**Note:** This documents aim to be used as a guide to support the development of lrgext.py, to those intending to modify the script in a future and those looking for a reference of the steps and features to be accomplished during script development in a collaborative environment.

# PROGRAM DEVELOPMENT

### /Need still to include all the things we have done that are included in the script. The final doc. Will contain just the things done and some outlook/

### Initial framework and structure: IGP, Done

### Features:

1. Testing /initial suggested tests/ :
   1. LRG file exists and is a readable file: IGP, Done
   2. Check that the xml version format (schema) is right: IGP, Done
   3. Strand test, check direction of strand: IGP, Done
   4. Check if the given gene has an allocated LRG ID: VF, IP
   5. Build test: VF, IP
      1. Build\_number: check the number of build provided. If more than one, check that coordinates are diff. VF, IP
      2. Build\_coor: check that the start and end of the coordinate for each build are different. VF, IP

/other tests proposed/

* 1. Data\_consistency: check that coordinates extracted are numbers: VF, IP
  2. Exon\_coor: check that exons do not overlap: IGP, IP
  3. Exon\_number: check that we get the right number of exons (assert) : IGP, IP

1. Usage:
   1. Extraction of input and outputs from different folders: IGP, Done
   2. Input LRG file from command line: IGP, Done
2. Parsing:
   1. Extract
   2. Extracting, exon, transcript and protein coordinates per exon: IGP, Done
   3. Extract gene name: VF, Done
   4. Compare builds: VF, IP
   5. Identify if variant changes between builds occur in introns or exons: VF, IP
3. File production:
   1. Create tab separated file: IGP, Done
   2. Create cvs file: IGP, Done
   3. Create bed file: IGP, Done
   4. Create build mapper file: VF, IP

### Disclaimer: IGP, Done

### Dealing with issues and bugs

1. Issue with LRG without protein coordinates in the first exon (e.g. LRG\_62 and LRG\_292): IGP, Done

* Commenting:

1. Initial commenting: IGP, Done
2. Update and revision: VF, IP

# Documentation:

### 1. COPYRIGHT:

1. Policy checking (University and NHS): IGP, Done
2. Structure and first drafting: IGP, Done
3. Completion: IGP, Done
4. Read proofing: VF, IP

### 2. README (IGP + VF)

1. Document structure: IGP, Done

2. Initial draft: IGP, Done

3. Content:

1. Global Background: VF, IP

2. Usage, installation: IGP, IP (pending on the addition of features)

4. Features description:

* + Background to group of features: Testing, Parsing & File Generation /VF/
  + Description of features /IGP + VF as required/

0. initial tests: IGP + VF (as added by author)

1. get\_structure(data) : VF

2. get\_background(root) : VF

3. get\_build\_info(up\_anno): VF

4. get\_up\_anno(data) : VF

5. get\_exon\_data(data) : IGP

6. output2file(): IGP

7. disclaimer(): IGP

8. final\_tests(): IGP + VF (as added by author)

1. Versioning: IGP, Done
2. Disclaimer: IGP, done
3. Final editing and read-proofing: VF, pending to the completion of doc

# OUTLOOK

1. Adding new test using 'assert': VF

2. Creating the name of the file automatically: IGP , IP

3. Including all in the main function (important when program is going to be used as a package): IGP

4. Access file from internet: IGP, deemed as not necessary for now, since LRG files will be downloaded/updated at periodic times to avoid delays due to internet connexion