, REF=A, ALT=[T])
```

Instação do package requer a instalação do "Build Tools for Microsoft Studio" disponivel em: https://visualstudio.microsoft.com/downloads/

Instalação via pip no anaconda prompt:

```
$ pip install PyVCF
```

VCF file:
https://www.dropbox.com/s/uu6av8ciffamtwd/HG00154.chr21.raw.vcf

NGS: Manipular ficheiros VCF

- O object Record contém diversas propriedades importantes para a interpretação dos variantes:

https://pyvcf.readthedocs.io/en/latest/API.html#vcf-model-record

```
>>> record = next(vcf_reader)
>>> print(record)
Record(CHROM=21, POS=21725686, REF=T, ALT=[C])
>>> print(record.INFO)
{'DP': 1, 'SGB': -0.379885, 'MQ0F': 0.0, 'AF1': 1.0, 'AC1': 2.0, 'DP4': [0, 0,
0, 1], 'MQ': 37, 'FQ': -29.9935}
>>> print(record.ALT)
[C]
>>> print(record.POS)
21725686
>>> print(record.CHROM)
21
>>> print(record.QUAL)
3.54557
>>> print(record.FILTER)
None
>>> print(record.FORMAT)
GT:PL
>>> print(record.samples)
[Call(sample=HG00154.chr21.aln.sort.bam, CallData(GT=0/1, PL=[31, 3, 0]))]
>>> print(record.genotype)
<bound method _Record.genotype of <vcf.model._Record object at 0x7fa8632acc50>>
>>> print(record.REF)
T
```

NGS: Formato GFF

- General Feature Format (GFF) é um formato usado para descrever anotações de estruturas de genes. As linhas de GFF requerem 9 campos separados por tabs.

Format:
1. seqname – Nome da sequência de cromossoma ou scaffold.
2. source – O programa utilizado para gerar o atributo.
3. feature – O nome do tipo do atributo. Os tipos comuns são "CDS", "start_codon", "stop_codon", e "exon".
4. start – A posição no cromossoma do inicio do atributo. A contagem começa em 1.
5. end – A posição no cromossoma do fim do atributo (inclusive).
6. score – Um valor entre 0 e 1000 utilizado para colorir o gene.
7. strand – Identifica a direcionalidade da sequencia pelos valores '+', '-', ou '.'
8. frame – Se o atributo é um exão codificante, a "frame" deve ter um valor entre 0 e 2 que representa a janela de leitura da primeira base. Se o atributo não for um exão codificante, o valor deve ser '.'.
9. group – Todas as linhas com o mesmo grupo estão ligadas no mesmo objeto/identificador.

NGS: Formato GTF

- GTF (Gene Transfer Format), um formato que evoluiu do GFF, contem as primeiras 8 colunas do GFF, mas a coluna group foi alterada para apresentar uma lista de atributos.

```
$ cd GenomeAnnotation
$ gzcat genes.gtf.zip | head

1  unknown     exon    11874   12227   .       +       .       gene_id
   "LOC100287102"; gene_name "LOC100287102"; p_id "P25115";
   transcript_id "XM_002342010.1"; tss_id "TSS26210";

1       unknown CDS     12190   12227   .       +       0       gene_id
"LOC100287102"; gene_name "LOC100287102"; p_id "P25115";
transcript_id "XM_002342010.1"; tss_id "TSS26210";
```

NGS: Manipular ficheiros GFF/GTF

- Leitor de GFF/GTF para Python: **BCBIO-GFF** (https://biopython.org/wiki/GFF_Parsing)
- É possível transformar as coordenadas dos genes presentes no ficheiro GFF/GTF em objetos **SeqRecord** do Biopython.
- O package também disponibiliza funcionalidades de parsing parcial dos ficheiros GFF/GTF de forma a obter somente os objetos **SeqRecord** que contenham a informação relevante para o estudo.

```
from BCBio import GFF

in_file = "your_file.gff"

in_handle = open(in_file)
for rec in GFF.parse(in_handle):
    print(rec)
in_handle.close()
```

Instalação via pip no anaconda prompt:

```
$ pip install bcbio-gff
```