CLINVITAE: an open database of clinically observed variants, and other open source tools from Invitae

Reece Hart*, Bruce Blyth, Tim Chiu, Vince Fusaro, John Garcia, Emily Hare, Kevin Jacobs, Naomi Most, Joshua Paul, Rudy Rico, Jody Westbrook

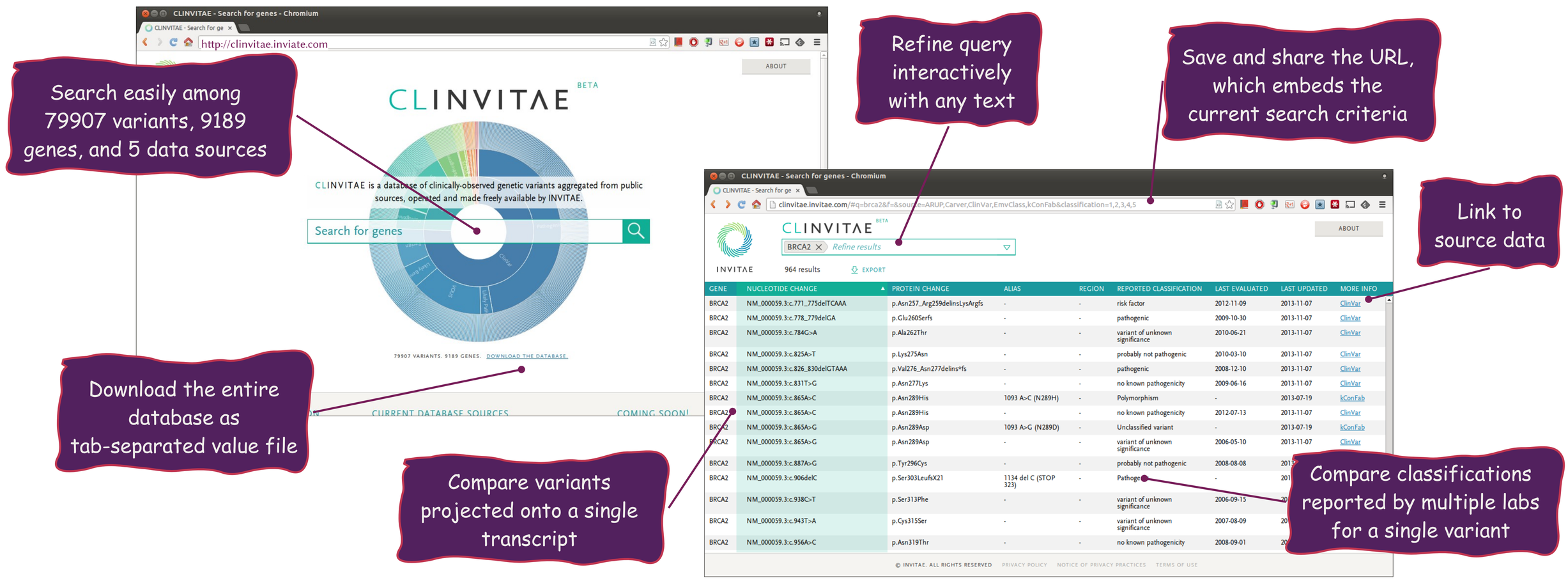
Invitae, Inc. ◆ invitae.com ◆ 458 Brannan St., San Francisco, CA 94107

Variants of uncertain significance (VUS) are routinely identified in clinical and research sequencing projects. Assessing these variants for clinical relevance presents a significant analysis burden for clinical geneticists and genetic counselors.

CLINVITAE is a free, comprehensive, and easy-to-use resource that aggregates variants and reported pathogenicity. It is especially focused on variants not currently in ClinVar.

The current beta release contains 79,907 variants in 9,189 genes from 5 sources. A new version of CLINVITAE with new sources, refreshed data, and additional features is underway.

CLINVITAE was built using our HGVS and UTA variant mapping tools, which are described at the bottom of this poster.



Other open source tools from Invitae

HGVS Parsing, Formatting, and Mapping in Python

http://bitbucket.org/invitae/hgvs

Genome, transcript, and protein sequence variants are typically reported using recommendations provided by the Human Genome Variation Society (HGVS) [1]. The complexity of this standard makes it difficult to use in software.

We have developed a easy-to-use and freely-available Python library for parsing, representing, formatting, and mapping variants between genome, transcript, and protein sequences. The current implementation handles most of the standard for precisely defined sequence variants.

[1] http://www.hgvs.org/mutnomen/

http://www.ncbi.nlm.nih.gov/projects/SNP/snp_ref.cgi?rs=509749

```
In [1]: import bdi.sources.uta0_pg
        import hgvs.hgvsmapper
        import hgvs.parser

In [2]: bdi = bdi.sources.uta0_pg.UTA0()
        hgvsmapper = hgvs.hgvsmapper.HGVSMapper(bdi)
        hgvsparser = hgvs.parser.Parser()

In [3]: var_c1 = hgvsparser.parse_hgvs_variant('NM_001261456.1:c.1762A>G')
        var_p1 = hgvsmapper.hgvsc_to_hgvsp(var_c1, None)
        var_c1, var_p1
Out[3]: (Variant(ac=NM_001261456.1, type=c, posedit=1762A>G),
         Variant(ac=NP_001248385.1, type=p, posedit=Met588Val))

In [4]: var_g = hgvsmapper.hgvsc_to_hgvsg(var_c1,'GRCh37.p10')
        var_g
Out[4]: Variant(ac=NC_000001.10, type=g, posedit=160793560A>G)

In [5]: txs = bdi.get_tx_for_gene('LY9')
        len(txs)
        [ tx['ac'] for tx in txs ]
Out[5]: ['NM_002348.3', 'NM_001261456.1', 'NM_001261457.1', 'NM_001033667.2
        ']

In [6]: var_c2 = hgvsmapper.hgvsg_to_hgvsc(var_g,'NM_001261457.1')
        var_p2 = hgvsmapper.hgvsc_to_hgvsp(var_c2, None)
        var_c2, var_p2
Out[6]: (Variant(ac=NM_001261457.1, type=c, posedit=1534A>G),
         Variant(ac=NP_001248386.1, type=p, posedit=Met512Val))
```

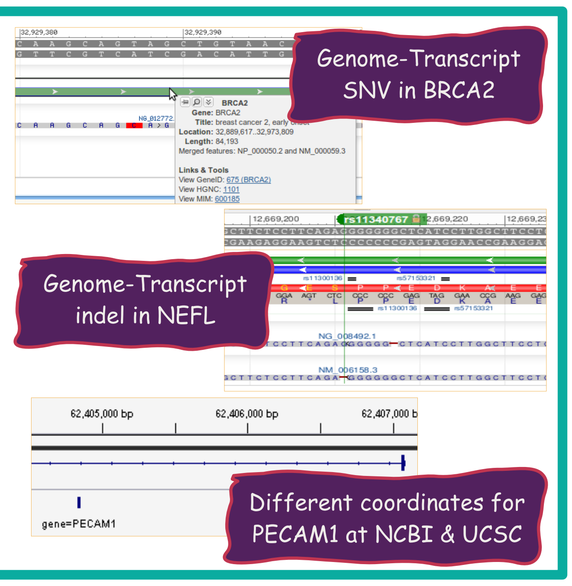
UTA: The Universal Transcript Archive

http://bitbucket.org/invitae/uta

PostgreSQL: uta.invitae.com:5432, uta_public/uta_public

Transcripts are the lens through which variants are reported and interpreted. Having a consistent, shared view of transcripts is essential to accurately inferring the clinical significance of sequence variants.

UTA provides the data necessary to map variants between genome assemblies and transcripts, and enables "liftover" between pairs of transcripts via a common reference assembly. The upcoming UTA release will identify alignment discrepancies across sources in order to identify regions where interpretation may be ambiguous.



Illuminate: shedding light on Illumina sequencing metrics

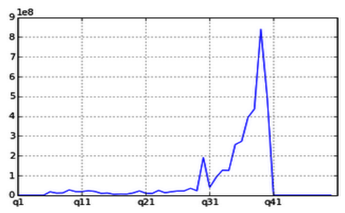
http://bitbucket.org/invitae/illuminate

Illuminate is a Python library that provides programmatic access to metadata and metrics from Illumina sequencers.

Illuminate provides access to every raw data point the Illumina sequencers collect during their runs, while streamlining delivery of the most commonly desired attributes like mean cluster density, q-score distribution, and indexing characteristics.

Illuminate enables monitoring of in-progress sequencing runs, aggregation of metrics for quality control, and interactive analysis via the Python pandas package.

```
In [1]: from illuminate import InteropDataset

In [2]: ID = InteropDataset('/Users/nthmost/projects/bitbuck

In [3]: QM = ID.QualityMetrics()

In [4]: print "Q Score HeatMap Across All Reads"
        QM.idf.sum().plot()
        Q Score HeatMap Across All Reads
Out[4]: <matplotlib.axes.AxesSubplot at 0x108c49e10>
```



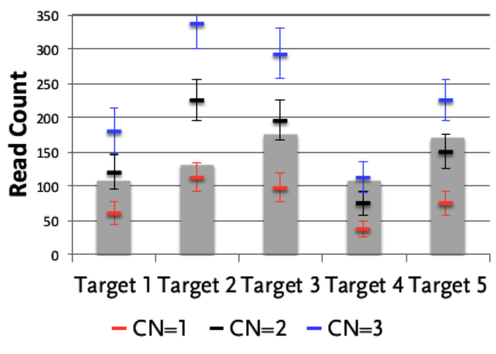

```
In [8]: print "% >= Q30 Per Read"

        print QM
        % >= Q30 Per Read
        Read 1: 92.598490
        Read 2: 92.008049 (Index)
        Read 3: 86.113045
```

CNVitae: Accurate detection of small and large copy number events from targeted next-generation sequence data

Coming Soon!

CNVitae is designed to detect single-exon CNVs as well as larger regions from next-generation sequence data. Its algorithm is based on a statistical model for read counts and employs model-based segmentation algorithms optimized for use with sparsely distributed and highly variable targets across the genome.

This framework estimates the most likely copy number for all segments, and, critically for clinical use, each called segment is assigned a robust quality score indicating confidence in the copy number determination.



Want more info?
ASHG 2013 CNVitae Poster
db.tt/0hCzK833