VCF and RDF

Jeremy J Carroll

April 3rd, 2013

Prepared for W3C Clinical Genomics Task Force

www.Syapse.com

jjc@Syapse.com

Contents

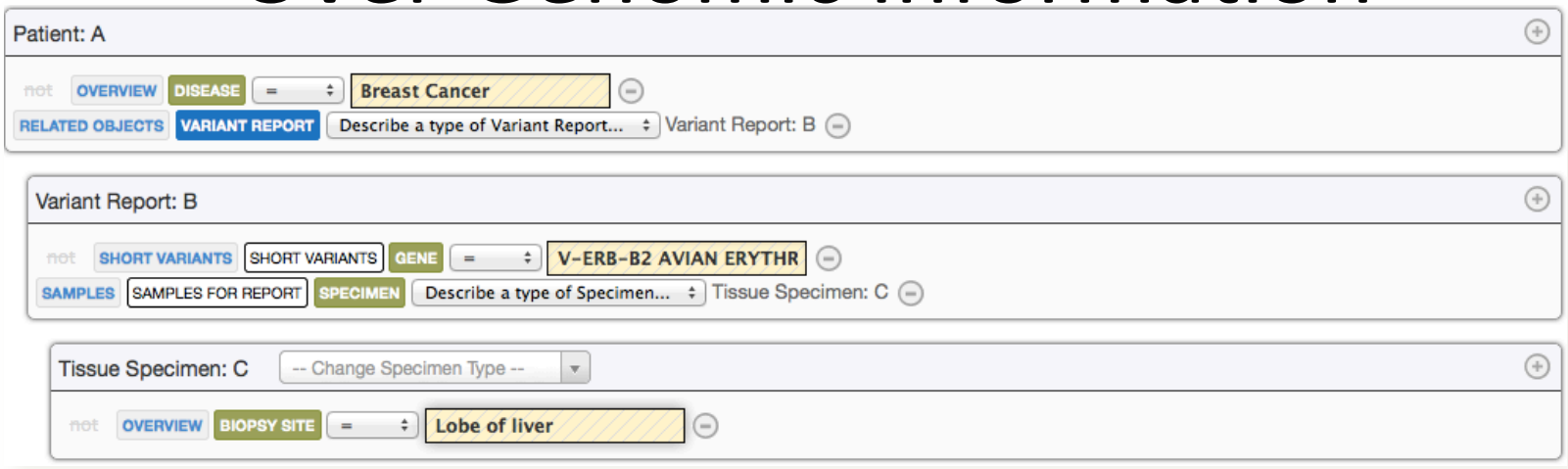
- Background
  - Business Background
  - Goal
  - VCF and RDF
- This specific work (exploration)
  - Stats
  - VCF example
  - Lots of example transforms: VCF and RDF
- Discussion: might the examples meet the business goals?

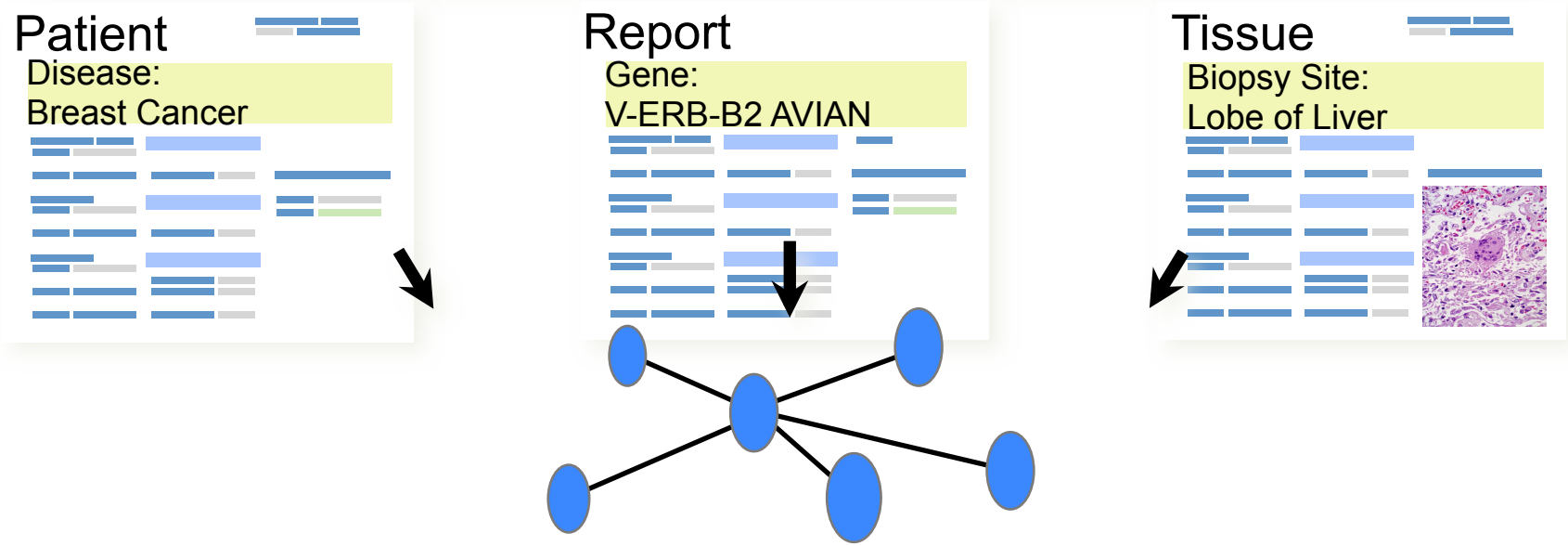
Background

- *Syapse Discovery provides an end-to-end solution for […] laboratories deploying next-generation sequencing-based diagnostics, […] configurable semantic data structure enables users to bring omics data together with traditional medical information […]*
- This work: initial exploration of VCF import, what useful information is in a VCF etc.

Syapse

Success = Useful Advanced Query Over Genomic Information





VCF, a Web Based Knowledge Format

- Variant Call Format
- Used for exchange from one system to another
- Used for exchange from one lab to another
- Used for exchange between universities and the wider community

RDF, a Web Based Knowledge Format

- Resource Description Framework
- Used for exchange from one system to another
- Used for exchange from one lab to another
- Used for exchange between universities and the wider community

VCF vs RDF

- VCF: efficient representation of huge quantities of data

- RDF: flexible representation with excellent interoperability and query

Materialized or Virtual?

- Materialized = Transform data:
  Extract Transform Load into RDF

- Virtual = Transform query:
  Runtime mapping

- Or some combination: transform VCF into some internal format, transform SPARQL queries into a query over internal format

This Work

- Initial mapping from VCF to RDF
- Materialized for simplicity

- Key questions:
  - Is this useful?
    - Would query answer interesting questions?
  - Scale?
    - Can we materialize? Should we go virtual?

Known limitations

- Lots not addressed ….
- Too many literals not enough URIs
- Slow

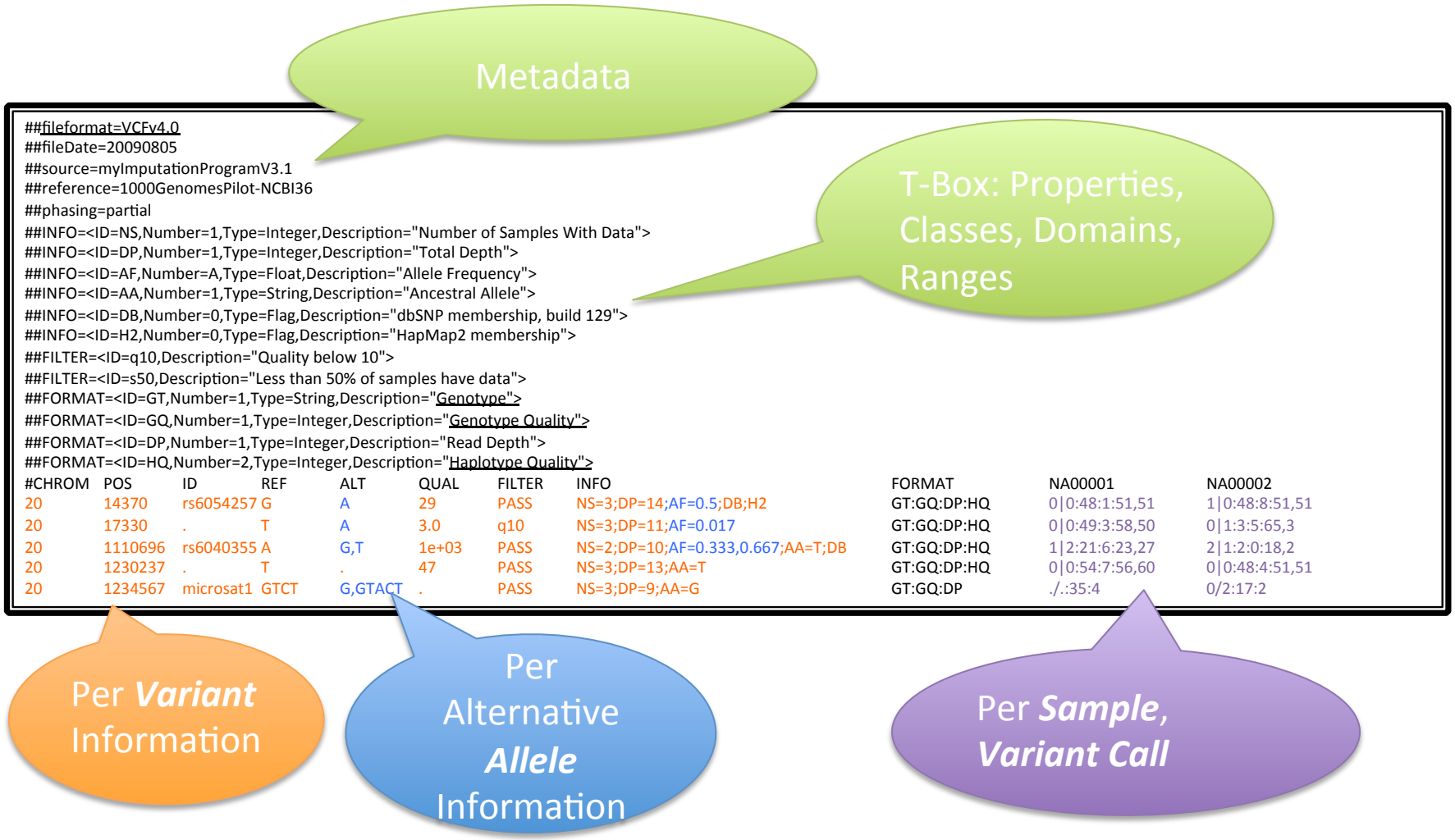
- Hoped to be fit for purpose, i.e. answering questions, not production code. (scale and utility, see previous slide)

Some Stats

- Working with Chromosome 7 from 1000G
  ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/release/20110521/
  ALL.chr7.phase1_release_v3.20101123.snps_indels_svs.genotypes.vcf.gz
- 704,243 Variants, 1,092 samples
  - 689,034,445 ref calls, 79,998,911 variant calls
- H/W: new mac book pro 2.3 GHz Intel Core i7 8 GB
- Gunzip: 1m37s
- PyVCF: 2h41m
- VCF2TTL: 37h57m
- serdi TTL 2 Ntriples: 13h57m
- triples: 14,780,932,912 (1.2 trillion bytes)

Syapse

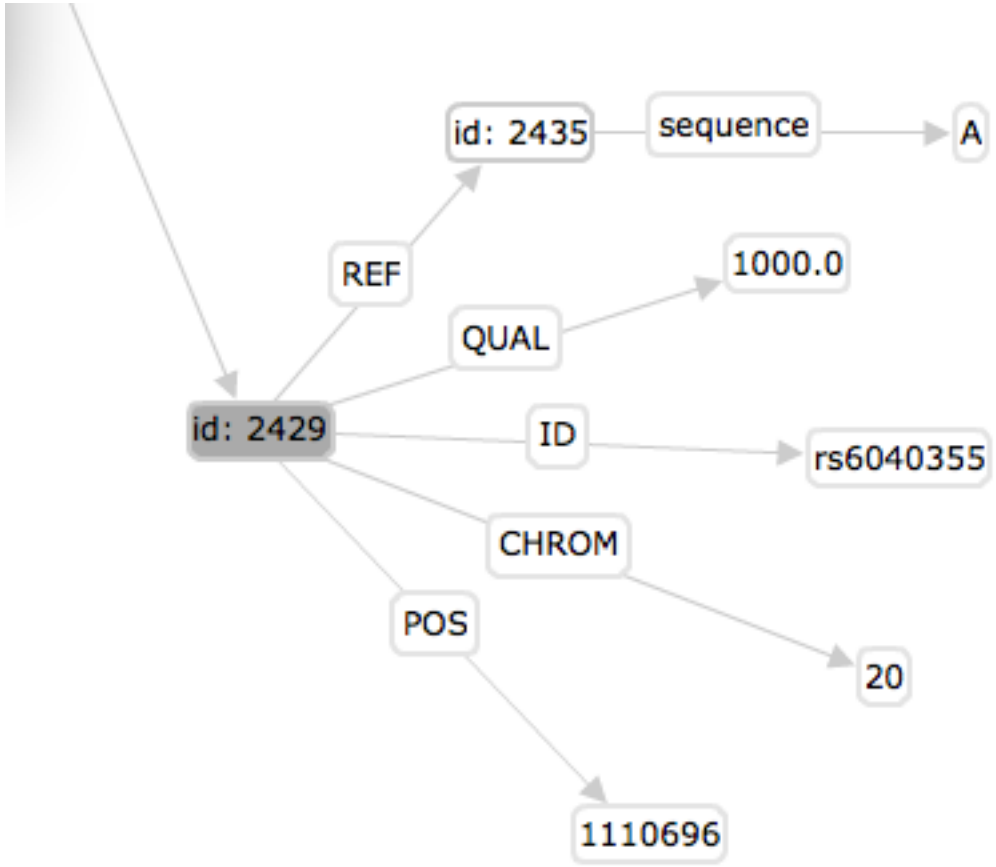
VCF Overview



Key Classes

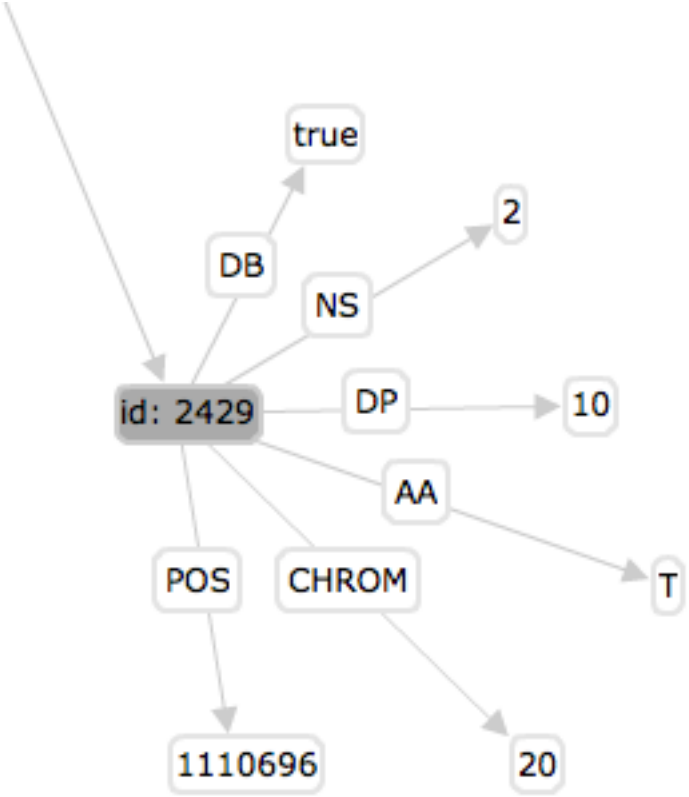
- ***Variant*** A position in the reference genome, where mutation may happen, and related information.
- ***Allele*** A possible sequence of bases at some variant, and related information.
- ***Sample*** A related set of Variant Calls
- ***Variant Call*** A selection of two Alleles (diploid) at a Variant (can be phased or unphased)

A Variant



#CHROM	POS	ID	REF	ALT	QUAL
20	1110696	rs6040355	A	G,T	1e+03

INFO about a Variant

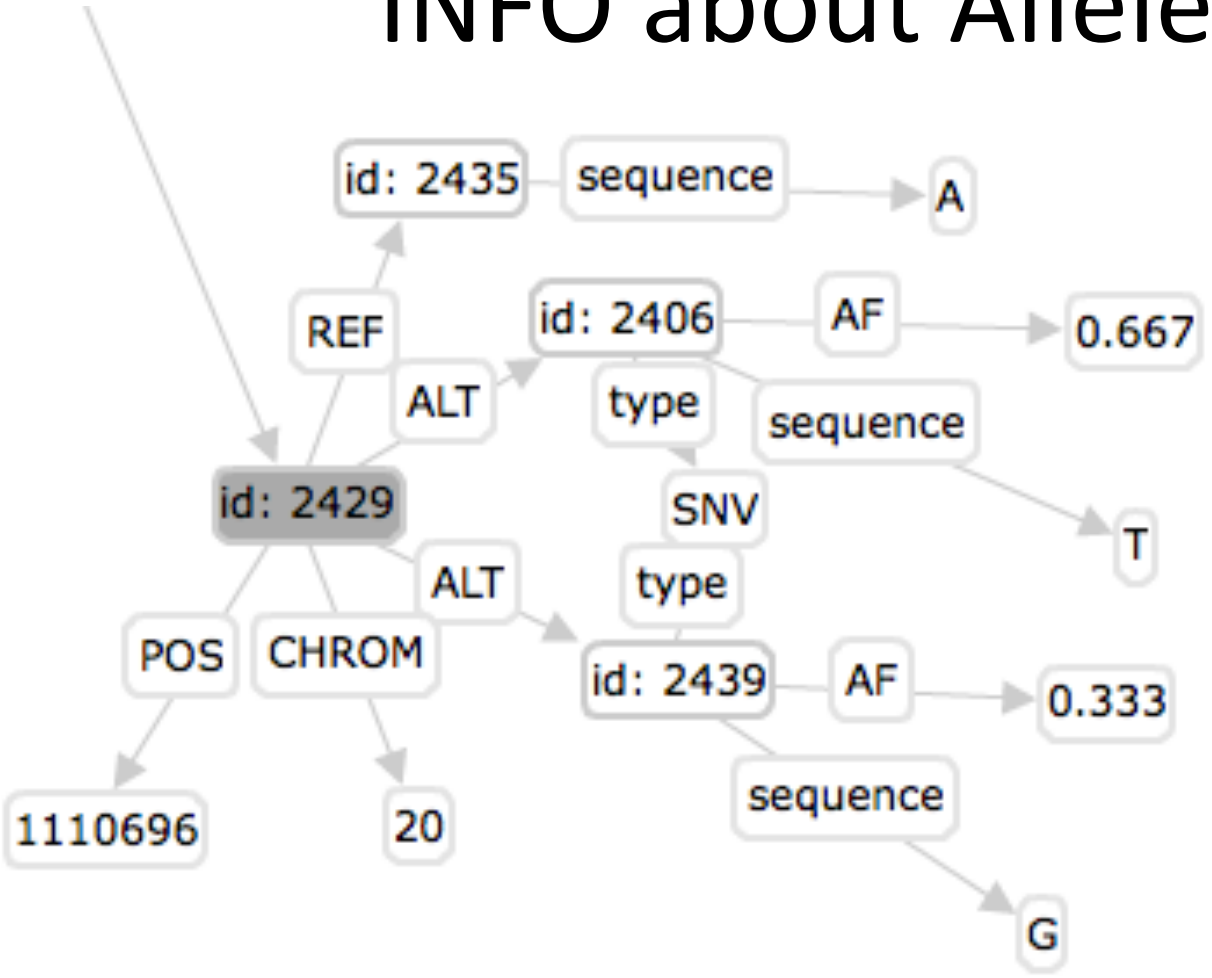


Same node as on previous slide, selecting different triples for display.

DB property should probably be a class (dbSNP membership); AF (allele frequency) is not a property of the variant but the allele.

#CHROM	POS	INFO
20	1110696	NS=2;DP=10;AF=0.333,0.667;AA=T;DB

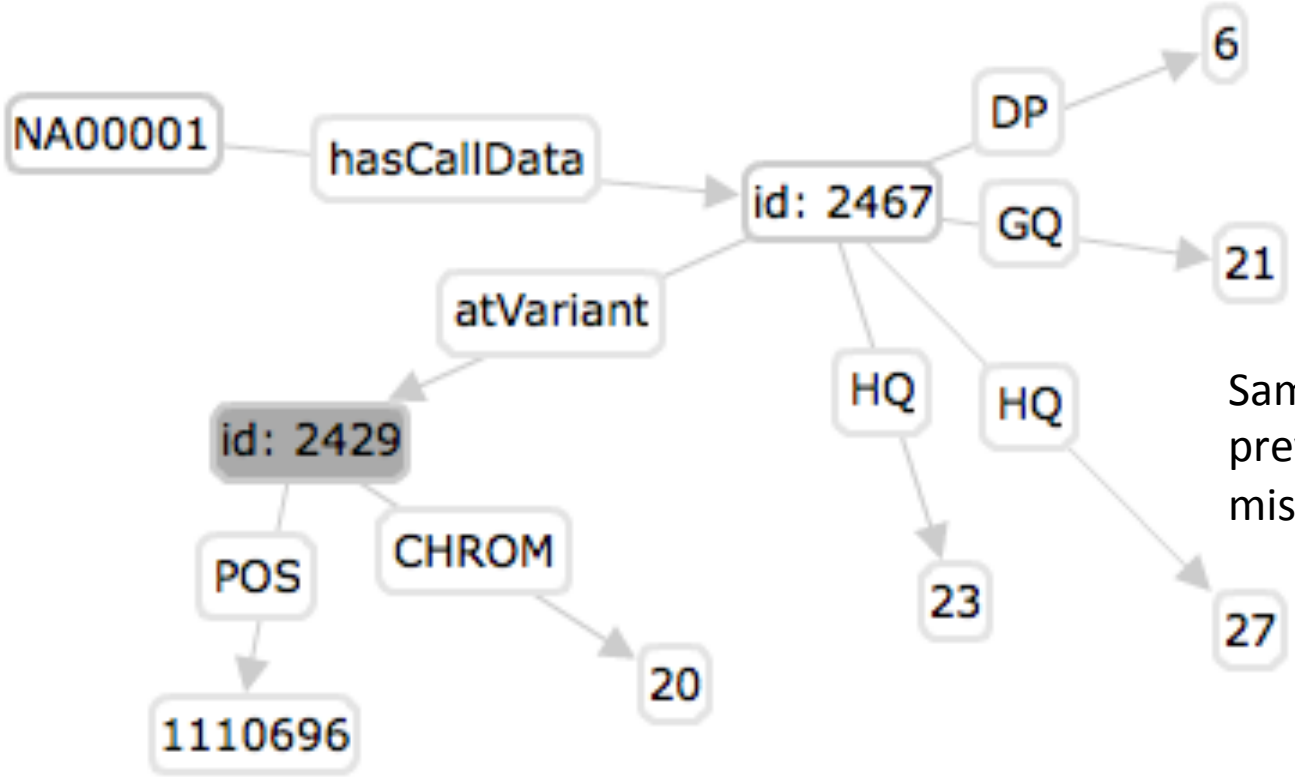
INFO about Alleles



Same node as on previous slide, selecting different triples for display.

#CHROM	POS	REF	ALT	INFO
20	1110696	A	G,T	AF=0.333,0.667;AA=T;DB

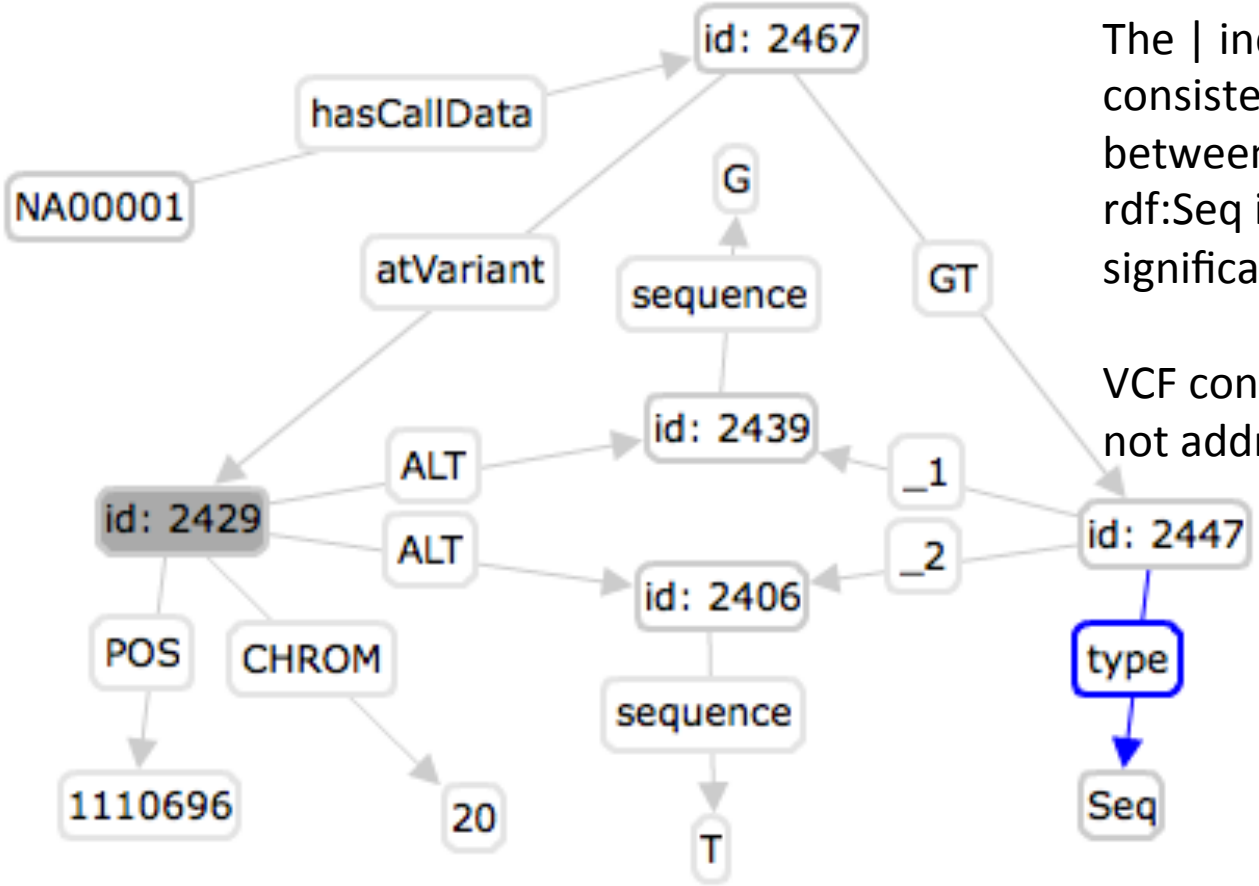
Detail about a Variant Call



Same variant node as on previous slide, HQ is mismodeled.

#CHROM	POS	REF	ALT	FORMAT	NA00001
20	1110696	A	G,T	GT:GQ:DP:HQ	1|2:**21:6:23,27**

A Phased Diploid Genotype Call: *rdf:Seq*

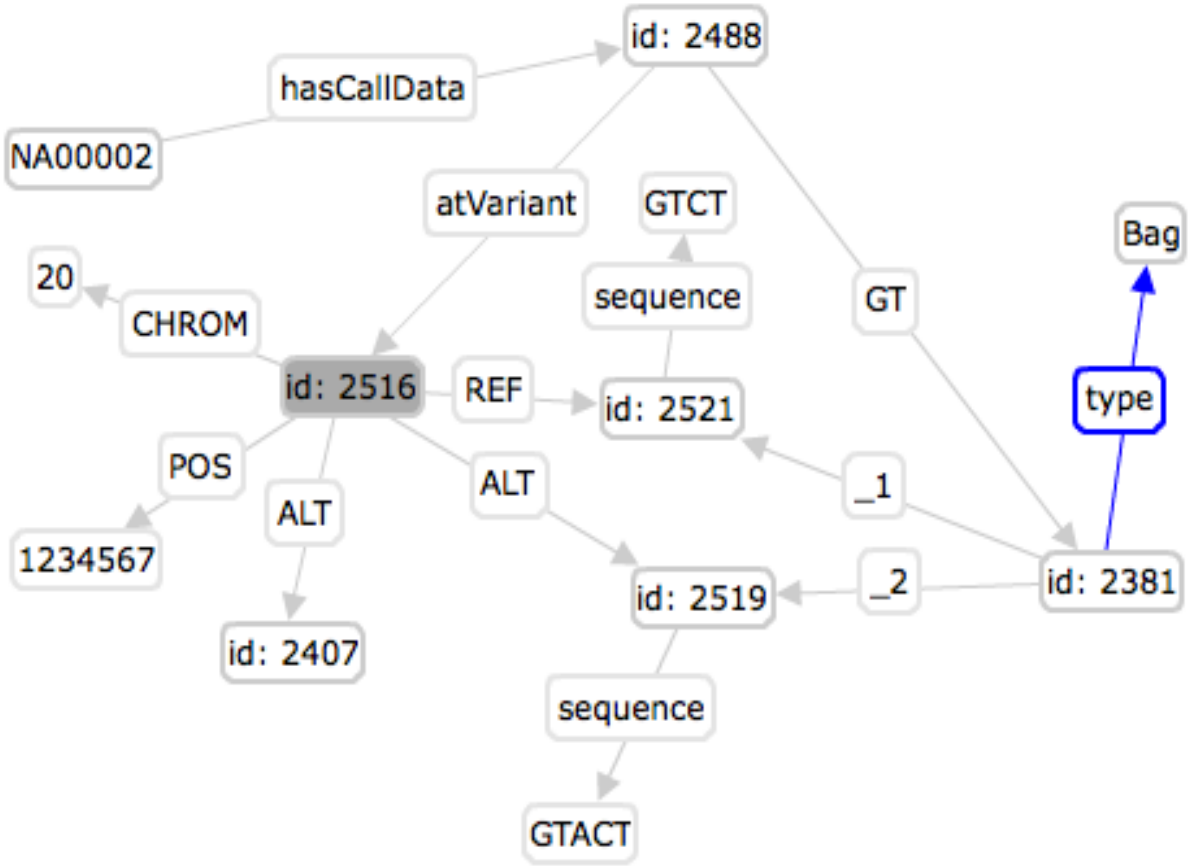


The | indicates phased, consistent ordering between VCF rows. rdf:Seq indicates order is significant. Heterozygous

VCF concept of phase set not addressed.

#CHROM	POS	REF	ALT	FORMAT	NA00001
20	1110696	A	G,T	**GT**:GQ:DP:HQ	**1|2**:21:6:23,27

Unphased Diploid Genotype Call: ***rdf:Bag***



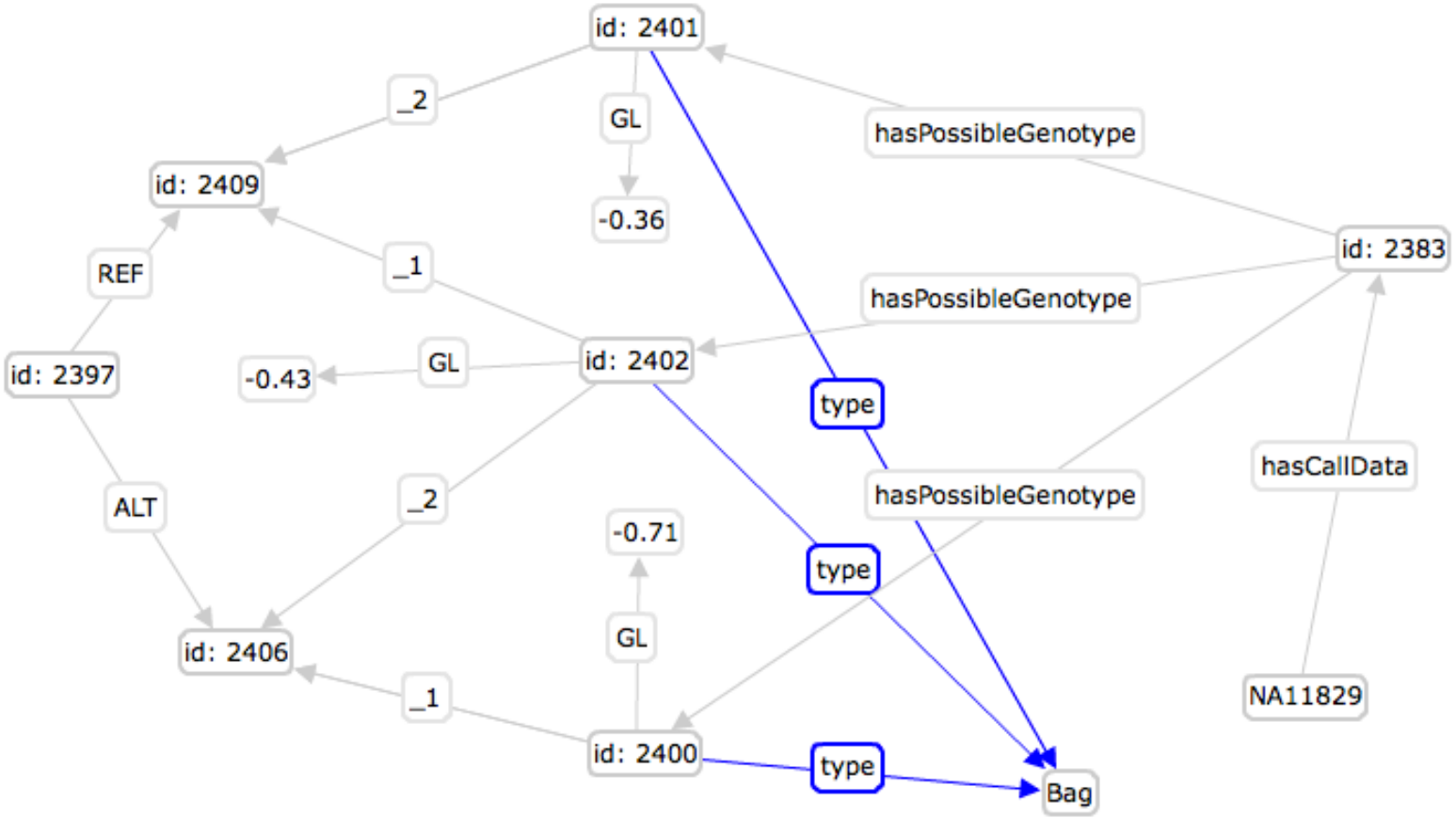
The / indicates unphased genotype. rdf:Bag indicates order is not significant. rdf:Bag also used for homozygous genotypes. The 0/2 indicates the use of the reference and the 2nd alternative.

This is a different variant

#CHROM	POS	REF	ALT	FORMAT	NA00002
20	1234567	GTCT	G,GTACT	GT:GQ:DP	0/2:17:2

Syapse

Genotype Likelihoods 1000G



#CHROM	POS	REF	ALT	FORMAT	NA11829
7	16161	A	G	GT:DS:GL	0|1:1.300:–0.36,–0.43,–0.71

Notes: _1 and _2 hide one another, id_2397 is variant of id_23830

Discussion Points

- How much of this is actually useful?
  - Maybe just artifacts of the history of the call?
  - Maybe just joins that we could/should recompute on the fly
  - Maybe much of the data is fundamentally uninteresting
- Do we know the queries? If so can we represent in a more compact way and optimize for those queries?

Further observations

- Metadata is a mess, not actually reusable by machine, e.g. why file date but not sample date … provokes the desire to show rationale for file, but not the performance
- Extensibility of Tbox allows English text to introduce new syntax (e.g. enumerations, per alt allele and reference). This is not a good idea.
- The per genotype call info is only for unphased data, yet the call can be phased