CNV GDRC Scripts

Local Environment

We recommend having four `conda` environments on your local machine to run our analysis. These environments are stored in the `envs/` directory. The following commands will create environments called `gdrc_data`, `gdrc_probprog`, `gdrc_snakemake`, and `gdrc_dx` using `conda`.

```
conda env create --file envs/data.yaml --name gdrc_data
conda env create --file envs/probprog.yaml --name gdrc_probprog
conda env create --file envs/snakemake.yaml --name gdrc_snakemake
conda env create --file envs/dx.yaml --name gdrc_dx
```

Analysis of Individual-Level Data

Analysis of the individual-level data requires access to the UK Biobank (UKB). Successful application to the data will provide access to the UKB research analysis platform (RAP).

Some scripts will need to be run on the RAP using the JupyterLab tool. The rest can be executed using the command line on your local computer. These scripts will spawn jobs on the RAP. Note that memory storage and computation on the RAP incurs expenses. We estimate that the total expenses to perform our analysis is less than £500.

The `dx` command line utility is useful for interacting with the UKB RAP. It can be used to upload/download files and execute jobs. It will be installed in the `gdrc_dx` environment but needs to be configured by each user.

For scripts that are executed locally using the `dx` command line utility provided by DNAnexus, there is a `project` variable defined at the top of the script. This needs to be modified to use your UKB RAP project string. This will look something like this and can be found on the UKB RAP project information pane.

```
project="project-[24 character hash]"
```

Some of our scripts require software that is not available on the UKB RAP. We use Docker containers to package such software and execute it.

1. FlashPCA2
    1. Use `docker build FlashPCA2/ -t 'flashpca2'` to build the container (on Apple silicon, the command is `docker buildx build --platform=linux/amd64 FlashPCA2 -t 'flashpca2'`).
    2. Use `docker save flashpca2 | gzip > flashpca2.tar.gz` to save the image.
    3. Upload the image to the UKB RAP at `/Software/`.
2. VEP with loftee
    1. Upload data required by loftee to the UKB RAP at `/Annotations/LoFs/`.
    2. Pull the VEP docker container using `docker pull ensemblorg/ensembl-vep`
    3. Use `docker save ensembl-vep | gzip > loftee.tar.gz` to save the image.
    4. Upload the image to the UKB RAP at `/Software/`.

Preparing for Association Analysis

Scripts should be executed in the following order.

Order	Script	Environment
1	`field_names.ipynb`	Local (`gdrc_data`)
2	`extract_covariates_and_phenotypes.ipynb`	UKB RAP
3	`extract_cnv_burden_genotypes.sh`	Local (`gdrc_dx`)*
4	`combine_cnv_burden_genotypes.sh`	Local (`gdrc_dx`)
5	`individuals_for_analysis.ipynb`	UKB RAP
6	`filter_exome_rare_variants_for_PCs.sh`	Local (`gdrc_dx`)

Order	Script	Environment
7	`estimate_fine_structure_PCs.sh`	Local (`gdrc_dx`)
8	`filter_genotypes_for_wgr.sh`	Local (`gdrc_dx`)
9	`process_covariates_and_phenotypes.ipynb`	UKB RAP
10	`combine_nmr_phases.ipynb`	Local (`gdrc_dx`)*
11	`wes_allele_frequencies.sh`	Local (`gdrc_dx`)
12	`lof_loftee.sh`	Local (`gdrc_dx`)
13	`process_loftee_annotation.sh`	Local (`gdrc_dx`)*
14	`pick_most_severe_loftee_consequence.sh`	Local (`gdrc_dx`)*
15	`extract_cnv_burden_cov_matrices.sh`	Local (`gdrc_dx`)*
16	`create_annotation_files_lof.ipynb`	UKB RAP
17	`create_annotation_files_cnv.ipynb`	UKB RAP
18	`ecdf_cnv_burden_length.ipynb`	Local (`gdrc_data`)

*Specific instructions below the table.

`combine_nmr_phases.ipynb` : This requires early access to the Phase 3 data from Nightingale. Please contact Nightingale PLC directly for requests to the data. Phase 3 data will eventually join the Phase 1 and Phase 2 data in the UKB.

`extract_cnv_burden_genotypes.sh` :

1. Upload `extract_cnv_burden_genotypes.py` to `/Genotypes/CNVs/` in the UKB RAP.
2. Run `extract_cnv_burden_genotypes.sh` on the local machine in the `gdrc_dx` environment.

`process_loftee_annotation.sh` :

1. Upload `process_loftee_annotation.py` to `/Annotations/LoFs/` in the UKB RAP.
2. Run `process_loftee_annotation.sh` on the local machine in the `gdrc_dx` environment.

`pick_most_severe_loftee_consequence.sh` :

1. Upload `pick_most_severe_loftee_consequence.py` to `/Annotations/LoFs/` in the UKB RAP.
2. Run `pick_most_severe_loftee_consequence.sh` on the local machine in the `gdrc_dx` environment.

`extract_cnv_burden_cov_matrices.sh` :

1. Upload `extract_cnv_burden_cov_matrices.py` to `/Genotypes/CNVs/` in the UKB RAP.
2. Run `extract_cnv_burden_cov_matrices.sh` on the local machine in the `gdrc_dx` environment.

Association Analysis

Scripts should be executed in the following order.

Order	Script	Environment
1	`regenie_step_1_lofs.sh`	Local (`gdrc_data`)
2	`regenie_step_1_lofs_nmr.sh`	Local (`gdrc_data`)
3	`regenie_step_1_lofs_stratification.sh`	Local (`gdrc_data`)
4	`regenie_step_1_cnvs.sh`	Local (`gdrc_data`)
5	`regenie_step_1_cnvs_nmr.sh`	Local (`gdrc_data`)
6	`regenie_step_2_lofs.sh`	Local (`gdrc_data`)
7	`regenie_step_2_lofs_nmr.sh`	Local (`gdrc_data`)
8	`regenie_step_2_lofs_stratification.sh`	Local (`gdrc_data`)
9	`regenie_step_2_cnvs.sh`	Local (`gdrc_data`)
10	`regenie_step_2_cnvs_nmr.sh`	Local (`gdrc_data`)

Simulations

Scripts should be executed in the following order.

Order	Script	Environment
1	`extract_cnv_burden_cov_matrices_for_simulations.sh`	Local (`gdrc_data`)
2	`simulations_fixed.sh`	Local (`gdrc_data`)*
3	`simulations_random.sh`	Local (`gdrc_data`)*

*Specific instructions below the table.

`simulations_fixed.sh` :

1. Upload `simulations_fixed_1.ipynb` and `simulations_fixed_2.ipynb` to `/Simulations/` in the UKB RAP.
2. Run `simulations_fixed.sh` on the local machine in the `gdrc_dx` environment.

`simulations_random.sh` :

1. Upload `simulations_random_1.ipynb` and `simulations_random_2.ipynb` to `/Simulations/` in the UKB RAP.
2. Run `simulations_random.sh` on the local machine in the `gdrc_dx` environment.

Analysis of Summary Statistics

These analyses can be performed on summary statistics. You can either use data that you generate yourself using the scripts above, our summary statistics, or any other burden summary statistics. If you are using other duplication burden summary statistics, you will need to obtain in-sample linkage disequilibrium (LD) estimates for the burden genotypes.

In the rest of the sections, we will assume that you are using our summary statistics. The data deposited in Zenodo is expected to be in `sum_stats/data/` . Here is an example of downloading some of the data.

```
mkdir -p sum_stats/data/
curl -L 'https://zenodo.org/records/13852455/files/LoF_Burden_Tests.tar.gz?download=1' \
    -o sum_stats/data/LoF_Burden_Tests.tar.gz
curl -L 'https://zenodo.org/records/13852455/files/CNV_Burden_Tests.tar.gz?download=1' \
    -o sum_stats/data/CNV_Burden_Tests.tar.gz
tar -xvf sum_stats/data/LoF_Burden_Tests.tar.gz -C sum_stats/data/
tar -xvf sum_stats/data/CNV_Burden_Tests.tar.gz -C sum_stats/data/
rm sum_stats/data/LoF_Burden_Tests.tar.gz
rm sum_stats/data/CNV_Burden_Tests.tar.gz
```

Use the following command to download s_{het} estimates from Zeng et al. 2024.

```
curl -L 'https://zenodo.org/records/7939768/files/s_het_estimates.genebayes.tsv?download=1' \
    -o sum_stats/data/s_het_estimates.genebayes.tsv
```

Monotonicity

We use `snakemake` to distribute the workload across a cluster. We have provided the `Snakefile` to execute the workflow. You will have to configure the execution to your specific cluster or local machine.

A basic command to execute the workflow can be found in `run.sh` , which should be executed in the `gdrc_snakemake` environment. This command can be configured extensively to match your environment.

We also provide a notebook (`monotonicity.ipynb`) that can be executed using the `gdrc_probprog` environment to perform the full analysis for a single trait. This can be used to explore individual traits in depth.

RSS MASH

We use `snakemake` to distribute the workload across a cluster. We have provided the `Snakefile` to execute the workflow. You will have to configure the execution to your specific cluster or local machine.

A basic command to execute the workflow can be found in `run.sh`, which should be executed in the `gdrc_snakemake` environment. This command can be configured extensively to match your environment.

Figures

Jupyter notebooks to generate all figures and tables included in the manuscript can be found in the `sum_stats/` directory. The resulting figures and tables are included in the `results/` directory. The data that is required for each notebook is mentioned in the first cell.

Some notebooks use a Python kernel, while others use an R kernel. The scripts and their environments are described below. The following order of execution is recommended. For example, results from `figure_1_analysis.ipynb` are used in `figure_1.ipynb` to generate plots.

Order	Notebook	Environment
1	`figure_1_analysis.ipynb`	Local (`gdrc_data`), Python
2	`figure_1.ipynb`	Local (`gdrc_data`), R
3	`figure_2_analysis.ipynb`	Local (`gdrc_data`), Python
4	`figure_2.ipynb`	Local (`gdrc_data`), R
5	`figure_3_analysis.ipynb`	Local (`gdrc_probprog`), Python
6	`figure_3.ipynb`	Local (`gdrc_data`), R
7	`figure_4_analysis.ipynb`	Local (`gdrc_data`), Python
8	`figure_4.ipynb`	Local (`gdrc_data`), R
9	`figure_5_analysis.ipynb`	Local (`gdrc_data`), Python
10	`figure_5.ipynb`	Local (`gdrc_data`), R
11	`figure_apx_genome_wide_burden.ipynb`	Local (`gdrc_data`), R
12	`figure_apx_simulations.ipynb`	Local (`gdrc_data`), R
13	`figure_apx_confounding_analysis.ipynb`	Local (`gdrc_data`), Python
14	`figure_apx_confounding.ipynb`	Local (`gdrc_data`), R
15	`figure_apx_duplication_ld.ipynb`	Local (`gdrc_data`), R
16	`table_apx_burden_effects.ipynb`	Local (`gdrc_data`), R
17	`gwas_trumpet_plots.sh`	Shell
18	`figure_apx_gwas_trumpet_plots.ipynb`	Local (`gdrc_data`), R
19	`figure_apx_monotonicity_analysis.ipynb`	Local (`gdrc_probprog`), Python
20	`figure_apx_monotonicity.ipynb`	Local (`gdrc_data`), R
21	`figure_apx_genetic_correlation_analysis.ipynb`	Local (`gdrc_probprog`), Python
22	`figure_apx_genetic_correlation.ipynb`	Local (`gdrc_data`), R

Documentation

To generate the documentation, run the following locally using the `gdrc_data` environment.

```
cd doc/
python compile_doc.py
```