FACULTAD DE MEDICINA
UNIVERSIDAD DE CHILE

GENOMED-Lab
http://genomed.med.uchile.cl

ICBM INSTITUTO DE CIENCIAS BIOMÉDICAS

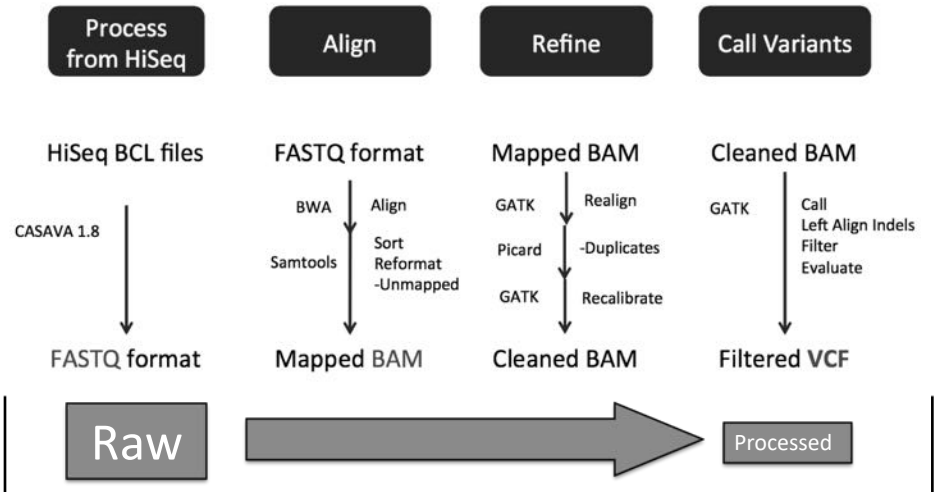
Análisis de datos NGS:
Llamado de variantes

RICARDO A. VERDUGO, Ph.D.

Programa de Genética Humana, ICBM
Facultad de Medicina, U. de Chile

Abril 2019

NGS data files



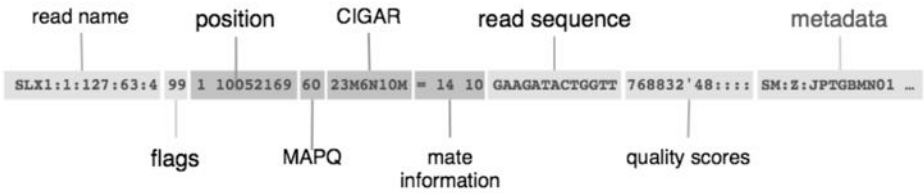
SAM file

```
@HD     VN:1.0  SO:coordinate
@SQ     SN:chr20        LN:64444167
@PG     ID:TopHat       VN:2.0.14       CL:/srv/dna_tools/tophat/tophat -N 3 --read-edit-dist 5 --read-rea
lign-edit-dist 2 -i 50 -I 5000 --max-coverage-intron 5000 -M -o out /data/user446/mapping_tophat/index/chr
20 /data/user446/mapping_tophat/L6_18_GTGAAA_L007_R1_001.fastq
HWI-ST1145:74:C101DACXX:7:1102:4284:73714       16      chr20   190930  3       100M    *       0       0
        CCGTGTTTAAAGGTGGATGCGGTCACCTTCCCAGCTAGGCTTAGGGATTCTTAGTTGGCCTAGGAAATCCAGCTAGTCCTGTCTCTCAGTCCCCCCTCT
C    BBDCCDDCCDDDDCDDDDDDCDCCCDBC?DDDDDDDDDDDDDDDCCDCDDDDDDDDDDCCCCEDDDC?DDDDDDDDDDDDDDDDDDDDDDBDHFFFFDC@@
    AS:i:-15        XM:i:3  XO:i:0  XG:i:0  MD:Z:55C20C13A9 NM:i:3  NH:i:2  CC:Z:=  CP:i:55352714   HI:i:0
HWI-ST1145:74:C101DACXX:7:1114:2759:41961       16      chr20   193953  50      100M    *       0       0
        TGCTGGATCATCTGGTTAGTGGCTTCTGACTCAGAGGACCTTCGTCCCCTGGGGCAGTGGACCTTCCAGTGATTCCCCTGACATAAGGGGCATGGACGA
G    DCDDDDEDDDDDDDCDDDDDDDCCCDDDCDDDDDEEC>DFFFEJJJJJIGJJJJIHGBHHGJIJJJJJJGJJJIJJJJJIHJJJJJJHHHHHFFFFCCC
    AS:i:-16        XM:i:3  XO:i:0  XG:i:0  MD:Z:60G16T18T3 NM:i:3  NH:i:1
HWI-ST1145:74:C101DACXX:7:1204:14760:4030       16      chr20   270877  50      100M    *       0       0
        GGCTTTATTGGTAAAAAAGGAATAGCAGATTTAATCAGAAATTCCCACCTGGCCCAGCAGCACCAACCAGAAAGAAGGGAAGAAGACAGGAAAAAACCA
C    DDDDDDDDDCCDDDDDDDDDDEEEEEEEFFFEFFEGHHHHFGDJJIHJJIJIJJJIIIIGGFJJIHIIIIJJJJJJIGHHFAHGFHJHFGGHFFFDD@BB
    AS:i:-11        XM:i:2  XO:i:0  XG:i:0  MD:Z:0A85G13    NM:i:2  NH:i:1
HWI-ST1145:74:C101DACXX:7:1210:11167:8699       0       chr20   271218  50      50M4700N50M     *       0
        0       GTGGCTCTTCCACAGGAATGTTGAGGATGACATCCATGTCTGGGGTGCACTTGGGTCTCCGAAGCAGAACATCCTCAAATATGACCTCTCG
accepted_hits.sam
```

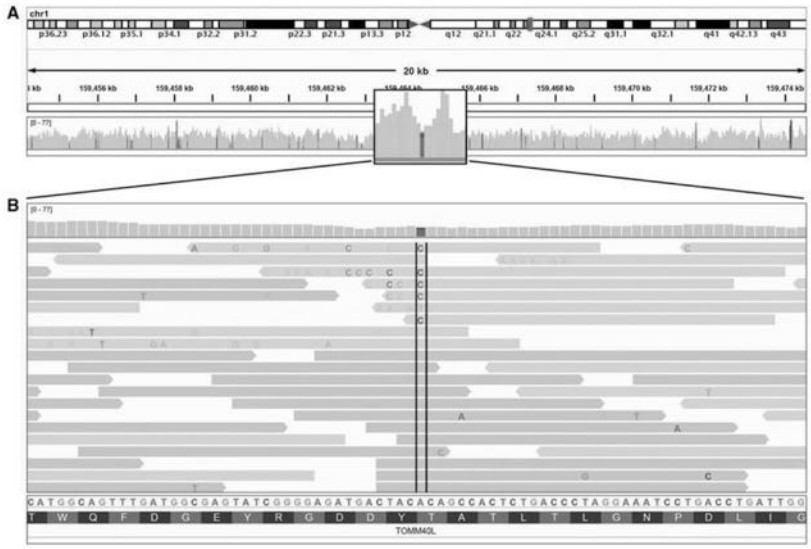
Tag	Description
@HD	The header line. The first line if present.
VN*	Format version. *Accepted format:* /^[0-9]+\.[0-9]+$/.
SO	Sorting order of alignments. *Valid values:* `unknown` (default), `unsorted`, `queryname` and `coordinate`. For coordinate sort, the major sort key is the RNAME field, with order defined by the order of @SQ lines in the header. The minor sort key is the POS field. For alignments with equal RNAME and POS, order is arbitrary. All alignments with '*' in RNAME field follow alignments with some other value but otherwise are in arbitrary order.
GO	Grouping of alignments, indicating that similar alignment records are grouped together but the file is not necessarily sorted overall. *Valid values:* `none` (default), `query` (alignments are grouped by QNAME), and `reference` (alignments are grouped by RNAME/POS).
@SQ	Reference sequence dictionary. The order of @SQ lines defines the alignment sorting order.
SN*	Reference sequence name. Each @SQ line must have a unique SN tag. The value of this field is used in the alignment records in RNAME and RNEXT fields. Regular expression: [!-)+-<>-~][!-~]*
LN*	Reference sequence length. *Range:* [1,2^{31}-1]
AS	Genome assembly identifier.
M5	MD5 checksum of the sequence in the uppercase, excluding spaces but including pads (as '*'s).
SP	Species.
UR	URI of the sequence. This value may start with one of the standard protocols, e.g http: or ftp:. If it does not start with one of these protocols, it is assumed to be a file-system path.
@RG	Read group. Unordered multiple @RG lines are allowed.
ID*	Read group identifier. Each @RG line must have a unique ID. The value of ID is used in the RG tags of alignment records. Must be unique among all read groups in header section. Read group IDs may be modified when merging SAM files in order to handle collisions.
CN	Name of sequencing center producing the read.
DS	Description.
DT	Date the run was produced (ISO8601 date or date/time).
FO	Flow order. The array of nucleotide bases that correspond to the nucleotides used for each flow of each read. Multi-base flows are encoded in IUPAC format, and non-nucleotide flows by various other characters. *Format:* /*|[ACMGRSVTWYHKDBN]+/
KS	The array of nucleotide bases that correspond to the key sequence of each read.
LB	Library.
PG	Programs used for processing the read group.
PI	Predicted median insert size.
PL	Platform/technology used to produce the reads. *Valid values:* `CAPILLARY`, `LS454`, `ILLUMINA`, `SOLID`, `HELICOS`, `IONTORRENT`, `ONT`, and `PACBIO`.
PM	Platform model. Free-form text providing further details of the platform/technology used.
PU	Platform unit (e.g. flowcell-barcode.lane for Illumina or slide for SOLID). Unique identifier.
SM	Sample. Use pool name where a pool is being sequenced.
@PG	Program.
ID*	Program record identifier. Each @PG line must have a unique ID. The value of ID is used in the alignment PG tag and PP tags of other @PG lines. PG IDs may be modified when merging SAM files in order to handle collisions.
PN	Program name
CL	Command line

Archivo SAM: registros

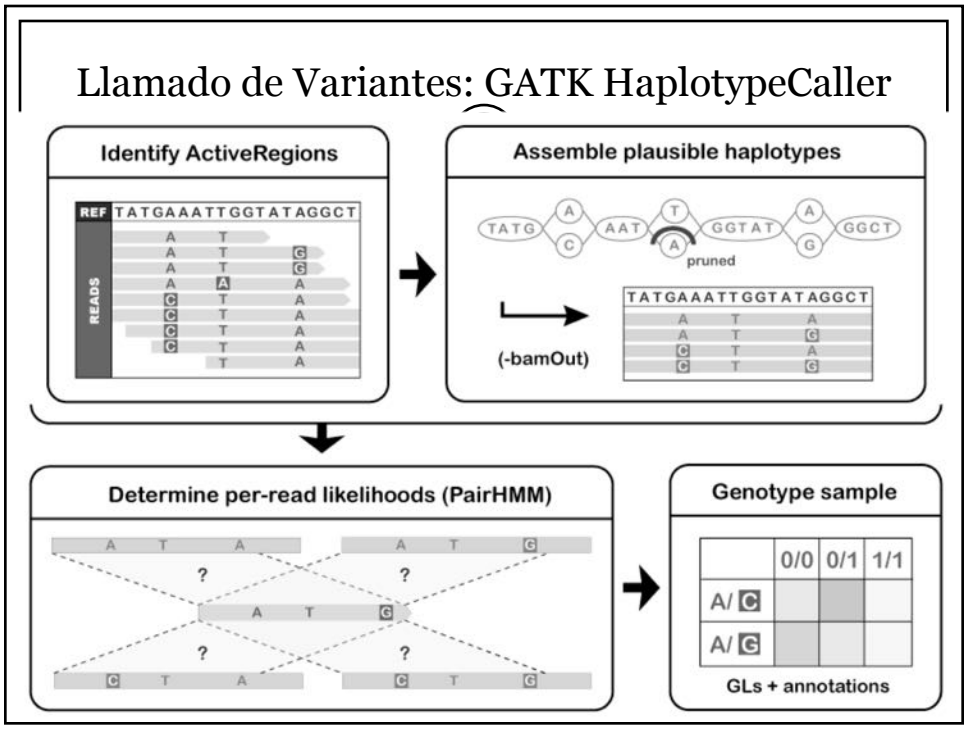
HEADER containing metadata (sequence dictionary, read group definitions etc)
RECORDS containing structured read information (1 line per read record)

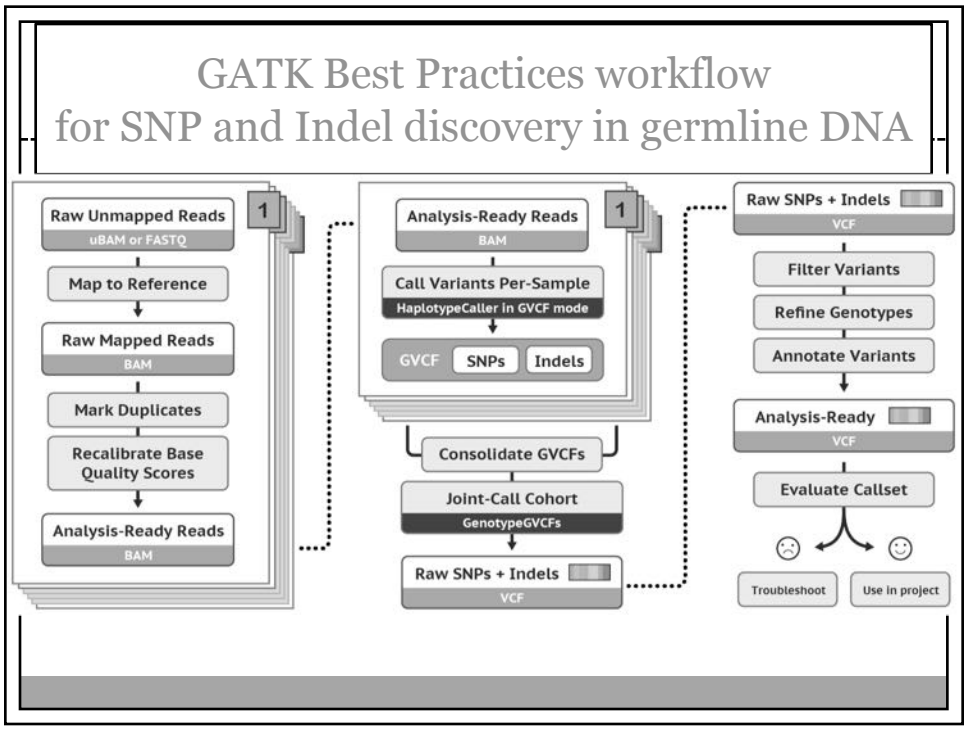


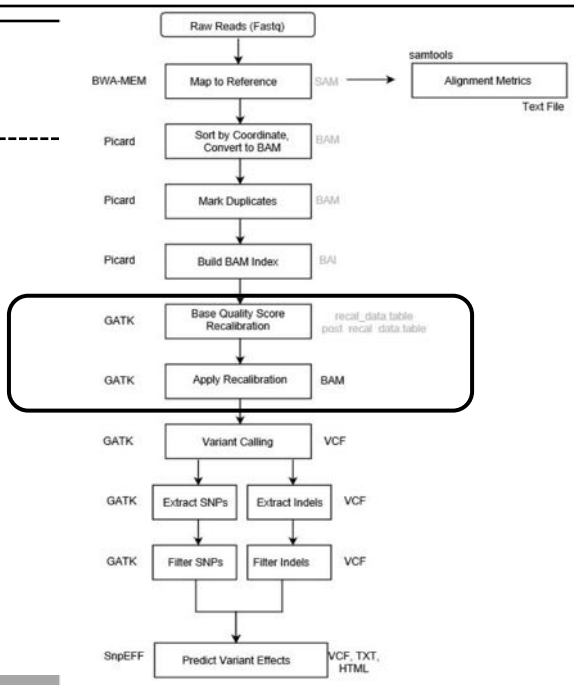
Integrative Genomic Viewer



http://software.broadinstitute.org/software/igv/


Llamado de Variantes: GATK HaplotypeCaller
Identify ActiveRegions
REF
TATGAAATTGGTATAGGCT
READS
Assemble plausible haplotypes
TATG
AAT
GGTAT
GGCT
pruned
(-bamOut)
Determine per-read likelihoods (PairHMM)
Genotype sample
0/0
0/1
1/1
A/C
A/G
GLs + annotations


GATK Best Practices workflow
for SNP and Indel discovery in germline DNA
Raw Unmapped Reads
uBAM or FASTQ
Map to Reference
Raw Mapped Reads
BAM
Mark Duplicates
Recalibrate Base Quality Scores
Analysis-Ready Reads
BAM
Analysis-Ready Reads
BAM
Call Variants Per-Sample
HaplotypeCaller in GVCF mode
GVCF
SNPs
Indels
Consolidate GVCFs
Joint-Call Cohort
GenotypeGVCFs
Raw SNPs + Indels
VCF
Raw SNPs + Indels
VCF
Filter Variants
Refine Genotypes
Annotate Variants
Analysis-Ready
VCF
Evaluate Callset
Troubleshoot
Use in project



Individual quality scores (blue bars) for each position in a single read. The horizontal blue line represents the Q20 phred score value

Why do we care about quality scores so much?

- Variant calling algorithms rely on the quality score assigned to the individual base calls
- Tells us how much we can trust that particular observation to inform us about the biological truth of the site
- If we have a basecall that has a low quality score, that means we're not sure we actually read that A correctly, and it could actually be something else
- So we won't trust it as much as other base calls that have higher qualities
- We use that score to weigh the evidence that we have for or against a variant existing at a particular site

Why Recalibrate?

- Scores produced by the machines are subject to various sources of systematic technical error
- Leads to over- or under-estimated base quality scores in the data.
- Errors can arise due to the physics or the chemistry of how the sequencing reaction works, possibly manufacturing flaws in the equipment.

Why Recalibrate?

Base quality score recalibration (BQSR) is a process in which we apply machine learning to model these errors empirically and adjust the quality scores accordingly.

Raw, high-sensitivity callsets contain many false positives

- Mutation calling algorithms are very permissive by design
- How to filter?
  - Hand-tuned hard-filtering requires time and expertise
  - Better to learn what the filters should be from the data itself
- Must enable analysts to trade off sensitivity and specificity depending on project goals

☑ **Building a model of what true genetic variation looks like will allow us to rank-order variants based on their likelihood of being real**

How does BQSR work?

1. You provide GATK Base Recalibrator with a set of known variants.
2. GATK Base Recalibrator analyzes all reads looking for mismatches between the read and reference, skipping those positions which are included in the set of known variants (from step 1).
3. GATK Base Recalibrator computes statistics on the mismatches (identified in step 2) based on the reported quality score, the position in the read, the sequencing context (ex: preceding and current nucleotide).
4. Based on the statistics computed in step 3, an empirical quality score is assigned to each mismatch, overwriting the original reported quality score.

From annotations to mixture models

- Each variant has a diverse set of statistics associated with it.
- These annotations tend to form Gaussian clusters
- We can fit a "Gaussian mixture model" to the annotations known variants in our dataset.
- Any new variant can be scored by evaluating the associated annotations in this model.

Variant annotations are the "features" of the model

VCF record for an A/G SNP at 22:49582364

```
22 49582364        .        A        G        198.96   .
   AC=3;
   AF=0.50;
   AN=6;
   DP=87;
   MLEAC=3;
   MLEAF=0.50;
   MQ=71.31;
   MQ0=22;
   QD=2.29;
   SB=-31.76
   GT:DP:GQ        0/1:12:99     0/1:11:89     0/1:28:37
```

INFO field

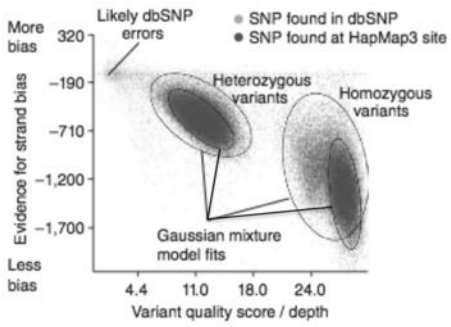
AC	No. chromosomes carrying alt allele	MLEAF	Max likelihood AF
AN	Total no. of chromosomes	MQ	RMS MAPQ of all reads
AF	Allele frequency	MQ0	No. of MAPQ 0 reads at locus
DP	Depth of coverage	QD	QUAL score over depth
MLEAC	Max likelihood AC		

Note that VQSR will only look at INFO annotations;

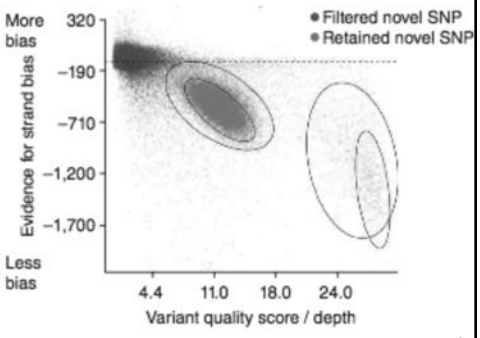
Two steps: (1) train a model then (2) apply to callset

Basic idea: training on high-confidence known sites to determine the probability that other sites are true

(1) Train model using HapMap

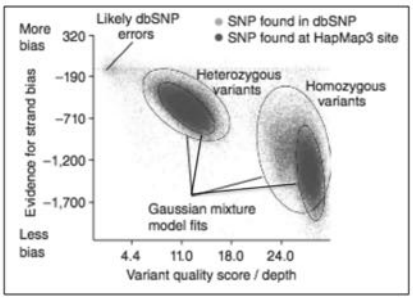


(2) Apply model to callset



(1) Training the model

(1) Train model using e.g. HapMap



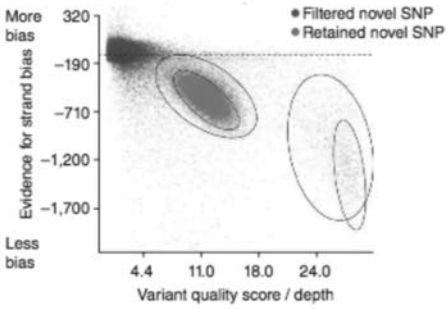
- We choose a training set
- Variants that are both in the training set and in our callset are selected.
- We train the model using the annotations of the selected variants

- This tells us **what good variants look like**
- A similar model for the variants in our callset that least look like good variants is also created (bad model, no biscuit!)
- All variants can now be ranked based on the ratio between their scores in the good model and the bad model (= VQSLOD)

(2) Applying the model to our callset

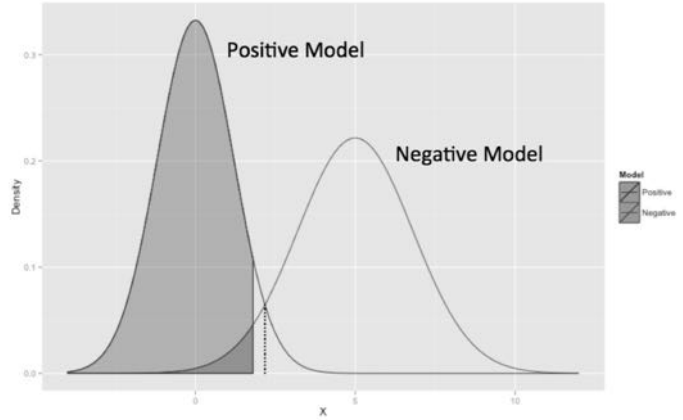
- Using the ranking produced by the model, filtering variants is as easy as setting a single threshold value
- Any variants whose score falls below the threshold is filtered out

(2) Apply model to callset



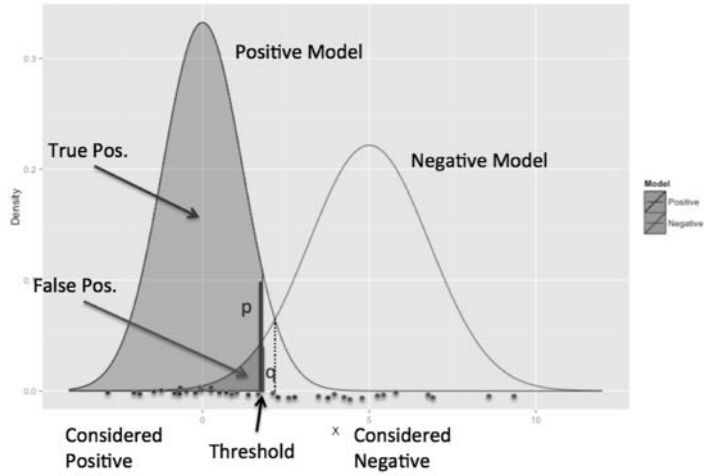
But how do we set that threshold?

There are in fact two components to the model



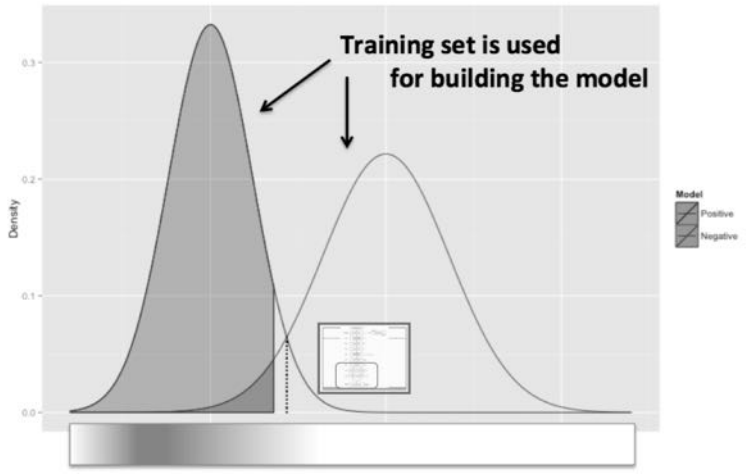
- A **negative model** is also built during training
- It represents the probability of variants to be **false positives**

The VQSLOD threshold is a tradeoff between TP and FP



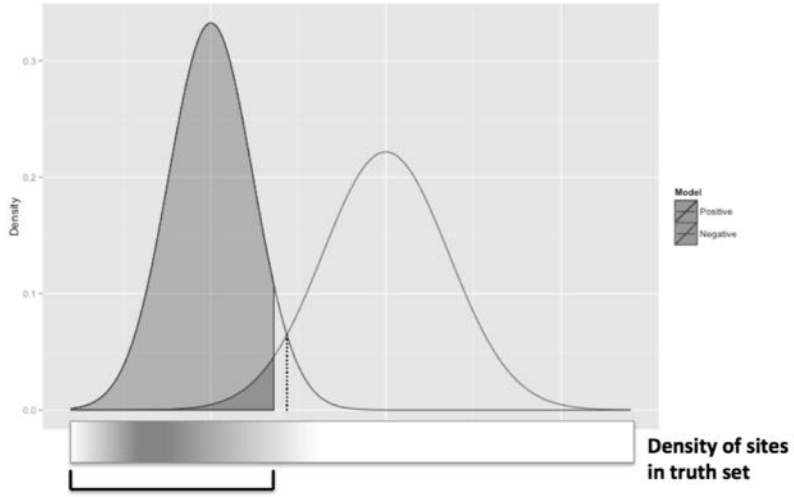
VQSLOD(x) = Log(p(x)/q(x)) (VQSLOD is distinct from QUAL!)

Role of training and truth resources



Truth set is used for translating VQSLOD values into sensitivity "tranches"

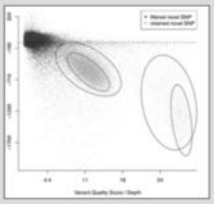
We set the threshold based on **sensitivity to truth data**



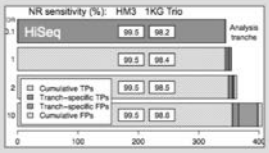
What threshold do we need to set to capture X % of the sites in the truth set?

Variant Recalibration steps & tools

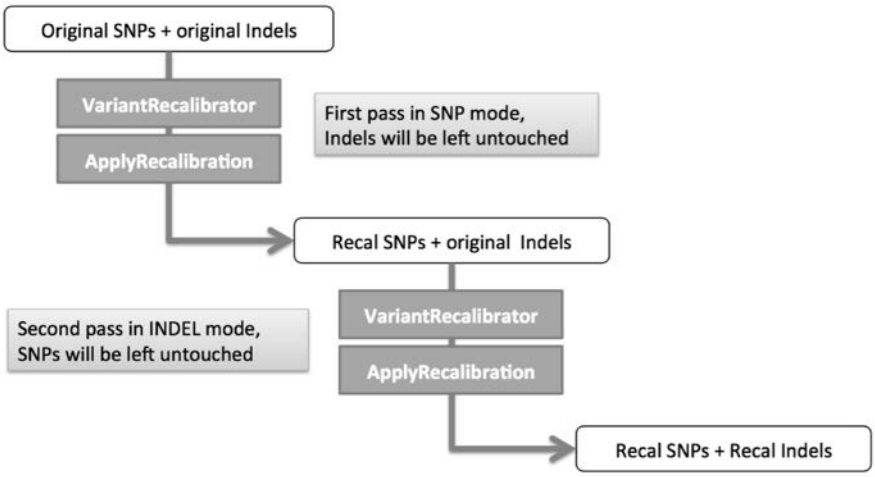
- Build and Apply the models (from resources and callset)
  - ➔ **VariantRecalibrator**



- Use VQSLOD to filter variants and write a new annotated VCF
  - ➔ **ApplyRecalibration**

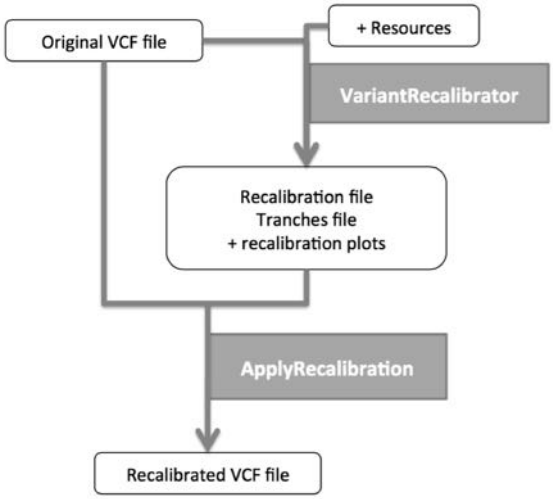


NOTE: SNPs and Indels must be recalibrated separately!



Pro-tip: Run VQSR twice in succession according to this workflow.
That way you avoid having to split them, recalibrate and combine them again.

Variant Recalibration workflow



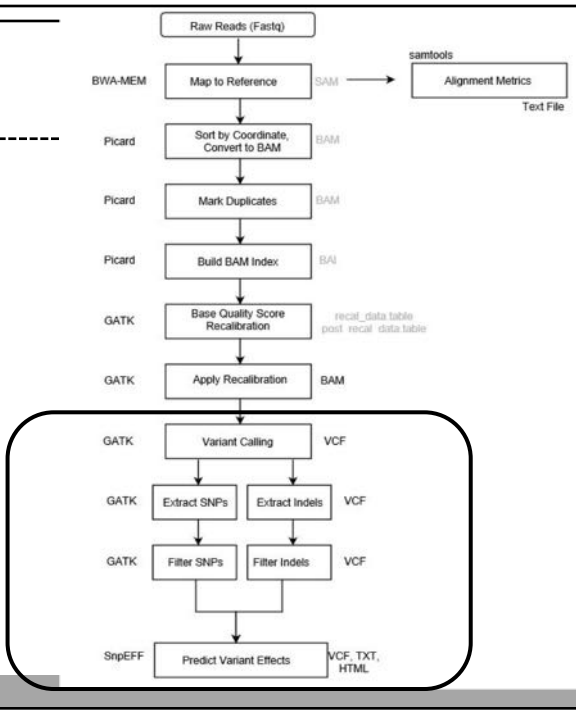
TOOL TIPS

VariantRecalibrator

- Build the Gaussian mixture model using the variants in the input callset which overlap the training data

```
java –jar GenomeAnalysisTK.jar –T VariantRecalibrator \
    –R human.fasta \
    –input raw.SNPs.vcf \
    –resource: {see next slide} \
    –an DP –an QD –an FS –an MQRankSum {…} \
    –mode SNP \
    –recalFile raw.SNPs.recal \
    –tranchesFile raw.SNPs.tranches \
    –rscriptFile recal.plots.R
```

SNP example – see documentation for indel recommendations



1) Call Variants

- We use the GATK HaplotypeCaller tool
- This step is designed to maximize sensitivity in order to minimize false negatives, i.e. failing to identify real variants
- Creates a single file with both SNPs and indels
- We extract each type of variant into it's own file so we can process them individually

2) Filter Variants

- The first step is designed to maximize sensitivity and is thus very lenient in calling variants
- Good because it minimizes the chance of missing real variants
- But means that we need to filter the raw call set in order to reduce the amount of false positives
- Important in order to obtain the the highest-quality call set possible

3) Annotation

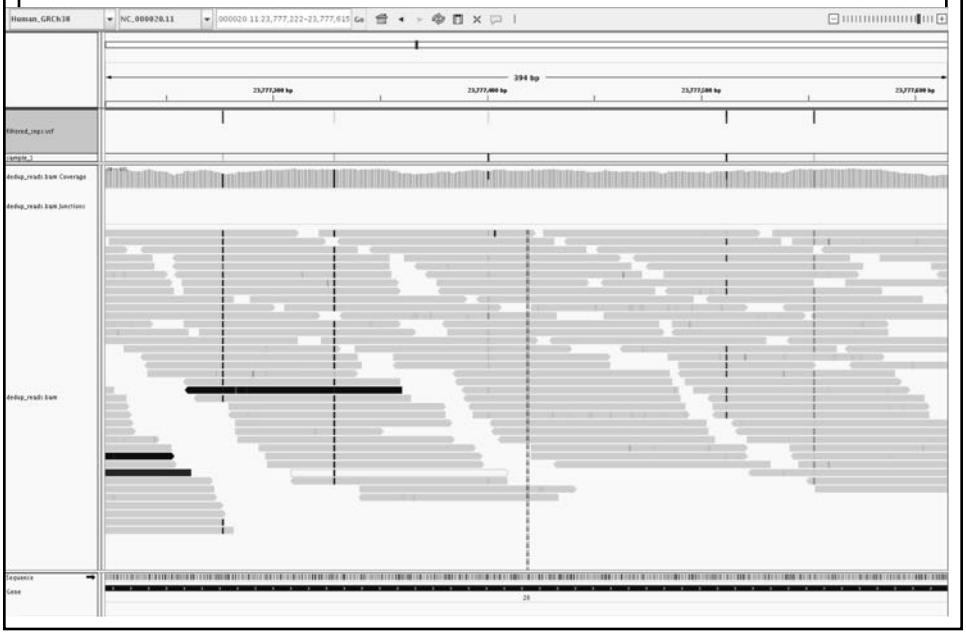
- We use SnpEff
- Annotates and predicts the effects of variants on genes
  - Codon changes
  - Amino acid changes
  - Genomic region
  - Functional effect (silent, missense)
- SnpEff has pre-built databases for thousands of genomes

Archivo VCF

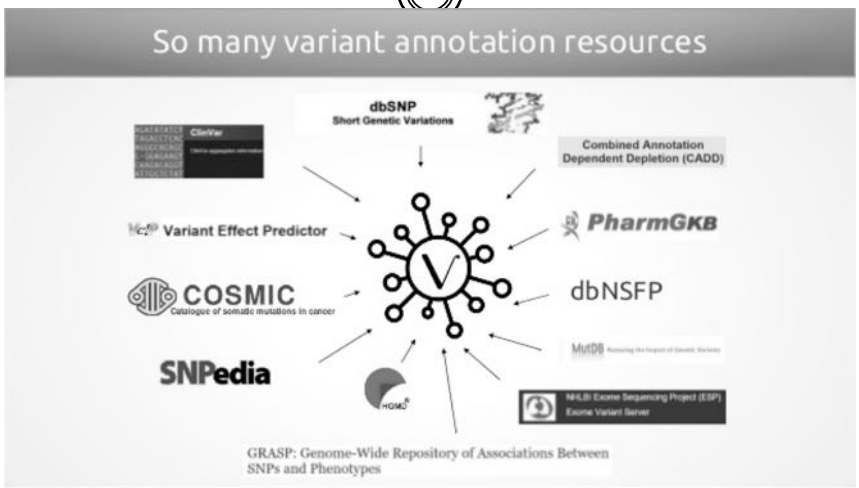
```
##fileformat=VCFv4.0
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=1000GenomesPilot-NCBI36
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF ALT    QUAL FILTER INFO                              FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G     A      29   PASS   NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:...
20     17330   .         T     A      3    q10    NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3   0/0:41:3
20     1110696 rs6040355 A     G,T    67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2   2/2:35:4
20     1230237 .         T     .      47   PASS   NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20     1234567 microsat1 GTCT  G,GTACT 50  PASS   NS=3;DP=9;AA=G                     GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

http://www.internationalgenome.org/wiki/Analysis/vcf4.0/

4) Visualization - IGV



Variant Annotation



Variant Annotation: SnpEff

- Variant annotation and effect prediction tool. It annotates and predicts the effects of genetic variants (such as amino acid changes).
- Many effects are calculated: such as SYNONYMOUS_CODING, NON_SYNONYMOUS_CODING, FRAME_SHIFT, STOP_GAINED just to name a few.

SnpEff: Public databases

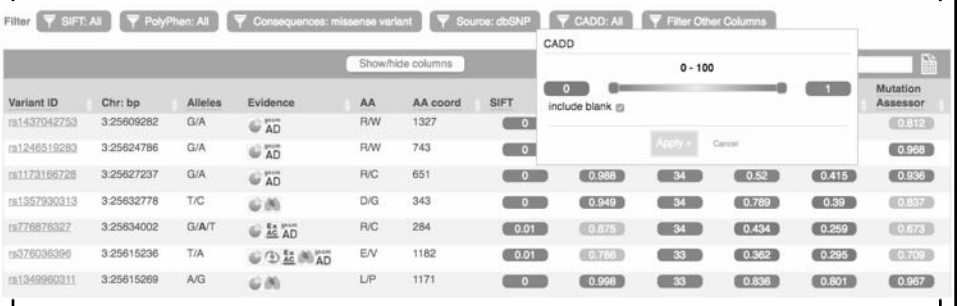
- **ENCODE** datasets are supported by SnpEff (by means of BigWig files provided by ENCODE project).
- **Epigenome Roadmap** provides data-sets that can be used with SnpEff.
- **TFBS** Transcription factor binding site predictions can be annotated. Motif data used in this annotations is generates by Jaspar and ENSEBML projects
- **NextProt** database can be used to annotate protein domains as well as important functional sites in a protein (e.g. phosphorilation site)

CADD - Combined Annotation Dependent Depletion

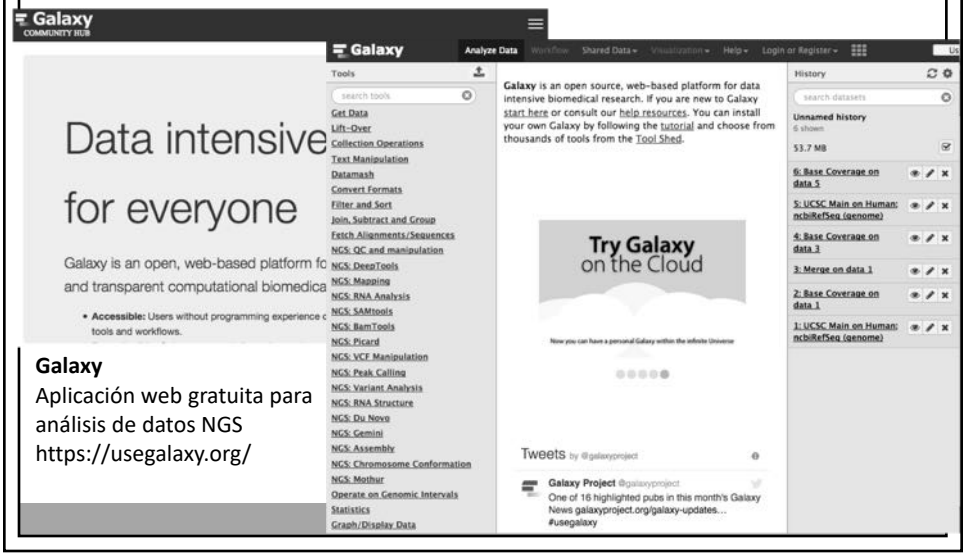
- Framework that integrates multiple annotations into one metric by contrasting variants that survived natural selection with simulated mutations
- C-scores strongly correlate with allelic diversity, pathogenicity of both coding and non-coding variants, and experimentally measured regulatory effects, and also highly rank causal variants within individual genome sequences.
- C-scores of complex trait-associated variants from genome-wide association studies (GWAS) are significantly higher than matched controls and correlate with study sample size, likely reflecting the increased accuracy of larger GWAS.

https://cadd.gs.washington.edu/

Puntajes CADD



Software Libre



Galaxy
Aplicación web gratuita para análisis de datos NGS
https://usegalaxy.org/

Resources in NGS data analysis

Public forums:

SEQanswers
the next generation sequencing community

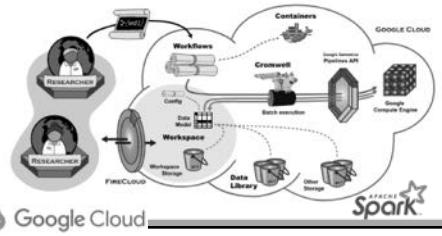
Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

BioBits: A Blog about bioinformatics

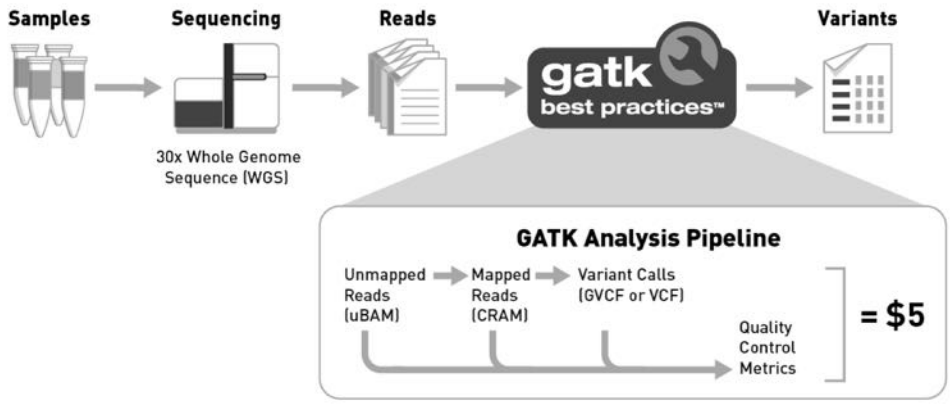
Biostars
BIOINFORMATICS EXPLAINED

BROAD INSTITUTE **FireCloud**

An open platform for secure and scalable analysis on the cloud



Google Cloud



	2016	2017	2018 (March)
# of Whole Genomes Analyzed	900	900	900
Total Compute Cost	$40,500	$12,150	$4,500
Cost per Genome Analyzed	$45	$13.50	$5