May 28, 2020

Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology

PLOS One

Shray Alag[1]

[1]The Harker School, San Jose

1 Works for me dx.doi.org/10.17504/protocols.io.bfacjiaw

Shray Alag

ABSTRACT

Researchers and clinicians face a significant challenge in keeping up-to-date with the rapid rate of new associations between genetic mutations and diseases. To remedy this problem, this research mined the ClinicalTrials.gov corpus to extract relevant biological insights, produce unique reports to summarize findings, and make the meta-data available via APIs. An automated text-analysis pipeline performed the following features: parsing the ClinicalTrials.gov files, extracting and analyzing mutations from the corpus, mapping clinical trials to Human Phenotype Ontology (HPO), and finding associations between clinical trials and HPO nodes. Unique reports were created for each mutation (SNPs and protein mutations) mentioned in the corpus, as well as for each clinical trial that references a mutation. These reports, which have been run over multiple time points, along with APIs to access meta-data, are freely available on http://snpminertrials.com. Additionally, HPO was used to normalize disease terms and associate clinical trials with relevant genes. The creation of the pipeline and reports, the association of clinical trials with HPO terms, and the insights, public repository, and APIs produced are all novel in this work. The freely-available resources present relevant biological information and novel insights between biomedical entities in a robust and accessible manner, mitigating the challenge of being informed about new associations between mutations, genes, and diseases.

EXTERNAL LINK

http://snpminertrials.com

THIS PROTOCOL ACCOMPANIES THE FOLLOWING PUBLICATION

Alag S (2020) Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. PLoS ONE 15(5): e0233438. doi: 10.1371/journal.pone.0233438

MATERIALS TEXT

Two publicly-available datasets were used in this study: ClinicalTrials.gov and HPO.

1 Two publicly-available datasets were used in this study: ClinicalTrials.gov and HPO.

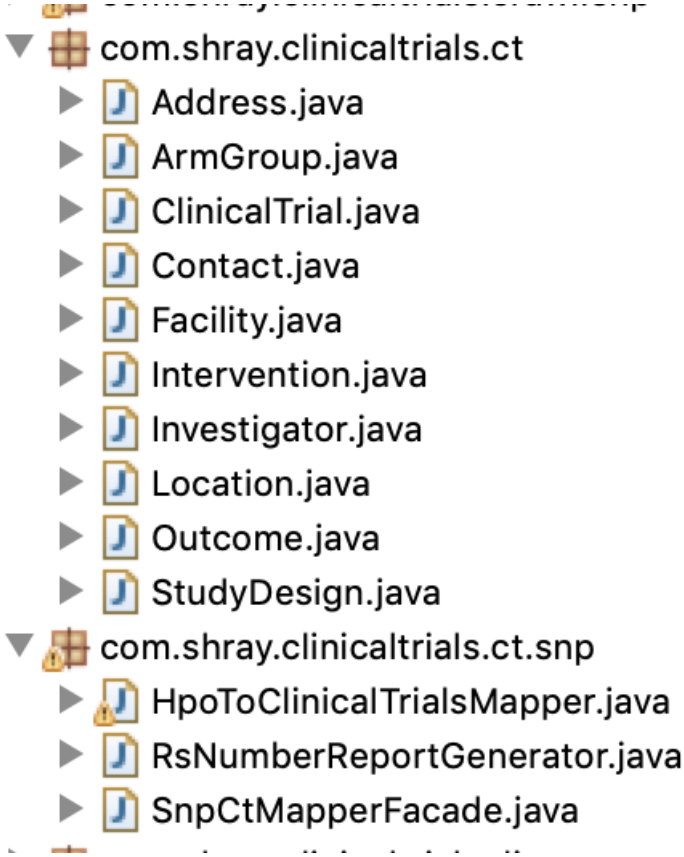
1.1 Clinical Trial Data

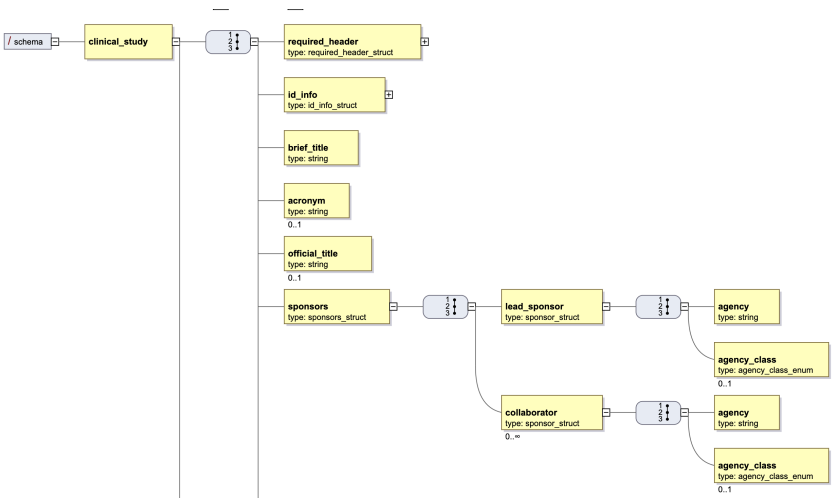
The complete repository of clinical trials displayed at ClinicalTrials.gov is available in XML format with a well-defined schema.

Download: https://clinicaltrials.gov/ct2/resources/download
Download all clinical trials in a zip file (1.2 GB): https://clinicaltrials.gov/AllPublicXML.zip

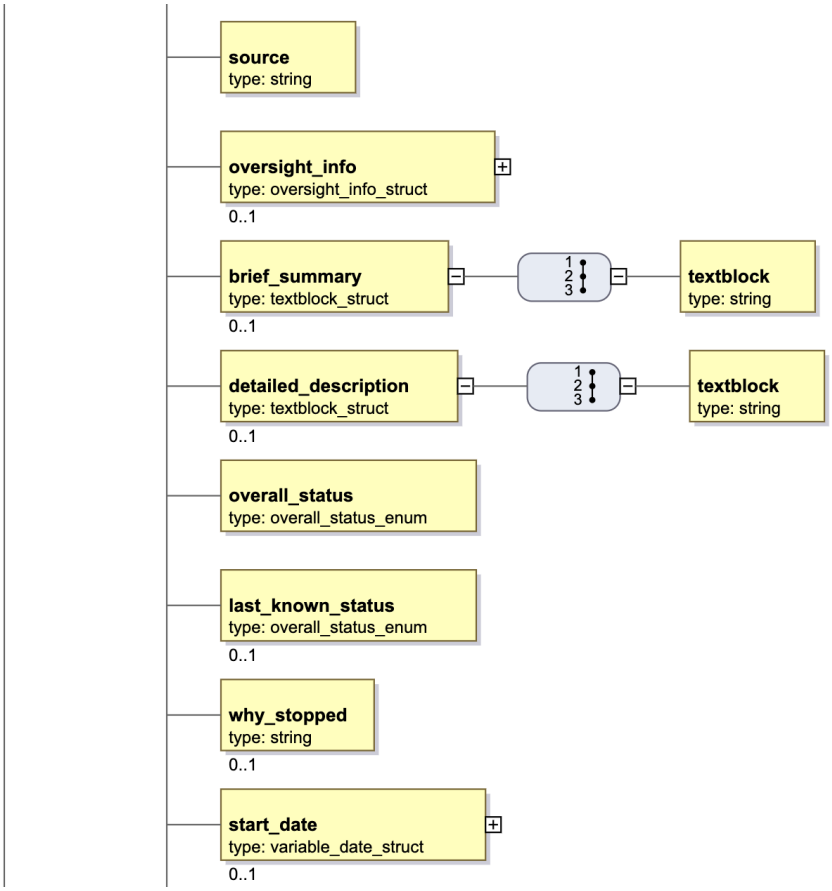
Parse XML. Schema files are below

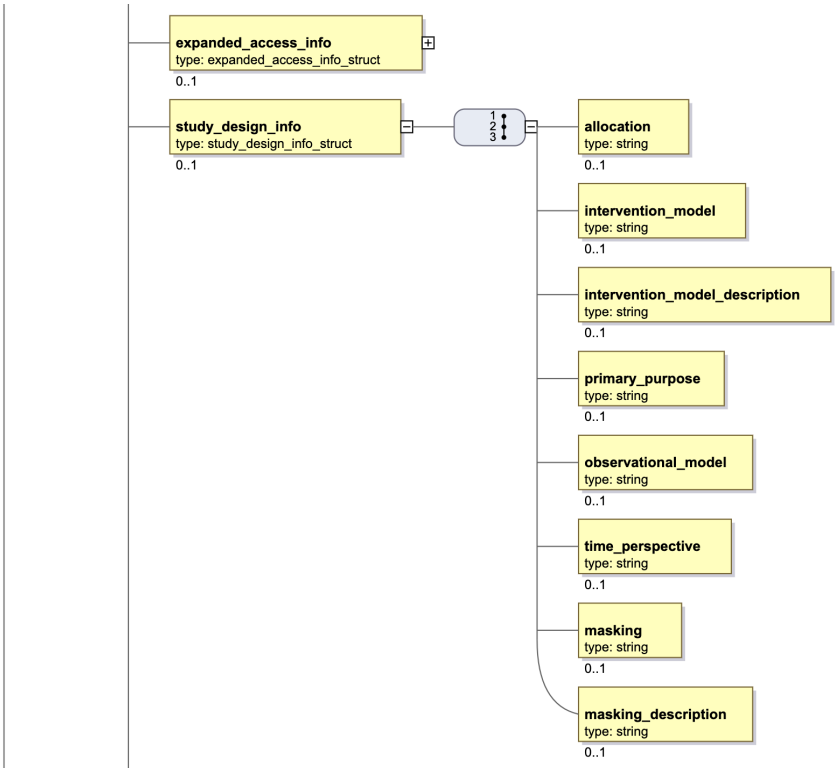
Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw



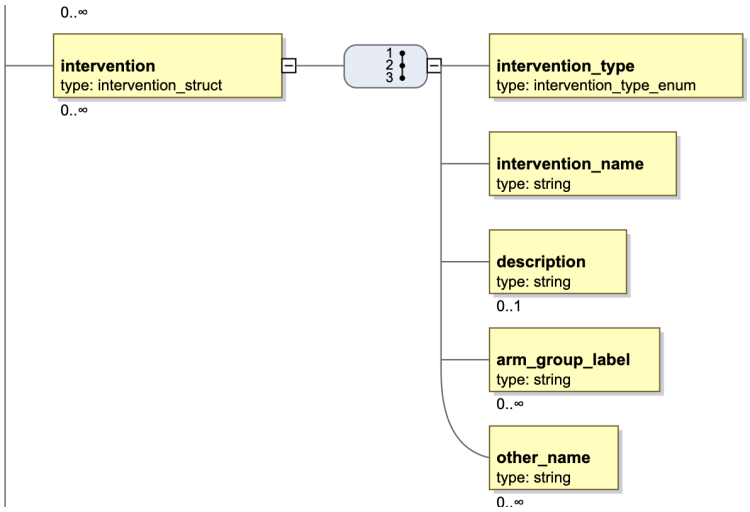


Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw





Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw

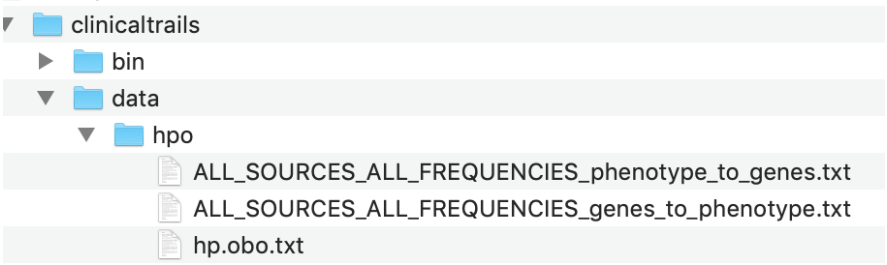


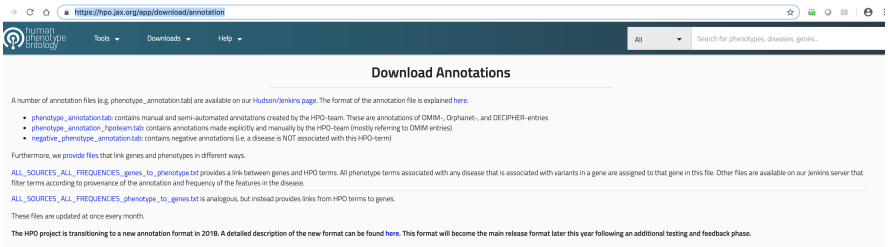
1.2

HPO - Human Phenotype Ontology

HPO is a standardized vocabulary of phenotype abnormalities that are seen in humans. HPO is a product of the Monarch Initiative and one of the thirteen driver projects in the Global Alliance for Genomics and Health strategic roadmap. The HPO ontology files are available in the OBO flat-file format and are easy to read and parse. HPO annotations provide a correlation between HPO terms and genes. There are three annotation files that contain associations between genes and phenotypes. The HPO files used in this project consisted of 14,961 HPO nodes, with 18,547 parent-child relationships between the nodes. Furthermore, 820,297 gene-phenotype annotations mapped across 4,312 unique genes and 8,947 individual HPO terms.

Human Phenotype Ontology -- HPO Ontology https://hpo.jax.org/app/
Disease terms with associated associations to genes
Ontology download OBO flat file http://purl.obolibrary.org/obo/hp.obo
Annotation download https://hpo.jax.org/app/download/annotation





Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw

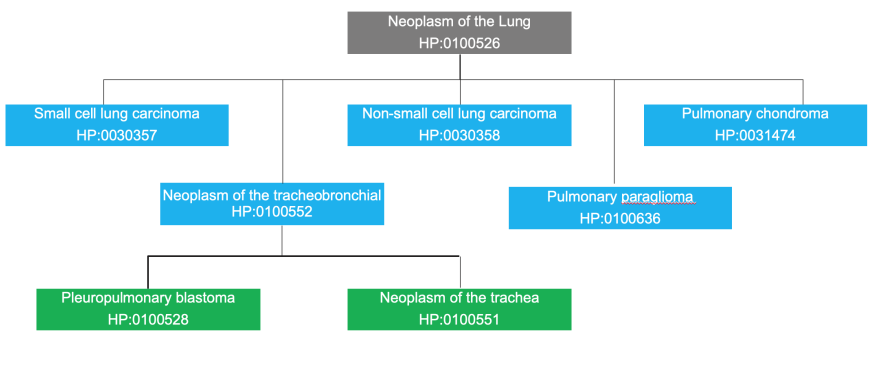
```
[Term]
id: HP:0000001
name: All
comment: Root of all terms in the Human Phenotype Ontology.
xref: UMLS:C0444868

[Term]
id: HP:0000002
name: Abnormality of body height
def: "Deviation from the norm of height with respect to that which is expected according to age and gender norms." [HPO:probinson]
synonym: "Abnormality of body height" EXACT layperson []
xref: UMLS:C4025901
is_a: HP:0001507 ! Growth abnormality
created_by: peter
creation_date: 2008-02-27T02:20:00Z

[Term]
id: HP:0000003
name: Multicystic kidney dysplasia
alt_id: HP:0004715
def: "Multicystic dysplasia of the kidney is characterized by multiple cysts of varying size in the kidney and the absence of a normal pelvicaliceal
system. The condition is associated with ureteral or ureteropelvic atresia, and the affected kidney is nonfunctional." [HPO:curators]
comment: Multicystic kidney dysplasia is the result of abnormal fetal renal development in which the affected kidney is replaced by multiple cysts and
has little or no residual function. The vast majority of multicystic kidneys are unilateral. Multicystic kidney can be diagnosed on prenatal ultrasound.
synonym: "Multicystic dysplastic kidney" EXACT []
synonym: "Multicystic kidneys" EXACT []
synonym: "Multicystic renal dysplasia" EXACT []
xref: MSH:D021782
xref: SNOMEDCT_US:204962002
xref: SNOMEDCT_US:82525005
xref: UMLS:C3714581
is_a: HP:0000107 ! Renal cyst
```

```
#Format: entrez-gene-id<tab>entrez-gene-symbol<tab>HPO-Term-Name<tab>HPO-Term-ID
8192    CLPP    Short stature   HP:0004322
8192    CLPP    Seizures        HP:0001250
8192    CLPP    Primary amenorrhea      HP:0000786
8192    CLPP    Autosomal recessive inheritance HP:0000007
8192    CLPP    Microcephaly    HP:0000252
8192    CLPP    Hypoplasia of the uterus        HP:0000013
8192    CLPP    Hypergonadotropic hypogonadism  HP:0000815
8192    CLPP    Congenital sensorineural hearing impairment     HP:0008527
2       A2M     Autosomal dominant inheritance  HP:0000006
8195    MKKS    Macrocephaly    HP:0000256
8195    MKKS    Abnormal electroretinogram      HP:0000512
8195    MKKS    Skeletal muscle atrophy HP:0003202
8195    MKKS    Multicystic kidney dysplasia    HP:0000003
8195    MKKS    Brachydactyly   HP:0001156
8195    MKKS    Cataract        HP:0000518
8195    MKKS    Syndactyly      HP:0001159
8195    MKKS    Autosomal recessive inheritance HP:0000007
8195    MKKS    Hypogonadism    HP:0000135
8195    MKKS    Abnormality of the ovary        HP:0000137
8195    MKKS    Postaxial hand polydactyly      HP:0001162
```

```
#Format: HPO-ID<tab>HPO-Name<tab>Gene-ID<tab>Gene-Name
HP:0001459      1-3 toe syndactyly      2737    GLI3
HP:0006088      1-5 finger complete cutaneous syndactyly        64327   LMBR1
HP:0010708      1-5 finger syndactyly   6469    SHH
HP:0010708      1-5 finger syndactyly   64327   LMBR1
HP:0010713      1-5 toe syndactyly      2737    GLI3
HP:0000878      11 pairs of ribs        545     ATR
HP:0000878      11 pairs of ribs        6497    SKI
HP:0000878      11 pairs of ribs        6657    SOX2
HP:0000878      11 pairs of ribs        8514    KCNAB2
HP:0000878      11 pairs of ribs        2563    GABRD
HP:0000878      11 pairs of ribs        6628    SNRPB
HP:0000878      11 pairs of ribs        6662    SOX9
HP:0000878      11 pairs of ribs        63976   PRDM16
HP:0000878      11 pairs of ribs        126792  B3GALT6
HP:0000878      11 pairs of ribs        650     BMP2
HP:0000878      11 pairs of ribs        2317    FLNB
HP:0000878      11 pairs of ribs        6223    RPS19
HP:0000878      11 pairs of ribs        22995   CEP152
HP:0000878      11 pairs of ribs        26229   B3GAT3
HP:0000878      11 pairs of ribs        473     RERE
HP:0000878      11 pairs of ribs        3930    LBR
HP:0000878      11 pairs of ribs        100151683       RNU4ATAC
HP:0000878      11 pairs of ribs        55835   CENPJ
HP:0000878      11 pairs of ribs        9469    CHST3
HP:0000878      11 pairs of ribs        10013   HDAC6
HP:0000878      11 pairs of ribs        2879    GPX4
HP:0001233      2-3 finger syndactyly   51057   WDPCP
HP:0001233      2-3 finger syndactyly   50964   SOST
```



Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw

```
/**
 * Private constructor
 */
private HPOTermNodeFacade() {
    this.allHPOTermNodes = new HashMap<String, HPOTermNode>();
    this.allGenes = new HashMap<String, Gene>();
    this.initialize();
}

/**
 * Get instance method
 */
public static HPOTermNodeFacade getInstance() {

/**
 * Get all node ids
 */
public Set<String> getAllHPONodeIds() {

/**
 * Get root node
 */
public HPOTermNode getRootNode() {
    return this.getHPOTermNode(ROOT_NODE_ID);
}
/**
 * Get HPO Term
 *
 * @param id
 * @return
 */
public HPOTermNode getHPOTermNode(String id) {
    return this.allHPOTermNodes.get(id);
}
```

2 Analysis Code

Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw

```
 * @author Shray Alag
 * @version 11/21/18
 */
public class ClinicalTrialIndexCrawler implements ClinicalTrialsConstants {
    private BufferedWriter outputBufferedWriter = null;
    private int counter = 0;

    public ClinicalTrialIndexCrawler(String outputFileName) {
        outputBufferedWriter = FileUtil.getBufferedWriter(outputFileName);
    }

    /**
     * Recursively processes the files in the directory and calls the subdirectory
     * @param topDir
     */
    public void crawl(String topDir) {

        File f = new File(topDir);
        this.counter = 0;
        this.crawl(f);
        this.cleanup();
    }

    /**
     * Recursively processes the files in the directory and calls the subdirectory
     * @param topFile
     */
    private void crawl(File topFile) {
        if (topFile.isDirectory()) {
            File[] folderFiles = topFile.listFiles();
            for (File file : folderFiles) {
                this.counter++;
                System.out.println(this.counter + " " + file.getName());
                analyzeFile(file);
                crawl(file);
            }
        }
    }

    /**
     * Callback method for each file that is found
     * @param f
     */
    protected void analyzeFile(File f) {
        if (!f.isDirectory()) {
            String fileName = f.getAbsolutePath();
            ClinicalTrial ct = new ClinicalTrial();
            ClinicalTrialHandler handler = new ClinicalTrialHandler(ct);
            try {
                File inputFile = new File(fileName);
```

Crawler code

```
 */
protected void analyzeFile(File f) {
    if (!f.isDirectory()) {
        String fileName = f.getAbsolutePath();
        ClinicalTrial ct = new ClinicalTrial();
        ClinicalTrialHandler handler = new ClinicalTrialHandler(ct);
        try {
            File inputFile = new File(fileName);
            SAXParserFactory factory = SAXParserFactory.newInstance();
            SAXParser saxParser = factory.newSAXParser();
            saxParser.parse(inputFile, handler);
        } catch (Exception e) {
            e.printStackTrace();
        }

        Map<String, String> genesMap = new HashMap<String, String>();
        MeSHToHPOFacade meshFacade = MeSHToHPOFacade.getInstance();
        MeshIdFacade meshIdFacade = MeshIdFacade.getInstance();
        List<String> meshTerms = ct.getMeshTerms();
        if (meshTerms != null) {
            this.numCTMesh++;
            write("CT:" + ct.getClinicalTrialId() + " " + ct.getMeshTerms().size());
            for (String meshTerm : meshTerms) {
                String meshId = meshIdFacade.getId(meshTerm);
                if (meshId != null) {
                    List<HPOTermNode> hpoNodes = meshFacade.getHPONodeForMeshId(meshId);
                    if ((hpoNodes != null) && (hpoNodes.size() > 0)) {
                        HPOTermNode node = hpoNodes.get(0);
                        int numGenes = 0;
                        if (node.getAssociatedGenes() != null) {
                            numGenes = node.getAssociatedGenes().size();
                        }
                        write("MeSH:" + meshTerm + "\tNode:" + node.getName() + " Id:" + node.getId() + "\tGene:"
                                + numGenes);
                        List<String> ctList = this.nodeToCTsMap.get(node.getId());
                        if (ctList == null) {
                            ctList = new ArrayList<String>();
                            this.nodeToCTsMap.put(node.getId(), ctList);
                        }
                        ctList.add(ct.getClinicalTrialId());
                        List<Gene> genes = node.getAssociatedGenes();
                        if (genes != null) {
                            for (Gene gene : genes) {
                                genesMap.put(gene.getName(), gene.getName());
                            }
                        }
                    } else {
                        write("MeSH:" + meshTerm + "\tNo Match");
                    }
                }
            }
```

HPO Mapper example

Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw

```
     * @param text
     * @return
     */
    public Map<String, List<String>> extractSNPs(String text) {
        int g = text.indexOf("、");
        while (g > -1) {
            text = text.substring(0, g) + ", " + text.substring(g + 1, text.length());
            g = text.indexOf("、");
        }

        Map<String, List<String>> snpResultsMap = new HashMap<String, List<String>>();
        if (text != null && text.length() > 0) {
            // Split the text into sentences
            OpenNLPFacade facade = new OpenNLPFacade();
            String[] sentences = facade.getSentences(text);
            for (String sentence : sentences) {
                // Get the list of takens in the sentence
                String[] tokens = facade.getTokens(sentence);
                for (String token : tokens) {
                    // Check to see if this token is a SNP
                    if (isPatternMatch(token)) {

                        String cleanSNP = cleanMatchedString(token);
                        String matchedString = getMatchedString(token);
                        if (matchedString.length() > 0) {
                            sentence += " --- " + matchedString + " --- ";
                        }
                        List<String> snipets = snpResultsMap.get(cleanSNP);
                        if (snipets == null) {
                            snipets = new ArrayList<String>();
                            snpResultsMap.put(cleanSNP, snipets);
                        }
                        snipets.add(sentence);
                    }
                }
            }
        }

        return snpResultsMap;
    }
```

SNP miner code

Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw

```java
     *
     * @param rawText
     * @return
     * @throws Exception
     */
    public Map<Mutation, Set<int[]>> extractMutations(String rawText) throws Exception {
        return this.mutationFinder.extractMutations(rawText);
    }

    public static void main(String[] args) throws Exception {
        String text = "(c.1517A>G) of exon 13, resulting in the substitution of"
                + " aspartic acid with glycine in codon 506 (p.Asp506Gly)."
                + " Ser42Thr and Trp36Tyr. The A42G mutation was made."
                + "The Ala42-->Gly mutation was made."
                + "The Ala42Gly mutation was made.";
        MutationFinderFacade f = MutationFinderFacade.getInstance();
        Map<Mutation, Set<int[]>>  mutations = f.extractMutations(text);
        int count = 1;
        for (Mutation mutation: mutations.keySet()) {
            Set<int[]> positions = mutations.get(mutation);
            System.out.println(count++ + " " + mutation);
            for (int[] position: positions) {
                System.out.println("\t[" + position[0] + "," + position[1] + "] - " +
                        text.substring(position[0], position[1]));
            }
        }
    }

}
```

```
Problems  Javadoc  Declaration  Console  Debug
<terminated> MutationFinderFacade [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java (Apr 18, 2020, 12:34:32 PM)
Loading regex's: 100
Loading regex's: 200
Loading regex's: 300
Loading regex's: 400
Loading regex's: 500
Loading regex's: 600
Loading regex's: 700
Completed loading of regular expressions: 759 loaded.
1 S42T
        [112,120] - Ser42Thr
2 W36Y
        [125,133] - Trp36Tyr
3 A42G
        [139,143] - A42G
        [200,208] - Ala42Gly
        [166,177] - Ala42-->Gly
```

MutationFinder facade and example code

Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw

```java
        return MutationFinder_PATTERN_CT_LIST_INDEX;
    }

    public static void main(String[] args) {
        MutationFinderCtMapperFacade facade = MutationFinderCtMapperFacade.getInstance();
        String[] snps = { "L858R", "R131H", "H1112L" };
        String[] ctIds = { "NCT00939770", "NCT02786342", "NCT02543216" };
        for (String snp : snps) {
            List<String> cts = facade.getAssociatedClinicalTrialIds(snp);
            System.out.println(snp + " ... " + cts.size());
            for (String ct : cts) {
                System.out.println("\t" + ct);
            }
        }
        for (String ct : ctIds) {
            List<String> rsNumbers = facade.getAssocaitedRsNumbers(ct);
            System.out.println(ct + " ... " + rsNumbers.size());
            for (String rs : rsNumbers) {
                System.out.println("\t" + rs);
            }
        }

        //Analytics
        List<SnpClinicalTrials> cts = facade.getSortedClinicalTrialsSnp();
        for (int i = 0; i < 12; i ++) {
            SnpClinicalTrials ct = cts.get(i);
            List<String> ids = ct.getClinicalTrialsId();
            System.out.println(i + " " + ct.getRsNumber() + "  " + ids.size() + " " );
        }
    }
```

```
Problems  Javadoc  Declaration  Console  Debug
<terminated> MutationFinderCtMapperFacade [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java (Apr 18, 2020, 12:47:47 PM)
        V1110L
        V1206L
        V1238I
        V941L
        Y1248C
        Y1248D
        Y1248H
        Y1253D
NCT02786342 ... 0
NCT02543216 ... 0
0 NCT00511368  37
1 NCT00460382  36
2 NCT03783286  33
3 NCT04172597  32
4 NCT02404233  29
5 NCT00959894  28
6 NCT00706147  28
7 NCT00121017  28
8 NCT01685801  27
9 NCT01736761  26
10 NCT02673398  23
11 NCT02499978  22
```

MutationFinder clinical trials analysis

```java
    */
    public static final void main(String[] args) {
        HPOTermNodeFacade facade = HPOTermNodeFacade.getInstance();
        System.out.println(facade.getNumberOfNodes());
        String[] ids = { "HP:0000013", "HP:0000003", "HP:0000020" };
        for (String id : ids) {
            System.out.println(facade.getHPOTermNode(id));
        }
        String [] genes = {"CLPP", "BRCA1", "ESR1"//, "RNF1"
            };
        for (String gene: genes) {
            System.out.println(facade.getGene(gene));
        }
        Collection<String> mappedNodes =  facade.getMappedHPONodesWithGenes() ;
        System.out.println("Number of  phenotype mapped to genes = " +
                mappedNodes.size());
        System.out.println("Number of Gene phenotype assoc = " +
                facade.getNumGenePhenotypeAssociations());
        System.out.println("Number of HPO Nodes = " + facade.getNumberOfNodes());
        System.out.println("Number of Unique Genes = " + facade.getGeneMap().keySet().size());
        System.out.println("Number parent-child relationships = " + facade.getNumberOfParentChildRelationships());
        System.out.println("Number of nodes with genes: " + facade.getNumberHPOTermsWithGenes());
    }
```

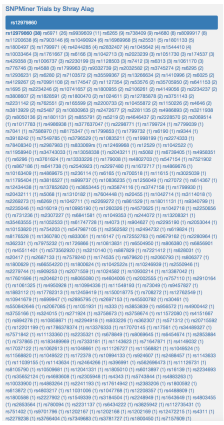
```
Problems  Javadoc  Declaration  Console  Debug
<terminated> HPOTermNodeFacade [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java (Apr 18, 2020, 12:49:47 PM)
        Node:HP:0012531, Pain
        Node:HP:0012638, Abnormality of nervous system physiology
        Node:HP:0012639, Abnormality of nervous system morphology
        Node:HP:0012647, Abnormal inflammatory response
        Node:HP:0012647, Abnormal inflammatory response
        Node:HP:0012649, Increased inflammatory response
        Node:HP:0012649, Increased inflammatory response
        Node:HP:0030012, Abnormal female reproductive system physiology
        Node:HP:0030012, Abnormal female reproductive system physiology
        Node:HP:0030338, Abnormal circulating gonadotropin level
        Node:HP:0030680, Abnormality of cardiovascular system morphology
        Node:HP:0031065, Abnormal ovarian morphology
        Node:HP:0031093, Abnormal breast morphology
        Node:HP:0031093, Abnormal breast morphology
        Node:HP:0032148, Episodic pain

Number of  phenotype mapped to genes = 8947
Number of Gene phenotype assoc = 820296
Number of HPO Nodes = 14961
Number of Unique Genes = 4312
Number parent-child relationships = 18547
Number of nodes with genes: 8947
```

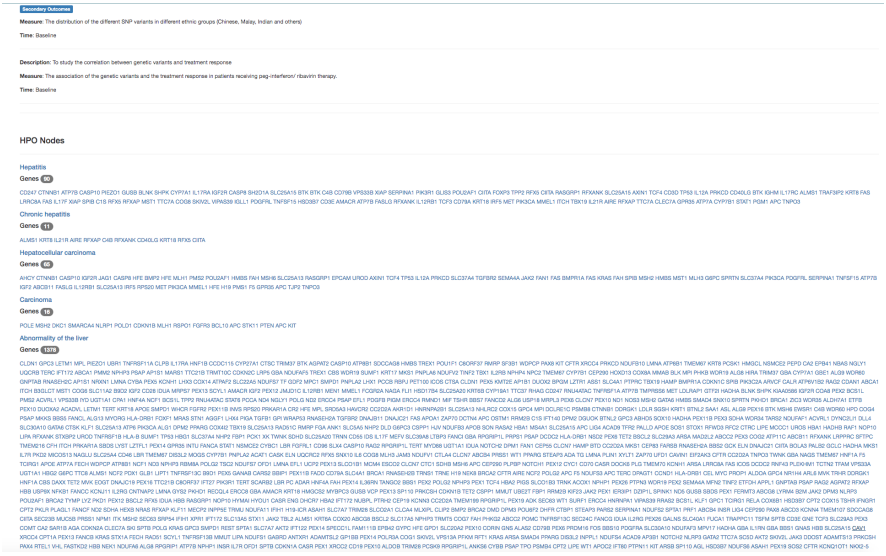
HPO analysis

Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw

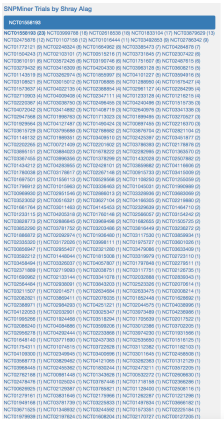
3 Reports



Reports for SNPs

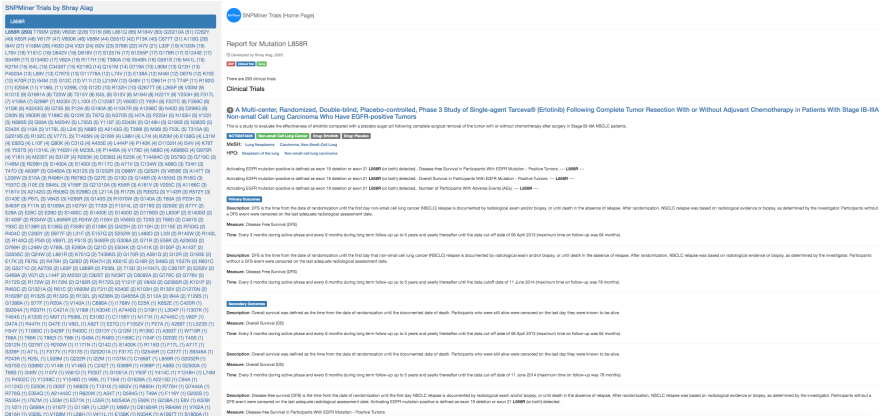


Bottom of report

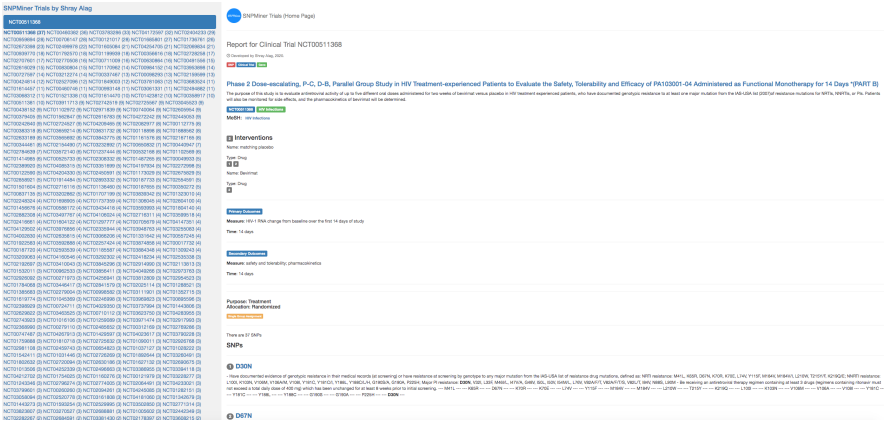


Report for clinical trial

Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw



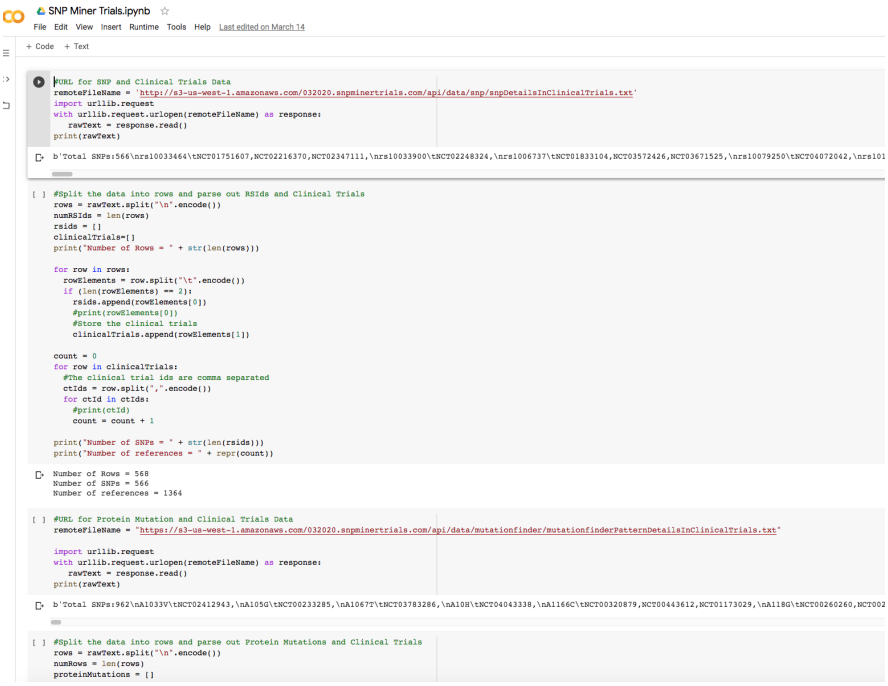
Report for protein mutations



Report for clinical trial

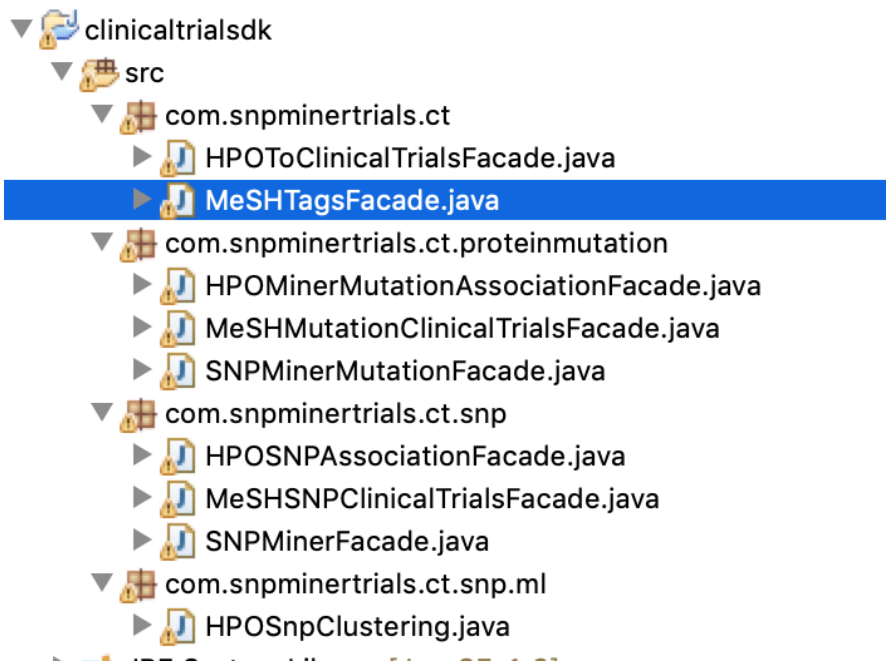
4 APIs and Toolkits

Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw



Google Colab

SDK



Available Java APIs to access the data

Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw

```
140      */
141     public static void main(String[] args) {
142         // Get the instance
143         MeSHTagsFacade facade = MeSHTagsFacade.getInstance();
144         // Get the ids
145         List<String> meshIds = facade.getAllMeSHIds();
146         List<String> terms = facade.getAllMeSHTerms();
147         System.out.println("Number MeSH terms: " + terms.size());
148         System.out.println("Number MeSH ids: " + meshIds.size());
149 
150         System.out.println();
151         for (int i = 0; i < 5; i++) {
152             String meshId = meshIds.get(i);
153             List<String> names = facade.getMeSHTermsForId(meshId);
154 
155             System.out.println("Terms for " + meshId +  " ... " + names.size()
156             System.out.print("\t");
157             for (String s : names) {
158                 System.out.print(s + ",");
159             }
160             System.out.println();
161         }
162 
163         System.out.println();
```

```
Console
<terminated> MeSHTagsFacade (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java (Apr 18, 2020, 12:58:42 PM)
MeSH id for ACTH-Secreting Pituitary Adenoma ... D049913
MeSH id for AIDS Dementia Complex ... D015526
MeSH terms for D013543 ... 1
        Sweat Gland Diseases,
MeSH terms for D013611 ... 1
        Tachycardia, Atrioventricular Nodal Reentry,
```

Ref: https://s3-us-west-1.amazonaws.com/032020.snpminertrials.com/api/code/clinicaltrialsdk/src/com/snpminertrials/ct/MeSHTagsFacade.java

```
155     public static void main(String[] args) {
156         // Get the instance
157         HPOToClinicalTrialsFacade facade = HPOToClinicalTrialsFacade.getInstance();
158         // Get the ids
159         List<String> hpoIds = facade.getHPOIds();
160         List<String> clinialTrials = facade.getClinicalTrialIds();
161         System.out.println("Number Clinical Trials: " + clinialTrials.size());
162         System.out.println("Number HPO terms: " + hpoIds.size());
163 
164         System.out.println();
165         for (int i = 0; i < 5; i++) {
166             String hpoId = hpoIds.get(i);
167             String name = facade.getHPONodeName(hpoId);
168             Collection<String> mutationIdsL = facade.getClinicalTrialsForHPOTerm(hpoI
169             System.out.println("Clincial Trials for " + hpoId + " " +
170                     name + " ... " + mutationIdsL.size());
171             System.out.print("\t");
172             for (String s : mutationIdsL) {
173                 System.out.print(s + ",");
174             }
175             System.out.println();
176         }
177 
```

```
Console
<terminated> HPOToClinicalTrialsFacade (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java (Apr 18, 2020, 1:03:02 PM)
        HP:0001627,
HPO terms for NCT00000108 ... 1
        HP:0001626,
HPO terms for NCT00000112 ... 1
        HP:0000956,
HPO terms for NCT00000113 ... 1
        HP:0000545,
HPO for NCT00444431 ... 3
        HP:0000003,HP:0000126,HP:0006000,
HPO for NCT00175123 ... 2
        HP:0000011,HP:0002475,
```

HPOToClinicalTrailsFacade

Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw

```
    public static void main(String[] args) {
        // Get the instance
        HPOMinerMutationAssociationFacade facade = HPOMinerMutationAssociationFacade.getInstance();
        // Get the ids
        List<String> hpoIds = facade.getHPOIds();
        List<String> mutationIds = facade.getMutationIds();
        System.out.println("Number Protein Mutations: " + mutationIds.size());
        System.out.println("Number HPO terms: " + hpoIds.size());

        System.out.println();
        for (int i = 0; i < 5; i++) {
            String hpoId = hpoIds.get(i);
            String name = facade.getHPONodeName(hpoId);
            Collection<String> mutationIdsL = facade.getMutationsForHPOTerm(hpoId);
            System.out.println("Protein Mutations for " + hpoId + " " + name + " ... " + mutationIdsL
            System.out.print("\t");
            for (String s : mutationIdsL) {
                System.out.print(s + ",");
            }
            System.out.println();
        }
```

```
Console
<terminated> HPOMinerMutationAssociationFacade (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java (Apr 18, 2020, 1:04:55 PM)
        HP:0002088,
HPO terms for A1166C ... 3
        HP:0000822,HP:0001658,HP:0012592,
HPO terms for A118G ... 10
        HP:0000822,HP:0002088,HP:0012393,HP:0006510,HP:0003418,HP:0003419,HP:0006536,HI
HPO terms for A119S ... 2
        HP:0000501,HP:0001087,
HPO for L858R ... 21
        HP:0002664,HP:0003002,HP:0030731,HP:0100526,HP:0030358,HP:0012174,HP:0100834,HI
HPO for H1047R ... 5
        HP:0002664,HP:0003002,HP:0030731,HP:0030153,HP:0100764,
```

HPO and protein mutations

```
    public static void main(String[] args) {
        // Get the instance
        SNPMinerMutationFacade facade = SNPMinerMutationFacade.getInstance();
        // Get the ids
        List<String> clinicalTrials = facade.getClinicalTrialIds();
        List<String> mutations = facade.getMutationIds();
        System.out.println("Number Mutations: " + mutations.size());
        System.out.println("Number Clincal Trials: " + clinicalTrials.size());

        System.out.println();
        for (int i = 0; i < 5; i++) {
            String ctid = clinicalTrials.get(i);
            Collection<String> mutationIds = facade.getProteinMutationsForClinical
            System.out.println("Mutations for " + ctid + " ... " + mutationIds.siz
            System.out.print("\t");
            for (String mutationId : mutationIds) {
                System.out.print(mutationId + ",");
            }
            System.out.println();
        }

        System.out.println();
        for (int i = 0; i < 5; i++) {
```

```
Console
<terminated> SNPMinerMutationFacade (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java (Apr 18, 2020, 1:06:15 PM)
        NCT00233285,
Clinical Trial for A1067T ... 1
        NCT03783286,
Clinical Trial for A10H ... 1
        NCT04043338,
Clinical Trial for A1166C ... 3
        NCT00320879,NCT00443612,NCT01173029,
Clinical Trial for L858R ... 293
        NCT00373425,NCT00381654,NCT00503971,NCT00567359,NCT00577707,NCT00752076,NCT008
Clinical Trial for M184V ... 60
        NCT00000838,NCT00040157,NCT00098293,NCT00106379,NCT00116636,NCT00121979,NCT001
```

Clinical trials for protein mutations

Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw

```
154      */
155     public static void main(String[] args) {
156         // Get the instance
157         HPOSNPAssociationFacade facade = HPOSNPAssociationFacade.getInstan
158         // Get the ids
159         List<String> hpoIds = facade.getHPOIds();
160         List<String> snpIds = facade.getSNPIds();
161         System.out.println("Number SNPs: " + snpIds.size());
162         System.out.println("Number HPO terms: " + hpoIds.size());
163 
164         System.out.println();
165         for (int i = 0; i < 5; i++) {
166             String hpoId = hpoIds.get(i);
167             String name = facade.getHPONodeName(hpoId);
168             Collection<String> snpIdsL = facade.getSNPsForHPOTerm(hpoId);
169             System.out.println("SNP for " + hpoId + " " + name + " ... " +
170             System.out.print("\t");
171             for (String s : snpIdsL) {
172                 System.out.print(s + ",");
173             }
174             System.out.println();
175         }
176 
```

```
Console
<terminated> HPOSNPAssociationFacade (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java (Apr 18, 2020, 1:07:14 PM)
        HP:0100753,HP:0100754,HP:0000709,
HPO terms for rs10079250 ... 1
        HP:0100242,
HPO terms for rs1011970 ... 1
        HP:0003002,
HPO terms for rs10153820 ... 1
        HP:0001677,
HPO for rs2246704 ... 2
        HP:0001370,HP:0001369,
HPO for rs10163409 ... 1
        HP:0001513,
```

HPO and snps

Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw

```
163    public static void main(String[] args) {
164        // Get the instance
165        SNPMinerFacade facade = SNPMinerFacade.getInstance();
166        // Get the ids
167        List<String> clinicalTrials = facade.getClinicalTrialIds();
168        List<String> snps = facade.getSNPIds();
169        System.out.println("Number SNPS: " + snps.size());
170        System.out.println("Number Clincal Trials: " + clinicalTrials.size())
171
172        System.out.println();
173        for (int i = 0; i < 5; i++) {
174            String ctid = clinicalTrials.get(i);
175            Collection<String> snpIds = facade.getSNPsForClinicalTrial(ctid);
176            System.out.println("SNPs for " + ctid + " ... " + snpIds.size());
177            System.out.print("\t");
178            for (String snpId : snpIds) {
179                System.out.print(snpId + ",");
180            }
181            System.out.println();
182        }
183
184        System.out.println();
```

Console

<terminated> SNPMinerFacade (2) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java (Apr 18, 2020, 1:08:09 PM)

```
Clinical Trial for rs10033464 ... 3
        NCT01751607,NCT02216370,NCT02347111,
Clinical Trial for rs10033900 ... 1
        NCT02248324,
Clinical Trial for rs1006737 ... 3
        NCT01833104,NCT03572426,NCT03671525,
Clinical Trial for rs10079250 ... 1
        NCT04072042,
Clinical Trial for rs1011970 ... 1
        NCT02618538,
```

Clinical trials and RSids

Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw

```
187    public static void main(String[] args) {
188        HPOSnpClustering f = HPOSnpClustering.getInstance();
189        String[] hpoIds = { "HP:0003002","HP:0001909","HP:0002511","HP:001211
190        for (String hpoId : hpoIds) {
191            String name = f.getHPOName(hpoId);
192
193            List<Node> nodes = f.getClosestHPONodes(hpoId);
194            System.out.println(hpoId + " " + name + " ... " + nodes.size());
195            for (int i = 0; i < nodes.size(); i++) {
196                System.out.println("\t" + i + " " + nodes.get(i));
197            }
198        }
199        Set<String> allHPOids = f.getAllHPOIds();
200        for (String hpoId : allHPOids) {
201            String name = f.getHPOName(hpoId);
202
203            List<Node> nodes = f.getClosestHPONodes(hpoId);
204            System.out.println(hpoId + " " + name + " ... " + nodes.size());
205            for (int i = 0; i < nodes.size(); i++) {
206                System.out.println("\t" + i + " " + nodes.get(i));
207            }
208        }
```

Console

<terminated> HPOSnpClustering [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/bin/java (Apr 18, 2020, 1:08:59 PM)

```
        2 HP:0002664 Neoplasm 0.24999999999999994
[D@3bfdc050
HP:0001402 Hepatocellular carcinoma ... 4
        0 HP:0030731 Carcinoma 0.3721042037676254
        1 HP:0200123 Chronic hepatitis 0.16012815380508716
        2 HP:0001392 Abnormality of the liver 0.09805806756909201
        3 HP:0012115 Hepatitis 0.06201736729460423
[D@1bce4f0a
HP:0001369 Arthritis ... 2
        0 HP:0001370 Rheumatoid arthritis 0.9999999999999999
        1 HP:0002206 Pulmonary fibrosis 0.31622776601683794
```

Machine learning example using meta-data

Citation: Shray Alag (05/28/2020). Unique insights from ClinicalTrials.gov by mining protein mutations and RSids in addition to applying the Human Phenotype Ontology. https://dx.doi.org/10.17504/protocols.io.bfacjiaw