title: "build_peak_EnPACT" author: "Saideep Gona" date: "2024-03-26" format: html: code-fold: true code-summary: "Show the code" execute: freeze: true warning: false

Context

In this notebook, I build linearized Con-EnPACT models for peak data from Aracena et al. The peak data are as follows:

ATAC - Chromatin Accessibility H3K27ac - Histone, Enhancers + TSS H3K27me3 - Repression H3K4me1 - "Active" enhancers H3K4me3 - Enhancer

In [ ]:
```
import os,sys
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import json

import subprocess

import pyliftover
```

In [ ]:
```
# Set up modalities

path_to_config_template = "/beagle3/haky/users/saideep/github_repos/Con-EnPA
path_to_run_template = "/beagle3/haky/users/saideep/github_repos/Con-EnPACT/
path_to_project_dir = "/beagle3/haky/users/saideep/projects/Con_EnPACT/model
path_to_count_table = "/beagle3/haky/users/saideep/projects/aracena_modeling
path_to_annotations = "/beagle3/haky/users/saideep/projects/aracena_modeling

modality_pcs = {
    "H3K27ac":{
        "Flu":1,
        "NI":1
    },
    "H3K27me3":{
        "Flu":2,
        "NI":1
    },
    "H3K4me1":{
        "Flu":4,
        "NI":2
    },
    "H3K4me3":{
        "Flu":2,
        "NI":2
    },
    "ATAC":{
        "Flu":3,
        "NI":3
    }
}
```

Create Peak annotation files

```
In [ ]: def liftover_count_table(count_table, from_build="hg19", to_build="hg38", va

            # Liftover index of dataframe to new build, filter rows if not possible

            lo = pyliftover.LiftOver(from_build, to_build)

            new_regions = []
            new_regions_keep = []
            for region in count_table.index:
                chrom = region.split("_")[0]
                if len(region.split("_")) != 3:
                    print("region length not 3")
                    new_regions_keep.append(False)
                    continue
                if chrom not in valid_chroms:
                    print("chromosome not in valid chroms")
                    new_regions_keep.append(False)
                    continue
                start = int(region.split("_")[1])
                end = int(region.split("_")[2])

                new_start_coords = lo.convert_coordinate(chrom, start)
                if len(new_start_coords) == 0:
                    print("No conversion")
                    new_regions_keep.append(False)
                    continue
                new_start = new_start_coords[0][1]

                new_end_coords = lo.convert_coordinate(chrom, end)
                if len(new_end_coords) == 0:
                    print("No conversion")
                    new_regions_keep.append(False)
                    continue
                new_end = new_end_coords[0][1]

                new_regions.append(chrom + "_" + str(new_start) + "_" + str(new_end)
                new_regions_keep.append(True)

            new_count_table = count_table[new_regions_keep]
            new_count_table.index = new_regions
            return new_count_table

        def liftover_predictdb_genotype_file(genotype_file, from_build="hg19", to_bu

            lo = pyliftover.LiftOver(from_build, to_build)

            genotypes_df = pd.read_csv(genotype_file, sep="\t")
```

```
    chroms = genotypes_df["varID"].apply(lambda x: x.split("_")[0])
    starts = genotypes_df["varID"].apply(lambda x: int(x.split("_")[1]))

    refs = genotypes_df["varID"].apply(lambda x: x.split("_")[2])
    alts = genotypes_df["varID"].apply(lambda x: x.split("_")[3])

    new_starts = []

    unconverted = 0

    for chrom, start in zip(chroms, starts):
        new_coords = lo.convert_coordinate(chrom, start)
        if len(new_coords) == 0:
            unconverted += 1
            new_starts.append("UNCONVERTED")
        else:
            new_starts.append(new_coords[0][1])


    if to_build == "hg19":
        build_tag = "b37"
        from_build_tag = "b38"
    elif to_build == "hg38":
        build_tag = "b38"
        from_build_tag = "b37"

    genotypes_df["varID"] = ["_".join([chrom,str(new_start),str(ref),str(alt

    genotypes_df_no_unconverted = genotypes_df[~genotypes_df["varID"].str.co

    genotypes_df_no_unconverted = genotypes_df_no_unconverted.drop_duplicate

    print(genotypes_df.shape)
    print(genotypes_df_no_unconverted.shape)


    out_file = genotype_file.replace(from_build_tag, build_tag)

    genotypes_df_no_unconverted.to_csv(out_file, sep="\t", index=False)

    print("Unconverted: ", unconverted)

def liftover_snp_annotation_file(annotation_file, from_build="hg19", to_buil

    lo = pyliftover.LiftOver(from_build, to_build)

    annotation_df = pd.read_csv(annotation_file, sep="\t")

    chroms = annotation_df["chr"].apply(lambda x: "chr"+str(x))
    starts = annotation_df["pos"].apply(lambda x: int(x))

    refs = annotation_df["ref_vcf"]
    alts = annotation_df["alt_vcf"]

    new_starts = []
    unconverted = 0

    for chrom, start in zip(chroms, starts):
        new_coords = lo.convert_coordinate(chrom, start)
        if len(new_coords) == 0:
            unconverted += 1
            new_starts.append("UNCONVERTED")
        else:
            new_starts.append(new_coords[0][1])

    if to_build == "hg19":
        build_tag = "b37"
        from_build_tag = "b38"
    elif to_build == "hg38":
        build_tag = "b38"
        from_build_tag = "b37"

    varids = ["_".join([chrom,str(new_start),str(ref),str(alt),build_tag]) f

    annotation_df["varID"] = varids
    annotation_df["rsid"] = varids

    annotation_df["pos"] = new_starts

    annotation_df_no_unconverted = annotation_df[~annotation_df["varID"].str

    print(annotation_df.shape)
    print(annotation_df_no_unconverted.shape)

    out_file = annotation_file.replace(from_build_tag, build_tag)
    annotation_df.to_csv(out_file, sep="\t", index=False)

    print("Unconverted: ", unconverted)

def make_annotation_from_count(cur_counts, modality, annotation_dir, valid_c

    predictdb_annotation_file_dict = {
        "chr":[],
        "gene_id":[],
        "gene_name":[],
        "start":[],
        "end":[],
        "gene_type":[]
    }

    annotation_file_dict = {"ensembl_gene_id":[],
                            "external_gene_name":[],
                            "chromosome_name":[],
                            "transcript_biotype":[],
                            "transcript_start":[],
                            "transcript_end":[],
                            "transcription_start_site":[],
                            "transcript_is_canonical":[],
                            "transcript_count":[],
                            "target_regions":[]
                            }
    for row in cur_counts.iterrows():

        region = row[0]
        if len(region.split("_")) != 3:
            print("region length not 3")
            continue
        chrom = region.split("_")[0]
        if chrom not in valid_chroms:
            print("chromosome not in valid chroms")
            continue
        start = int(region.split("_")[1])
        end = int(region.split("_")[2])

        middle = (start + end) // 2

        predictdb_annotation_file_dict["chr"].append(chrom.strip("chr"))
        predictdb_annotation_file_dict["gene_id"].append(region)
        predictdb_annotation_file_dict["gene_name"].append(region)
        predictdb_annotation_file_dict["start"].append(start)
        predictdb_annotation_file_dict["end"].append(end)
        predictdb_annotation_file_dict["gene_type"].append("protein_coding")


        annotation_file_dict["ensembl_gene_id"].append(region)
        annotation_file_dict["external_gene_name"].append(region)

        annotation_file_dict["chromosome_name"].append(chrom.strip("chr"))
        annotation_file_dict["transcript_biotype"].append("NA")
        annotation_file_dict["transcript_start"].append(start)
        annotation_file_dict["transcript_end"].append(end)
        annotation_file_dict["transcription_start_site"].append(middle)
        annotation_file_dict["transcript_is_canonical"].append(1)
        annotation_file_dict["transcript_count"].append("NA")
        annotation_file_dict["target_regions"].append(region)


    predictdb_annotation_file = pd.DataFrame(predictdb_annotation_file_dict)
    print(predictdb_annotation_file.shape)
    o_file = os.path.join(annotation_dir, f"{modality}_predictdb_annotation.
    predictdb_annotation_file.to_csv(o_file, sep="\t", index=False)

    annotation_file = pd.DataFrame(annotation_file_dict)
    print(annotation_file.shape)
    o_file = os.path.join(annotation_dir, f"{modality}_annotation.txt")
    annotation_file.to_csv(o_file, sep=",", index=False)
```

Reformat normalized count tables, split Flu and NI and subset to genotype file

In [ ]:

```
# # Reformat normalized count tables, split Flu and NI and subset to genotyp

# genotype_file = "/beagle3/haky/users/saideep/projects/aracena_modeling/Inp
# genotype_data = pd.read_csv(genotype_file, sep="\t", index_col=0)
# genotype_data.head()
# genotype_samples = genotype_data.columns

# for modality in modality_pcs.keys():

#     cur_count_file = os.path.join(path_to_count_table,f"fully_preprocessed
#     all_expression_ori = pd.read_csv(
#         cur_count_file,
#          sep=" ")


#     valid_chroms = "chr1,chr2,chr3,chr4,chr5,chr6,chr9,chr10,chr11,chr16,c
#     # Liftover peaks to hg38
#     all_expression= liftover_count_table(all_expression_ori, from_build="h

#     # Make annotation file for EnPACT
#     make_annotation_from_count(all_expression, modality, path_to_annotatio

#     print(all_expression_ori.shape)
#     print(all_expression.shape)

#     # samples = all_expression.columns

#     samples = all_expression.columns

#     flu_samples = [x for x in samples if "Flu" in x]
#     ni_samples = [x for x in samples if "NI" in x]

#     flu_samples = [x for x in flu_samples if x.split("_")[0] in genotype_s
#     ni_samples = [x for x in ni_samples if x.split("_")[0] in genotype_sam

#     flu_expression = all_expression[flu_samples]
#     ni_expression = all_expression[ni_samples]

#     flu_expression = flu_expression.rename(columns={x:x.replace("_Flu","")
#     ni_expression = ni_expression.rename(columns={x:x.replace("_NI","") fo

#     print(flu_expression.shape)
#     print(ni_expression.shape)

#     # Write reformated files

#     flu_expression.to_csv(cur_count_file.replace("preprocessed_", "preproc
#     ni_expression.to_csv(cur_count_file.replace("preprocessed_", "preproce
```

Liftover inputs for PredictDB

This isn't always necessary if your inputs are all already in the same genome build.

In [ ]:
```
genotype_file_to_convert = "/beagle3/haky/users/saideep/projects/aracena_mod
annotation_file_to_convert = "/beagle3/haky/users/saideep/projects/aracena_m
```

```
converted_genotype_file = genotype_file_to_convert.replace("b37", "b38")
converted_annotation_file = annotation_file_to_convert.replace("b37", "b38")
```

In [ ]:
```
# Convert genotype file to hg38 (shared among all runs)

# liftover_predictdb_genotype_file(genotype_file_to_convert, from_build="hg1
```

In [ ]:
```
# Convert SNP annotation file to hg38 (shared among all runs)

# liftover_snp_annotation_file(annotation_file_to_convert, from_build="hg19"
```

Train EnPACT models

In [ ]:
```
window_sizes = [2,4,8,16,32,64,128]
```

In [ ]:
```
for modality in modality_pcs.keys():
    for context in ["Flu","NI"]:
        for window_size in window_sizes:
            cur_proj_dir = os.path.join(path_to_project_dir,context+"_"+moda
            os.makedirs(cur_proj_dir,exist_ok=True)

            with open(path_to_config_template,"r") as f:

                config = json.load(f)
                config["general_parameters"]["project_directory"] = cur_proj
                config["general_parameters"]["context"] = context

                config["generate_enpact_training_data"]["input_files"]["norm
                config["generate_enpact_training_data"]["reference_epigenome
                config["generate_enpact_training_data"]["input_files"]["gene

                with open(os.path.join(cur_proj_dir,"config.json"),"w") as f
                    json.dump(config,f, indent=4)

            with open(path_to_run_template,"r") as f:
                run = f.read()
                run = run.replace("CONFIG_FILE",os.path.join(cur_proj_dir,"c
                run = run.replace("JOBNAME",context+"_"+modality)
                run = run.replace("ERROR_LOG", os.path.join(cur_proj_dir,"er
                run = run.replace("OUTPUT_LOG", os.path.join(cur_proj_dir,"c
                with open(os.path.join(cur_proj_dir,"run_training.sbatch"),"
                    f.write(run)

            # subprocess.run(["sbatch",
            #                 os.path.join(cur_proj_dir,"run_training.sbatch
            #                 cwd=cur_proj_dir)
```

Personalized prediction evaluation

Before moving on to do linearization and TWAS, it is important to examine the raw personalized prediction performance of the EnPACT model. Also, based on analyzing the window sizes during training, we can decide on the window size for personalized prediction and linearization. They are stored below:

In [ ]:
```
optimal_window_sizes = {
    "H3K27ac":8,
    "H3K27me3":64,
    "H3K4me1":32,
    "H3K4me3":8,
    "ATAC":8
}

nfolds_per_mode = {
    "H3K27ac":3,
    "H3K27me3":3,
    "H3K4me1":3,
    "H3K4me3":3,
    "ATAC":3
}

closest_enformer_track = {
    "H3K27ac":8,
    "H3K27me3":64,
    "H3K4me1":32,
    "H3K4me3":8,
    "ATAC":517
}

max_features = 500

path_to_pdb_standard_template = "/beagle3/haky/users/saideep/github_repos/Co
path_to_pp_template = "/beagle3/haky/users/saideep/github_repos/Con-EnPACT/r
```

Prepare VCF files for personalized prediction

Personalized prediction requires a VCF file encoding individual's genotypes. In this case, the existing VCF is in hg19 format, and the sample names are somewhat confusingly named relative to the the other files. Liftover is not required as long as the other input files are in hg19 format, but it will help to rename the VCFs samples for consistency.

In [ ]:
```
vcf_dir = "/beagle3/haky/users/saideep/projects/aracena_modeling/Inputs/VCF/

import glob
import cyvcf2

vcf_files = glob.glob(os.path.join(vcf_dir,"*.dose.vcf.gz.chr.vcf.gz.snps.vc

new_vcf_dir = "/beagle3/haky/users/saideep/projects/aracena_modeling/Inputs/

reheader_script_dir = "/beagle3/haky/users/saideep/github_repos/Daily-Blog-S

bcftools = "/beagle3/haky/users/saideep/software/bcftools-1.19/bcftools"

# with open(os.path.join(reheader_script_dir,"reheader.sh"), "w") as rs:
```

```
#     for vcf_file in vcf_files:


#         new_vcf_file = os.path.join(new_vcf_dir,os.path.basename(vcf_file)

#         cur_samples = cyvcf2.VCF(vcf_file).samples
#         new_samples = [x.split("_")[1] for x in cur_samples]

#         new_samples_file = os.path.join(reheader_script_dir,os.path.basena
#         with open(new_samples_file, "w") as f:
#             f.write("\n".join(new_samples))

#         rs.write(f"{bcftools} reheader -s {new_samples_file} -o {new_vcf_f
#         rs.write(f"tabix -p vcf {new_vcf_file}\n\n")
```

Prepare config file for personalized prediction and run the relevant pipeline step

Personalized prediction requires running PredictDB training directly on the study population. This helps select features we will use for EnPACT personalized prediction as well as provide a model comparison. Therefore, the code below also generates the run script for standard predictDB. You should run that first, and then after the filtered db is created run the personalized prediction runscript.

In [ ]:
```
for modality in modality_pcs.keys():
    for context in ["Flu","NI"]:
        cur_window_size = optimal_window_sizes[modality]
        cur_proj_dir = os.path.join(path_to_project_dir,context+"_"+modality

        with open(os.path.join(cur_proj_dir,"config.json"),"r") as f:

            config = json.load(f)

            config["predictDB_standard"]["genotype_file"] = converted_genoty
            config["predictDB_standard"]["snp_annotation_file"] = converted_
            config["predictDB_standard"]["feature_annotation_file"] = os.pat
            config["predictDB_standard"]["nfolds"] = nfolds_per_mode[modalit

            config["personalized_predictions"]["max_features"] = max_feature
            config["personalized_predictions"]["path_to_epigenome_predictior
            config["personalized_predictions"]["path_to_epigenome_config"] =
            config["personalized_predictions"]["path_to_vcf"] = new_vcf_dir
            config["personalized_predictions"]["epigenome_config_parameters"
            config["personalized_predictions"]["num_bins"] = optimal_window_
            config["personalized_predictions"]["date"] = "04-16-2024"

            config["personalized_predictions"]["liftover"] = True
            config["personalized_predictions"]["liftover_target"] = "hg19"


        with open(os.path.join(cur_proj_dir,"config.json"),"w") as f:
            json.dump(config,f, indent=4)
        with open(path_to_pdb_standard_template,"r") as f:
            run = f.read()
            run = run.replace("CONFIG_FILE",os.path.join(cur_proj_dir,"confi
            run = run.replace("JOBNAME",context+"_"+modality)
            run = run.replace("ERROR_LOG", os.path.join(cur_proj_dir,"error_
            run = run.replace("OUTPUT_LOG", os.path.join(cur_proj_dir,"outpu
            with open(os.path.join(cur_proj_dir,"run_predictdb_standard.sbat
                f.write(run)

        with open(path_to_pp_template,"r") as f:
            run = f.read()
            run = run.replace("CONFIG_FILE",os.path.join(cur_proj_dir,"confi
            run = run.replace("JOBNAME",context+"_"+modality)
            run = run.replace("ERROR_LOG", os.path.join(cur_proj_dir,"error_
            run = run.replace("OUTPUT_LOG", os.path.join(cur_proj_dir,"outpu
            with open(os.path.join(cur_proj_dir,"run_personalized_prediction
                f.write(run)

        # subprocess.run(["sbatch", os.path.join(cur_proj_dir,"run_personali
        # if modality != "ATAC":
        #     if context == "Flu":
        #         subprocess.run(["sbatch", os.path.join(cur_proj_dir,"inter
        #                                 "run_predictDB_standard.sba
        #                                 cwd=os.path.join(cur_proj_c
```

Linearize Peak EnPACT models

Unlike gene expression, peak data is relatively heterogenous. Different studies can have different sets of peaks, and the total number often greatly exceeds the number of genes. It makes sense, therefore, to have a selection process rather than trying to train PredictDB models for every peak available. In this case, since we are doing follow-up TWAS analysis, we can use GWAS loci to help us select peaks of interest.

We can start by examining examining a few GWAS summary stats and look for strong signals. These loci can be used as target sites for linearization since they are likely to have causal signal and are the places with potential to show up in downstream TWAS analysis.

Which GWAS to use?:

- Broad Allergy
- COVID GWAS
- Other infectious disease GWAS

GPU-hour costs for Enformer inference: 5hr per 20,000 sites

Set up loci selection

I am using Temi's repo: https://github.com/hakyimlab/TFXcan-snakemake/tree/main, which performs fine-mapping from GWAS summary stats, selects loci of interest, and then runs Enformer predictions for these regions. Once these personalized predictions are made, they can be used in all downstream EnPACT analyses.

EDIT: I've decided not to automate this here because it can be a pretty big pain to deal with.

In [ ]:
```
# Dictionary to be added to config file
linearization_datasets = {
    "COVID":{
        "features":"/beagle3/haky/users/saideep/projects/aracena_modeling/SP
        "individuals":"/beagle3/haky/users/saideep/projects/enformer_all_gen
        "genotype_file":"/beagle3/haky/users/charles/project/singleXcanDL/Pr
        "snp_annotation_file":"/beagle3/haky/users/charles/project/singleXca
        "gene_annotation_file":"/beagle3/haky/users/saideep/projects/aracena
        "epigenome_pred_dir":"/beagle3/haky/users/saideep/projects/aracena_m
    }
}
```

In [ ]:
```
path_to_linearization_template = "/beagle3/haky/users/saideep/github_repos/C

for modality in modality_pcs.keys():
    for context in ["Flu","NI"]:
        cur_window_size = optimal_window_sizes[modality]
        cur_proj_dir = os.path.join(path_to_project_dir,context+"_"+modality

        with open(os.path.join(cur_proj_dir,"config.json"),"r") as f:

            config = json.load(f)
            config["linearization"]["linearization_datasets"] = linearizatio

        with open(os.path.join(cur_proj_dir,"config.json"),"w") as f:
            json.dump(config,f, indent=4)

        with open(path_to_linearization_template,"r") as f:
            run = f.read()
            run = run.replace("CONFIG_FILE",os.path.join(cur_proj_dir,"confi
            run = run.replace("JOBNAME",context+"_"+modality)
            run = run.replace("ERROR_LOG", os.path.join(cur_proj_dir,"error_
            run = run.replace("OUTPUT_LOG", os.path.join(cur_proj_dir,"outpu
            with open(os.path.join(cur_proj_dir,"run_linearization.sbatch"),
                f.write(run)

        subprocess.run(["sbatch", os.path.join(cur_proj_dir,"run_linearizati
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
```

```
Submitted batch job 21726208
Submitted batch job 21726209
```

```
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
```

```
Submitted batch job 21726210
Submitted batch job 21726211
Submitted batch job 21726212
```

```
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
```

```
Submitted batch job 21726213
Submitted batch job 21726214
Submitted batch job 21726215
Submitted batch job 21726216
Submitted batch job 21726217
```

```
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
```

Run XWAS

Finally, we are ready to run XWAS analysis. Luckily, assuming all of the above have run successfully this should be a reasonable task.

In [ ]:
```
# Dictionary to be added to config file
XWAS_datasets = {
    "COVID":{
        "GWAS_sum_stats":"/beagle3/haky/users/saideep/projects/aracena_model
        "linearization_dataset":"COVID"
    }
}
```

In [ ]:
```
path_to_xwas_template = "/beagle3/haky/users/saideep/github_repos/Con-EnPACT

for modality in modality_pcs.keys():
    for context in ["Flu","NI"]:
        cur_window_size = optimal_window_sizes[modality]
        cur_proj_dir = os.path.join(path_to_project_dir,context+"_"+modality

        with open(os.path.join(cur_proj_dir,"config.json"),"r") as f:

            config = json.load(f)
            config["XWAS"]["XWAS_datasets"] = XWAS_datasets

        with open(os.path.join(cur_proj_dir,"config.json"),"w") as f:
            json.dump(config,f, indent=4)

        with open(path_to_xwas_template,"r") as f:
            run = f.read()
            run = run.replace("CONFIG_FILE",os.path.join(cur_proj_dir,"confi
            run = run.replace("JOBNAME",context+"_"+modality)
            run = run.replace("ERROR_LOG", os.path.join(cur_proj_dir,"error_
            run = run.replace("OUTPUT_LOG", os.path.join(cur_proj_dir,"outpu
            with open(os.path.join(cur_proj_dir,"run_xwas.sbatch"),"w") as f
```

```
                f.write(run)

        subprocess.run(["sbatch", os.path.join(cur_proj_dir,"run_xwas.sbatch
```

```
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
Submitted batch job 21740221
Submitted batch job 21740222
Submitted batch job 21740223
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
Submitted batch job 21740224
Submitted batch job 21740225
Submitted batch job 21740226
```

```
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
Submitted batch job 21740227
Submitted batch job 21740228
Submitted batch job 21740229
Submitted batch job 21740230
sbatch: Verify job submission ...
sbatch: Using a shared partition ...
sbatch: Partition: caslake
sbatch: QOS-Flag: caslake
sbatch: Account: pi-haky
sbatch: Verification: ***PASSED***
```

Let's also collect the XWAS results

In [ ]:

```
results_tables = []

for modality in optimal_window_sizes.keys():
    for cond in ["Flu","NI"]:

        for xd in XWAS_datasets.keys():

            cur_window_size = optimal_window_sizes[modality]

            cur_proj_dir = os.path.join(path_to_project_dir,cond+"_"+modalit
            path_to_results = os.path.join(cur_proj_dir,"intermediates","XWA

            results = pd.read_csv(path_to_results, sep=",")
            results["modality"] = modality
            results["condition"] = cond
            results["GWAS_source"] = xd

            header=results.iloc[0]

            print(results.head())

            results_tables.append(results)
```

```
results_table_full = pd.concat(results_tables)
print(results_table_full.shape)

results_table_full.to_csv("/beagle3/haky/users/saideep/github_repos/Daily-Bl
                        gene                  gene_name    zscore  \
0     chr3_45838989_45838991     chr3_45838989_45838991 -4.046273   
1      chr19_4719431_4719433      chr19_4719431_4719433  3.616343   
2   chr1_155203736_155203738   chr1_155203736_155203738 -3.470710   
3  chr12_112936943_112936945  chr12_112936943_112936945  3.347433   
4   chr9_133273813_133273815   chr9_133273813_133273815  3.083231   

   effect_size    pvalue     var_g  pred_perf_r2  pred_perf_pval  \
0   -12.541206  0.000052  0.000035      0.679051   4.986974e-140   
1     2.609658  0.000299  0.001313      0.742334   8.630024e-166   
2   -29.283244  0.000519  0.000006      0.757438   4.275992e-200   
3     3.680519  0.000816  0.000349      0.915979    0.000000e+00   
4     1.167531  0.002048  0.003344      0.993883    0.000000e+00   

   pred_perf_qval  n_snps_used  n_snps_in_cov  n_snps_in_model modality  \
0             NaN          191            213              213  H3K27ac   
1             NaN          226            317              317  H3K27ac   
2             NaN           14             15               15  H3K27ac   
3             NaN          133            174              174  H3K27ac   
4             NaN           45             56               56  H3K27ac   

  condition   GWAS  
0       Flu  COVID  
1       Flu  COVID  
2       Flu  COVID  
3       Flu  COVID  
4       Flu  COVID  
                       gene                 gene_name    zscore  effect_siz
e  \
0    chr3_45838989_45838991    chr3_45838989_45838991 -6.463302   -16.45281
8   
1   chr17_49863303_49863305   chr17_49863303_49863305 -4.596455    -1.93195
1   
2   chr21_33242905_33242907   chr21_33242905_33242907 -3.570149    -4.81877
0   
3  chr9_133273813_133273815  chr9_133273813_133273815  3.549046     1.29734
8   
4   chr19_50379362_50379364   chr19_50379362_50379364  3.024097     2.90513
5   

         pvalue     var_g  pred_perf_r2  pred_perf_pval  pred_perf_qval  \
0  1.024422e-10  0.000062      0.710164   6.453670e-144             NaN   
1  4.297390e-06  0.002672      0.882065    0.000000e+00             NaN   
2  3.567786e-04  0.000196      0.772582   6.033189e-179             NaN   
3  3.866297e-04  0.003634      0.995342    0.000000e+00             NaN   
4  2.493761e-03  0.000384      0.854325   8.185288e-274             NaN   

   n_snps_used  n_snps_in_cov  n_snps_in_model modality condition   GWAS  
0          154            162              162  H3K27ac        NI  COVID  
1          136            160              160  H3K27ac        NI  COVID  
2          167            187              187  H3K27ac        NI  COVID  
3           70             74               74  H3K27ac        NI  COVID  
4           97            128              128  H3K27ac        NI  COVID  
                      gene                gene_name    zscore  effect_size
\
0   chr6_31153455_31153457   chr6_31153455_31153457 -3.686652    -1.041724
```

```
1  chr17_49863303_49863305  chr17_49863303_49863305 -3.314173    -3.383596
2    chr19_4719431_4719433    chr19_4719431_4719433 -3.077552    -1.828643
3  chr19_48867352_48867354  chr19_48867352_48867354 -2.821971    -2.351931
4   chr3_45838989_45838991   chr3_45838989_45838991  2.700055     1.143763

     pvalue     var_g  pred_perf_r2  pred_perf_pval  pred_perf_qval  \
0  0.000227  0.004325      0.989233    0.000000e+00             NaN   
1  0.000919  0.000471      0.865508   4.692603e-262             NaN   
2  0.002087  0.000904      0.902207    0.000000e+00             NaN   
3  0.004773  0.000683      0.898959    0.000000e+00             NaN   
4  0.006933  0.001806      0.893569    0.000000e+00             NaN   

   n_snps_used  n_snps_in_cov  n_snps_in_model  modality condition   GWAS  
0          127            138              138  H3K27me3       Flu  COVID  
1          158            190              190  H3K27me3       Flu  COVID  
2          102            152              152  H3K27me3       Flu  COVID  
3          119            165              165  H3K27me3       Flu  COVID  
4          115            126              126  H3K27me3       Flu  COVID  
                         gene                  gene_name    zscore  \
0      chr19_4719431_4719433      chr19_4719431_4719433 -4.186584   
1     chr6_31153455_31153457     chr6_31153455_31153457 -3.738601   
2    chr17_49863303_49863305    chr17_49863303_49863305 -2.978499   
3  chr12_112936943_112936945  chr12_112936943_112936945  2.758486   
4     chr3_45838989_45838991     chr3_45838989_45838991  2.684772   

   effect_size    pvalue     var_g  pred_perf_r2  pred_perf_pval  \
0    -2.311632  0.000028  0.001086      0.893918    0.000000e+00   
1    -1.018564  0.000185  0.004601      0.989930    0.000000e+00   
2    -3.028647  0.002897  0.000472      0.870368   2.503403e-267   
3     6.500054  0.005807  0.000068      0.835311   2.062655e-226   
4     1.130361  0.007258  0.001833      0.902027    0.000000e+00   

   pred_perf_qval  n_snps_used  n_snps_in_cov  n_snps_in_model  modality  \
0             NaN          161            228              228  H3K27me3   
1             NaN          123            132              132  H3K27me3   
2             NaN          175            208              208  H3K27me3   
3             NaN          153            186              186  H3K27me3   
4             NaN          120            131              131  H3K27me3   

  condition   GWAS  
0        NI  COVID  
1        NI  COVID  
2        NI  COVID  
3        NI  COVID  
4        NI  COVID  
                       gene                 gene_name    zscore  effect_siz
e  \
0    chr19_4719431_4719433    chr19_4719431_4719433  7.166042     5.19273
0   
1   chr21_33242905_33242907   chr21_33242905_33242907 -6.021281    -4.21204
7   
2   chr17_49863303_49863305   chr17_49863303_49863305 -5.044558    -4.01721
6   
3  chr9_133273813_133273815  chr9_133273813_133273815  4.082147     1.98663
2   
4    chr6_31153455_31153457    chr6_31153455_31153457  4.068298     1.30432
```

```
8

         pvalue     var_g  pred_perf_r2  pred_perf_pval  pred_perf_qval  \
0  7.719687e-13  0.000858      0.742689   6.624718e-166             NaN   
1  1.730420e-09  0.000650      0.917593    0.000000e+00             NaN   
2  4.545710e-07  0.000746      0.787245    0.000000e+00             NaN   
3  4.462149e-05  0.002065      0.993723    0.000000e+00             NaN   
4  4.735777e-05  0.003252      0.981365    0.000000e+00             NaN   

   n_snps_used  n_snps_in_cov  n_snps_in_model modality condition   GWAS  
0          131            193              193  H3K4me1       Flu  COVID  
1          139            162              162  H3K4me1       Flu  COVID  
2          131            163              163  H3K4me1       Flu  COVID  
3           66             83               83  H3K4me1       Flu  COVID  
4          174            192              192  H3K4me1       Flu  COVID  
                   gene                 gene_name    zscore  effect_siz
e  \
0     chr19_4719431_4719433     chr19_4719431_4719433  6.871495     5.05601
0   
1   chr21_33242905_33242907   chr21_33242905_33242907 -5.683134    -4.02402
9   
2   chr17_49863303_49863305   chr17_49863303_49863305 -4.940898    -3.61424
9   
3    chr6_31153455_31153457    chr6_31153455_31153457  4.062374     1.26317
4   
4  chr9_133273813_133273815  chr9_133273813_133273815  3.940389     1.82773
3   

         pvalue     var_g  pred_perf_r2  pred_perf_pval  pred_perf_qval  \
0  6.353239e-12  0.000833      0.697355   1.105875e-150             NaN   
1  1.322482e-08  0.000652      0.918320    0.000000e+00             NaN   
2  7.776357e-07  0.000882      0.890964    0.000000e+00             NaN   
3  4.857627e-05  0.003434      0.978835    0.000000e+00             NaN   
4  8.134961e-05  0.002284      0.994095    0.000000e+00             NaN   

   n_snps_used  n_snps_in_cov  n_snps_in_model modality condition   GWAS  
0          156            225              225  H3K4me1        NI  COVID  
1          171            195              195  H3K4me1        NI  COVID  
2          108            134              134  H3K4me1        NI  COVID  
3          162            177              177  H3K4me1        NI  COVID  
4           44             60               60  H3K4me1        NI  COVID  
                     gene                   gene_name    zscore  \
0    chr21_33242905_33242907    chr21_33242905_33242907 -6.554030   
1  chr12_112936943_112936945  chr12_112936943_112936945  6.393212   
2    chr17_49863303_49863305    chr17_49863303_49863305  4.900311   
3    chr19_50379362_50379364    chr19_50379362_50379364  4.222593   
4      chr19_4719431_4719433      chr19_4719431_4719433  4.037497   

   effect_size        pvalue     var_g  pred_perf_r2  pred_perf_pval  \
0    -6.260170  5.600477e-11  0.000413      0.879327   1.692943e-283   
1     4.881488  1.624368e-10  0.000647      0.974020    0.000000e+00   
2     1.507517  9.568527e-07  0.004953      0.933669    0.000000e+00   
3     3.417448  2.415081e-05  0.000501      0.953136    0.000000e+00   
4     3.229335  5.402464e-05  0.000879      0.738895   1.763889e-158   

   pred_perf_qval  n_snps_used  n_snps_in_cov  n_snps_in_model modality  \
0            NaN          205            244              244  H3K4me3
1            NaN           84            102              102  H3K4me3
2            NaN           91            101              101  H3K4me3
3            NaN           54             66               66  H3K4me3
4            NaN          149            220              220  H3K4me3

  condition   GWAS
0       Flu  COVID
1       Flu  COVID
2       Flu  COVID
3       Flu  COVID
4       Flu  COVID
                      gene                  gene_name    zscore  \
0    chr21_33242905_33242907    chr21_33242905_33242907 -7.217360
1  chr12_112936943_112936945  chr12_112936943_112936945  5.867753
2    chr17_49863303_49863305    chr17_49863303_49863305  4.460480
3   chr9_133273813_133273815   chr9_133273813_133273815  4.048731
4    chr19_50379362_50379364    chr19_50379362_50379364  3.871139

   effect_size        pvalue     var_g  pred_perf_r2  pred_perf_pval  \
0    -6.778290  5.300657e-13  0.000445      0.875814   1.098475e-276
1     5.788117  4.417404e-09  0.000384      0.912724    0.000000e+00
2     3.788385  8.177643e-06  0.000671      0.890077    0.000000e+00
3     3.623972  5.149614e-05  0.000568      0.963383    0.000000e+00
4     3.475773  1.083280e-04  0.000425      0.935749    0.000000e+00

   pred_perf_qval  n_snps_used  n_snps_in_cov  n_snps_in_model modality  \
0            NaN          137            159              159  H3K4me3
1            NaN          172            227              227  H3K4me3
2            NaN          102            117              117  H3K4me3
3            NaN          145            178              178  H3K4me3
4            NaN           93            120              120  H3K4me3

  condition   GWAS
0        NI  COVID
1        NI  COVID
2        NI  COVID
3        NI  COVID
4        NI  COVID
                     gene                 gene_name    zscore  effect_siz
e  \
0   chr21_33242905_33242907   chr21_33242905_33242907 -5.589137    -6.95990
4
1     chr19_4719431_4719433     chr19_4719431_4719433  5.080926    12.28892
5
2   chr17_49863303_49863305   chr17_49863303_49863305 -4.737094    -3.25776
4
3  chr9_133273813_133273815  chr9_133273813_133273815  3.925288     2.30850
2
4   chr19_50379362_50379364   chr19_50379362_50379364  2.589867     2.19456
4

         pvalue     var_g  pred_perf_r2  pred_perf_pval  pred_perf_qval  \
0  2.282012e-08  0.000218      0.852828   1.751426e-251             NaN
1  3.755987e-07  0.000101      0.511263    8.683987e-80             NaN
2  2.168046e-06  0.000995      0.911933    0.000000e+00             NaN
```

```
3  8.662607e-05  0.001287      0.996425    0.000000e+00             NaN
4  9.601297e-03  0.000431      0.950349    0.000000e+00             NaN

   n_snps_used  n_snps_in_cov  n_snps_in_model modality condition   GWAS
0          138            158              158     ATAC       Flu  COVID
1           90            138              138     ATAC       Flu  COVID
2           74             93               93     ATAC       Flu  COVID
3           52             60               60     ATAC       Flu  COVID
4          107            139              139     ATAC       Flu  COVID
                        gene                  gene_name    zscore  effect_siz
e  \
0   chr17_49863303_49863305   chr17_49863303_49863305 -4.878925    -2.39873
4
1   chr21_33242905_33242907   chr21_33242905_33242907 -4.070918    -5.62426
0
2  chr9_133273813_133273815  chr9_133273813_133273815  3.819061     3.07464
4
3     chr19_4719431_4719433     chr19_4719431_4719433 -3.488570    -7.91230
8
4    chr3_45838989_45838991    chr3_45838989_45838991 -3.384359    -7.73003
9

     pvalue     var_g  pred_perf_r2  pred_perf_pval  pred_perf_qval  \
0  0.000001  0.001937      0.920091    0.000000e+00             NaN
1  0.000047  0.000179      0.826215   1.277127e-227             NaN
2  0.000134  0.000689      0.992612    0.000000e+00             NaN
3  0.000486  0.000033      0.501268    2.060973e-77             NaN
4  0.000713  0.000069      0.800622   1.797418e-213             NaN

   n_snps_used  n_snps_in_cov  n_snps_in_model modality condition   GWAS
0           82            104              104     ATAC        NI  COVID
1          127            143              143     ATAC        NI  COVID
2           58             78               78     ATAC        NI  COVID
3           80            112              112     ATAC        NI  COVID
4          153            167              167     ATAC        NI  COVID
(220, 15)
```