SNP Overlap Manual [*]

David Nicholson

June 16, 2017

Quick Start

This program is broken down into three commands: **Plink**, **Bedtools**, **Heatmap**. The options for these commands will be explained further in this document; however, this segment is designed to provide a quick tutorial on how to run this program. Please use the following code segments below:

```
python snp_overlap_runner.py Plink --snp_file <GWAS snps file>
--imputation_path <Imputed SNPS folder>

python snp_overlap_runner.py Bedtools --peak_file <Peak Desc File>

python snp_overlap_runner.py Heatmap --peak_file <Peak Desc File>
```

From these provided commands the program will automatically generate files and folders within the directory these commands were issued in. The folders generated will be called `overlap_files` and `plink_temp`. These generated folders are used to store temporary files and the final results as well. Plink temp folder is a temporary folder that holds plink files that can be removed in the end. The overlap files folder is what you will be navigating through to obtain the final results of each run. Lastly, the rest of this manual will further describe the available arguments this program will take as input.

Program Requirements

This program needs to have the follow system requirements:

- python 2.7
- python packages:

[*] Any questions or concerns email: davnic@mail.med.upenn.edu

  - pandas (current version)
  - colour (current version)
  - matplotlib (current version)
  - seaborn (current version)

- bedtools (current version)

- plink (current version)

Plink Command [1]

This command tells the program to execute plink. Plink is necessary to calculate R squared values between imputed SNPS and a given sentinel SNP. Currently, plink is not insalled on the cluster, so for this command to work you will have to manually download plink from an online resource. Try this link here.
Following this link, all you have to do is download the linux binary file and place it into the same directory the runner script. After plink has been installed, feel free to ignore the rest of this section and refer to the quick start above to run the command. Nonetheless, here is a description of the arguments this program uses below:

- `imputation_path`

  This **required** argument is used to specify the location of the imputed snps. These imputed snps need to be in variant call format (vcf). Currently I am using the WTCCC imputation path for this manual, but feel free to change it any other desired path.

  Ex.

  ```
  python snp_overlap_runner.py Plink
  --imputation_path /mnt/isilon/grant_lab/WTCCCC snps/
  ```

- `snp_file`

  This **required** argument is used to specify the file location of sentinel snps obtained from genome wide assocation studeis (GWAS). Usually these files are in either comma separated value format (CSV) or tab separated value format (TSV). This program does not accept any other file formats, so please do keep that in mind. Furthermore, this file needs the following column headers: SNP, Locus and Chr. They dont have to be in that particular order,

[1] If running on cluster, make sure to allocate at least 8Gs. Otherwise the submitted job will be killed.

but the header name needs to be extacly as seen in this manual. Last important disclaimer: within each field of the file do not have a tab or comma. These characters will cause an error within the program or produce an undesired result. To reinforce this concept an example of this problematic entry is highlighted below.

Ex.

```
python snp_overlap_runner.py Plink --snp_file /path to snps/snp_file.csv
```

Ex. File:

SNP	Locus	Chr	Trait	EA	NEA	Position
rs1234	PPARG	1	WHR	T	C	1234
rs4567	PPARG	1	WHR	T	C	1,234,123

- `r_squared`

  This **optional** argument tells the program to use a different cut off r^2 value than the default (0.4).

  Ex.

  ```
  python snp_overlap_runner.py --r_squared 0.2
   --snp_file <your favorite path here>
   --imputation_path <your favority path here>
  ```

- `output_file`

  This **optional** argument tells the program, where you want to save the results of the plink run. The file generated is in bed format. Reason for this format is that this file is needed for the next command (Bedtools command). Note: This file will not have a header, so please refer to the example output lines below. The implied headers are: chromosome, start, end, snp, locus, `r_squared`.

  Ex.

  ```
  python snp_overlap_runner.py  --snp_file <your favorite path here>
   --imputation_path <your favorite path here>
   --output_file overlap_files/imputed_snps.bed
  ```

  Ex. File

chr1	12345	12346	rs40394857	TBX15-WARS2	1.0
chr1	12455	12456	rs2948576	TBX15-WARS2	0.96

Bedtools Command

This command tells the program to use the Bedtools program to determine what SNPs overlaps next generation sequencing (NGS) called peaks. A note for this part is to make sure the SNPS in the previous command are in the same genome build as the NGS peaks. (hg19 vs hg18 vs hg38). If you do not understand this bit of information, please navigate to this link: What is a genome build? If these coordinates are not in the same version, then the results can't be trusted.
Luckily this tool is on the cluster, so no need to install it. Please use this command to load the tool:

```
module load bedtools/2.25.0
```

- `peak_file`

  This **required** argument tells the program the location of the peak description file. This file is designed to point the program to the destination of each called peak file and the file format is either a TSV file or a CSV file. (if you don't know what TSV or CSV is please refer to snp file.) The headers for this file are as follows: Label, Tissue, Score, `File_Path`, and Priority. The Label field tells the program what nickname to use for each peak file. The Tissue field tells the program what tissue these called peaks originated. The Score field ranks each overlap peak with a user specified weight (preferably whole numbers). The File Path field is the absolute file path where each bed peak call is located. Lastly, the prioity field tells the program which peaks to sort by during the write out phase.

  Ex.

  ```
  python snp_overlap_runner.py Bedtools
  --peak_file <path to peak desc file>
  ```

  Ex File.

Label	Tissue	Score	`File_Path`	Priority
ATAC	Fat	1	BED/ATAC.bed	1
H3K4me1	Fat	3	ENCODE/H3K4me1.bed	0
H3K36me3	Fat	-3	ENCODE/H3K36me3.bed	0

- `snp_file`

  This **optional** argument tells the program where to find the imputed snp bed file. If you used the Plink command prior to the

Bedtool command, then you might be able to ignore this command. This program is designed look for the necessary files within the default settings. If a different setting was used then, you will need to use this argument to tell the program where to look.

Ex.

```
python snp_overlap_runner.py Bedtools
--peak_file <path to peak desc file>
--snp_file <path to imputed snps bed file>
```

- window_size

  This **optional** argument tells the program what window buffer to use when determineing what snp overlaps a peak. The default for this command is 50 basepairs. Disclaimer for this command, if you use a higher window size, the chance of a snp overlapping multiple peaks in one file increases as well; therefore, if the results look abnormal, please keep this fact in mind.

  Ex.

  ```
  python snp_overlap_runner.py Bedtools
  --peak_file <path to peak desc file>
  --window_size 100
  ```

- locus

  Sometimes the user may not want to print out every peak result at once. That's where this **optional** argument comes into play. This argument tells the program to only focus on this specific locus (case sensitive) and ignore the others.

  Ex

  ```
  python snp_overlap_runner.py Bedtools
  --peak_file <path to peak desc file>
  --locus TCF7L2
  ```

- tissue

  Sometimes the user may not want to print out every peak result at once. That's where this **optional** argument comes into play. This argument tells the program to only be focused on this specific tissue (case sensitive) and ignore the others.

  Ex

```
python snp_overlap_runner.py Bedtools
--peak_file <path to peak desc file>
--tissue Fat
```

- overlap_output

  This **optional** argument tell the program where to save the output of the bedtools command. The output of the results will be in bed format with no header. The following implied headers are: peak chromosome, peak start, peak stop, snp chromosome, snp start, snp end, snp, locus, sentinel snp, r_squared, peak path, tissue. A visual depiction is proivided below.

  Ex.

  ```
  python snp_overlap_runner.py Bedtools
  --peak_file <path to peak desc file>
  --overlap_output <path to overlap output>
  ```

  Ex File.

Peak Chr	Peak Start	Peak Stop	SNP Chr	SNP Start	SNP Stop	SNP	Locus	Sentinel	r_squared	Peak Path	Tissue
chr1	119	320	chr1	200	201	rs1234	TBX15-WARS2	rs674849	0.46	BED/ATAC	Fat
chr20	544	744	chr20	600	601	rs7890987	GDF5	rs234235235	0.653	BED/ATAC	Fat

- snp_output

  This **optional** argument tell the program where to output a summary of interesting snps and the peaks they overlap.

  Ex.

  ```
  python snp_overlap_runner.py Bedtools
  --peak_file <path to peak desc file>
  --snp_output <path to summary output>
  ```

  Ex File.

Chr	Start	End	SNP	Locus	r_squared	Sentinel	Tissue	ATAC 1	ATAC 2	Combined
chr6	12345	12346	rs1234	RSPO3	0.9	rs556655	Fat	3	3	6
chr12	22345	22346	rs41234	CCDC92	0.61	rs39495866	Fat	3	1	4

Heatmap Command

This command is used to generate an overlap heatmap. This kind of heatmap is designed to graphically show how many peaks a snp overlaps. This command works both in 2D and 3D; however, please note the 3D version is still in development and may not work as intended. Lastly, after these commands are described an example of each figure type will be displayed.

- peak_file

  This **required** argument tells the program where to find the peak description file. This file is described in detail above, so please refer to peak desc file.

  Ex.

  ```
  python snp_overlap_runner.py Heatmap
  --peak_file <path to peak desc file>
  ```

- snp_file

  This **optional** argument tells the program where to find the imputed snp bed file. If you used the Plink command prior to the Heatmap command and you used the default settings, feel free to ignore this arugment. This program is designed search through the default folders and find the necessary files. If different settings were changed then, you will need to use this argument to tell the program where to look.

  Ex.

  ```
  python snp_overlap_runner.py Heatmap
  --peak_file <path to peak desc file>
  --snp_file <path to imputed snps bed file>
  ```

- overlap_file

  This **optional** argument tells the program where to look for the output of the Bedtools command above. If you are sticking with the default setting, then you dont have to use this argument.

  Ex.

  ```
  python snp_overlap_runner.py Heatmap
  --peak_file <path to peak desc file>
  --overlap_file <path to bedtools output>
  ```

- heatmap_folder

  This **optional** argument tells the program where to output the generated heatmap files. Since each file is broken down by locus, a separate folder is needed to organize the output. If using the default settings, this command is not needed.

  Ex.

  ```
  python snp_overlap_runner.py Heatmap
  --peak_file <path to peak desc file>
  --heatmap_folder <path to a heatmap folder>
  ```

- show_plot

  This **optional** argument tells the program where to interactively show each figure. This argument only works if you can display window panes. Since this manual assumes cluster usage, a graphical user interface won't be provided and as a result feel free to ignore this argument.

  Ex.

  ```
  python snp_overlap_runner.py Heatmap
  --peak_file <path to peak desc file>
  --show_plot
  ```

- locus

  Sometimes the user may not want to print out every peak result at once. That's where this **optional** argument comes into play. This argument tells the program to only be focused on this specific locus (case sensitive) and ignore the others.

  Ex

  ```
  python snp_overlap_runner.py Heatmap
  --peak_file <path to peak desc file>
  --locus TCF7L2
  ```

- tissue

  Sometimes the user may not want to print out every peak result at once. That's where this **optional** argument comes into play. This argument tells the program to only be focused on this specific tissue (case sensitive) and ignore the others.

  Ex

  ```
  python snp_overlap_runner.py Heatmap
  --peak_file <path to peak desc file>
  --tissue Fat
  ```

- cube

  This **optional** argument tells the program to generate the 3D figure instead of deafulting to 2D. Again this is an experimental figure, there could be bugs/crashes within the program.

  Ex

```
python snp_overlap_runner.py Heatmap
--peak_file <path to peak desc file>
--cube
```

- color

  This **optional** argument tells the program where it can look to find the color scheme file. This file will be a little complicated to explain, but is fundamental towards this whole command. The default file is called `color_scheme.csv` and it needs to be in the same directory where the python commands were executed. The color scheme file only needs two headers which are Colors and Rank. The colors field contains the html codes for each color and the rank field ranks each color in terms of their corresponding score. HTML ref here Remember the score field in the peak desc file? If not please take time to jump to that section here: peak desc file. Now the number of colors needed for the heatmap completely depends on the min and max of score field. For example if the min and max for a given figure is -3 to 3, you will need 7 colors. (one for $-3 - 3$ including 0). Also do note the min and max are determined by the highest number found. So if the you unintentially cause a -4 or a 4 to appear, then the new max and min will be those mentioned numbers. It's annoying I know, but that's the current implementation of the heatmap. Once everything is figured out, all thats left is to run the program.

  Ex

```
python snp_overlap_runner.py Heatmap
--peak_file <path to peak desc file>
--color <path to color scheme file>
```

  Ex File.

Colors	Color Names	Rank
#FF0000	Red	0
#FF5200	Red-Orange	1
#FFa500	Orange	2
#FFFFFF	White	3
#FFFF00	Yellow	4
#60bf00	Yellow-Green	5
#008000	Green	6

  Note the color name field is optional. It is added here so the user knows what the html code corresponds to. If you are an expert at color codes, then feel free to ignore that field. Since this component doesn't seem straight forward, here is an explanation. Baseed on the peak desc file, the max and min values are −3 and 3. The corresponding color for −3 should be red (to signify bad) which is why it gets the 0 rank. Basically, a rank of 0 corresponds with the min value. White is neutral ground so it should match with the middle of the color coordinates, which is why it gets the median rank between 0 and 6 (0 − 6 because only seven possible vaues). Green signifies a good result so it should correspond to the max value which is why it gets the 6 rank. The other colors fall into place.

Figures Example

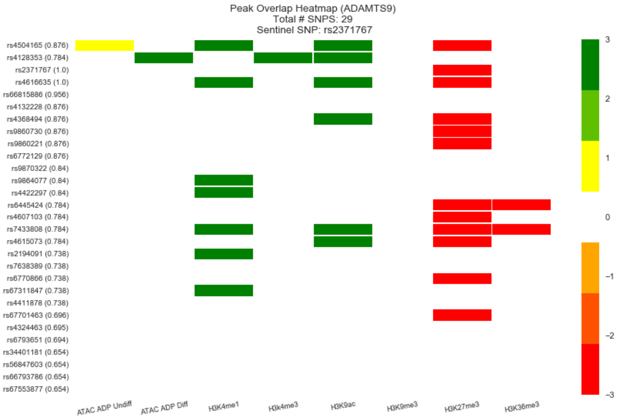

Figure 1: An example of the 2D file. The snps that overlap a peak are on the y-axis. The color corresponds to the amount of peaks that a snp overlaps. The x-axis contains the unique labels defined in the peak description file.

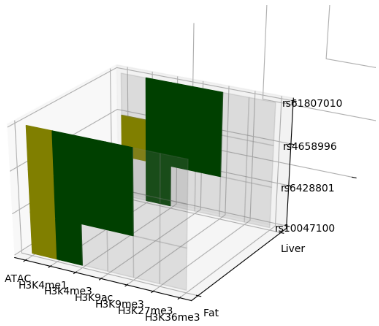

Figure 2: An example of the 3D file. The snps that overlap a peak are on the z-axis. The color corresponds to the amount of peaks that a snp overlaps. The x-axis contains the unique labels defined in the peak description file. The y-axis contins the different tissue types. Still in construction, but will be definitely useful as more data gets added.