Abstract

Searchlight 2 automates the data exploration and visualisation stage of RNA-seq differential analysis as far as its possible, by assuming that most experiments can be simplified into a combination of pre-set workflows, based on the experimental design. In doing so it has the potential to save days, weeks and even months worth of labour and associated cost per RNA-seq experiment - at no loss to specificity. It provides a comprehensive, yet breadth and depth curated selection of results including intermediate files, statistical analysis, extensive visualisation, simple and modifiable r-code, a Shiny app and fully realised reports. It is compatible with any annotated organism and any experimental design regardless of complexity. Searchlight 2 is easy to setup and use, has minimal requirements, and typically runs in under 5 minutes per workflow. We envisage Searchlight 2 to be of use to a wide range of RNA-seq users. Namely project bioinformaticians, RNA-seq service providers and to bench scientists with a cursory understanding of RNA-seq data analysis. Searchlight 2 is available at: https://github.com/Searchlight2/Searchlight2 and as a galaxy module.

Citation

John J. Cole, Bekir Faydaci, Anargyros Megalios, Robin Shaw, Neil Robertson & Carl S. Goodyear. Searchlight 2: rapid and comprehensive RNA-seq data exploration and communication for unlimited differential datasets. 2018-2019. https://github.com/Searchlight2/Searchlight2

Table of Contents

Download and Setup

Searchlight 2 can be dowloaded from https://github.com/Searchlight2/Searchlight2. It requires no installation though requires both **python 2.7** and **R > 3.0** to be installed on your computer. Note: R must be in your path, and an R script executable from the command line with the command "Rscript". This is the default with most installations of R. Searchlight 2 requires the following python libraries to be installed on your computer: **numpy**, **scipy**. To generate visualisations it so requires the following R libraries to be installed on your computer: **ggplot2**, **reshape**, **amap**, **grid**, **gridExtra**, **gtable**, **ggally**, **network**, **sna.** These can all be installed directly from CRAN using the typical R command e.g. install.packages("ggplot2"). To run the Shiny app (for modifying visualisations once generated via a GUI) the following libraries are also required: **shiny**, **shinyFiles**, **fs**, **shinycssloaders**, **graphics**, **dplyr**. We additionally recomend installing Rstudio.

Usage

Input Files

Searchlight accepts input files that are typical to RNA-seq (see the list of file formats for details). Namely a matrix of normalised expression values, a sample sheet, a transcriptome background file and tables of differential expression values (fold, p, adjusted p).

Quick Usage – Example Data

An example dataset is included in the Searchlight2 folder (example_data). The dataset is of mouse CD103+ and CD11b- dendritic cells taken from the Lamina Propria (LP), Mesenteric Lymph (ML) and Mesenteric Lymph Node (MLN). I.e. it measures transcriptomics changes as dendritic cells migrate from the gut through the lymph node into the lymph duct. To explore this dataset run (**as a single line, using full file and folder paths**):

```
python software/searchlight2.py
--out path=example_data/results/
--ss file=example_data/sample_sheet.tsv
--bg file=databases/background/mouse/Ensembl.GRCm38.p6.tsv
--normexp file=example_data/normexp.tsv
--pde file=example_data/ML_vs_LP,numerator=ML,denominator=LP
--pde file=example_data/MLN_vs_ML,numerator=MLN,denominator=ML
--mpde name=all,numerator=ML*denominator=LP,numerator=MLN*denominator=ML
```

This will run one normalised expression workflow, two differential expression workflows and one multiple differential expression workflow. The differential expression workflows will explore LP vs ML and ML vs MLN individually, whilst the multiple differential expression workflow will explore the interactions between all three groups directly.

Parameters and Sub Parameters

Searchlight has several parameters. Most have sub-parameters. Sub parameters must be separated by a comma with no spaces. Sub-parameters have the format parameter=value. There should be no space between the parameter, equals sign and value. Searchlight 2 will provide extensive instructions should it identify issues with any inputs. All input files are tab delimited. All file paths must be full paths.

Core Parameters

There are four core parameters that must be included in every run. These are:

--out (the folder to save the results to)
--ss (the sample sheet for the experiment)
--bg (the background file for the genome / transcriptome version)
--normexp (the matrix of normalised expression values for the experiment)

Running the core parameters will execute the default normalised expression workflow, which analyses the expression data specifically giving an overview of the (transcriptional) relationship between each sample group, and the behaviour of the most highly expressed genes. It produces 5 different types of intermediate or useful files and 8 different types of analysis and visualisation. For example:

```
python software/searchlight2.py
--out path=example_data/results/
--ss file=example_data/sample_sheet.tsv
--bg file=databases/background/mouse/Ensembl.GRCm38.p6.tsv
--normexp file=example_data/normexp.tsv
```

Please note:

- Descriptions of each input file type are provided in the file formats section.
- Searchlight is compatible with normalised expression matrices generated using any method (e.g. FPKM, TPM, RLog, DESeq2, EdgeR, etc).
- For convenience a range of pre-made background files can be found in the “databases” folder. Searchlight has no restrictions as to organism. For ogranisms not included in the download, background files can easily be generated by the user from e.g. Ensembls Biomart. See file formats for more details.
- The gene IDs in the background file must be of the same type as those in the normexp file. Only genes that are in both files, with a matching ID will be included in the downstream analysis. The order that genes appear in both files does not have to be the same.
- The sample names in the matrix of normalised expression values header line must be identical to that of the sampes column in the sample sheet. All samples must match. The order that samples appear in both files does not have to be the same.

Differential Expression Workflow

The differential expression workflow is the core workflow for exploring the differences in expression between two groups of samples. It provides analysis of the size and consistency of differences at the global and single gene level, the chromosomal distribution of differences, over-represented gene-sets (e.g. GO, KEGG, etc) and upstream regulators (e.g. TRRUST). It generates 42 different types of intermediate, statistical or useful files and 31 different types of analysis and visualisation. It is also suitable for use with comparisons generated using complex linear models (e.g. WT vs KO + age + gender + BMI).

Running a differential expression workflow requires the user to supply a table of differential expression values (see “List of Input Files” for more details), alongside the names of the sample groups that are

the numerator and denominator. Searchlight 2 is compatible with differential expression data generated using any method (DESeq2, EdgeR, etc). The Numerator is the sample group where a positive fold change indicates a gain in expression. The Denominator is the sample group where a negative fold change indicates a gain in expression. Both the numerator and denominator sample groups must be in the sample sheet. To run the differential expression workflow use (alongside the core parameters):

```
python software/searchlight2.py
--out path=example_data/results/
--ss file=example_data/sample_sheet.tsv
--bg file=databases/background/mouse/Ensembl.GRCm38.p6.tsv
--normexp file=example_data/normexp.tsv
--pde file=example_data/ML_vs_LP,numerator=ML,denominator=LP
```

This workflow may be included unlimited times per run with any combination of sample groups – provided the user supplies a table of differential expression values. A seperate set of results will be generated for each. The gene IDs in the differential expression file must be the same as those in the background and normexp files. Only genes that are in all three files, with a matching ID will be included in the downstream differential analysis.

Multiple Differential Expression Workflow

The multiple differential expression workflow is recommended where it is desirable to explore the interactions between three or more groups of samples directly and simultaneously. Such as in a time course or a healthy control vs disease vs disease plus treatment. The multiple differential expression workflow explores the expression, gene and fold change overlap between sets of differentially expressed genes and generates and analyses differential expression signatures with heatmaps, meta-plots and over-representation analysis. It creates 26 different types of intermediate, statistical or useful files and 9 different types of analysis and visualisation. It requires the user to have supplied at least two differential expression workflows and to specify which differential comparisons to compare to each other. This is echieved by adding denomonator=my_group_A*numerator=my_group_B the following sub-parameter multiple times as part of the –MPDE parameter. The numerator*denominator combo must match that of an existing –pde command. E.g.

```
python software/searchlight2.py
--out path=example_data/results/
--ss file=example_data/sample_sheet.tsv
--bg file=databases/background/mouse/Ensembl.GRCm38.p6.tsv
--normexp file=example_data/normexp.tsv
--pde file=example_data/ML_vs_LP,numerator=ML,denominator=LP
--pde file=example_data/MLN_vs_ML,numerator=MLN,denominator=ML
--mpde name=all,numerator=ML*denominator=LP,numerator=MLN*denominator=ML
```

This workflow may be included unlimited times per run with any combination of differential expression comparisons or sample groups – provided the user supplies a differential expression workflow for each comparison.

Over-representation and Upstream Regulator Analysis

To include the over-representation (hypergeometric) and upstream regulator analysis the --hgsea and --ureg parameters must be supplied, alongside a path to a suitable gene-set database (such as GO and KEGG) or upstream regulator database (such as TRRUST). For convenience several are supplied with th software (databases folder). E.g.

```
python software/searchlight2.py
--out path=example_data/results/
--ss file=example_data/sample_sheet.tsv
--bg file=databases/background/mouse/Ensembl.GRCm38.p6.tsv
--normexp file=example_data/normexp.tsv
--pde file=example_data/ML_vs_LP,numerator=ML,denominator=LP
--ureg file=databases/gene_sets/mouse/go_bp,name=TRRUST
--hgsea file=databases/ureg/mouse/trrust.mouse.tsv,name=GO_BP
```

The --hgsea and --ureg parameters can be supplied unlimited times per run, so long as each specifies a different database. Seperate analysis for each will be performed.

Changing Default Cut-Offs

Most significance and other thresholds can be explicitly set via sub-parameters. The default for differential expression is adjusted $p < 0.05$ and absolute log2fold > 1. For more details see the list of options section.

The Order Sub-Parameter

The order sub-parameter is a highly useful visualisation feature, and is used alongside differential expression or multiple differential expression workflows. It allows the user to specify the order that sample groups appear on visualisations and legends. It also allows users to specify any additional sample groups to be added to visualisations. To use simply add order=my_group_1+ my_group_2,etc. Specifying any groups you wish to include in the order that you wish them to appear.

```
python software/searchlight2.py
--out path=example_data/results/
--ss file=example_data/sample_sheet.tsv
--bg file=databases/background/mouse/Ensembl.GRCm38.p6.tsv
--normexp file=example_data/normexp.tsv
--pde file=example_data/ML_vs_LP,numerator=ML,denominator=LP,order=LP+ML
--pde file=example_data/MLN_vs_ML,numerator=MLN,denominator=ML,order=LP+ML+MLN
```

Customising Aesthetics

Searchlight 2 provedes several means for visualisations to be modified, to e.g. make them aestheticly consistent with existing or further work, or simply to the users personal taste.

Customising the Default Aesthetics

Firstly, users may customise Searchlight 2 to produce images of their own visual flavour as default. All Searchlight 2 images are generated using R script. These scripts are dynamically parsed from a central bin of R code “snippets”, located at searchlight2/software/bin/r. Each snippet contains well commented and easy to understand code for a small section of the final R scripts. By modifying these snippets the default behaviour of Searchlight can be changed. For example:

r/default_aesthetic/default_three_tone_heatmap_colours.txt

```
##---- Default Three Tone Heatmap Colours  ----##
default_three_tone_heatmap_colours = c("blue","pink","red")
```

to

```
##---- Default Three Tone Heatmap Colours  ----##
default_three_tone_heatmap_colours = c("blue","black","yellow")
```

Would change the default expression heatmap colours from blue / pink / red to blue / black / yellow.

R Scripts – Per Plot

Secondly, users may cutomise images once they have been generated. Searchlight 2 generates an R script for each type of analysis and plot. It is these R scripts that allow Searchlight to generate visualisations. By modifying these scripts (and directly re-running them) the user can modify any aspect of the plot aesthetics for any plot. It is worth noting that these scripts are designed to be easy to understand and modify. They all have the same layout and use many shared features. Many common visual features (e.g. font type, font size, colours, labels, etc) are entered as named parameters. The per plot R scripts are located next to the plots in the plots folder.

R Scripts – Per Workflow

In addition to per plot R scripts Searchlight 2 also generates a script for each workflow. Executing this script will re-generate all images in the workflow. Thus by modifying aethetic parameters in this Script users can change all plots simultaneously. This is particualrly useful when changing the theme (e.g. fonts, grids, border, etc) or sample group colours and labels. The per workflow R scripts are located the plots folder for each workflow.

Shiny (Graphical Interface)

Finally, an extensive range of visual features for all images can be modified via a Shiny GUI. This is recommended for users with limited R experience. Upon execution Searchlight 2 generates a bespoke R Shiny GUI and places it into the output folder (base). By either opening the “server.r” file in Rstudio and running the app or by opening R in the terminal and running:

```
library("shiny")
runApp(appDir = "/path/to/Shiny")
```

The app will open in the users default web browser. The user can then navigate the various workflows and images, modify them and save as png, svg or jpeg.

List of File Formats

Sample Sheet. The sample sheet is simply a list of each sample (first column) in the experiment and any number of extra columns stating sample group. Typically, there will be at least one sample group column. The file requires a header, and the first header entry must be "SAMPLE". Sample names and sample groups must include only the characters a-z, A-Z, 0-9 and _ (underscore). They also cannot start with a number (though they may include them). Sample names must be unique. An example of this file can be found in example_data/sample_sheet.tsv. E.g.

SAMPLE	SAMPLE_GROUP
gut_r1	gut
gut_r2	gut
gut_r3	gut
node_r1	node
node_r2	node
node_r3	node

Background File. The background file lists all genes in the transcriptome and contains the following information for each gene: gene ID (such as Ensembl), gene symbol, biotype (such as coding gene, lincRNA), chromosome, gene start coordinate, gene stop coordinate. The file must have a header row with the following headers in this order: id, symbol, biotype, chromosome, start, stop. If you wish to ignore the Searchlight 2 biotype function (faster) please set all biotypes to "gene". All gene IDs must be unique, but symbols do not have to be. For convenience background files for several common organisms and transcriptome versions are provided in databases/background/. Alternatively, these can be downloaded from Ensembls Biomart.

ID	SYMBOL	BIOTYPE	CHROMOSOME	START	STOP
ENSG00000148677	ANKRD1	protein_coding	10	92671853	92681033
ENSG00000187908	DMBT1	protein_coding	10	124400252	124403252
ENSG00000185324	CDK10	protein_coding	16	89747145	89762772

Matrix of Normalised Expression Values. A typical matrix of normalised expression values. This can include any normalisation type (e.g. FPKM, CPM, DESeq2, Rlog, etc). The file must include a header row, with the first column as "ID" and the rest of the columns as sample names. The sample names in the header row must match exactly the sample names in the sample sheet. They do not have to be in the same order but all the sample names in the sample sheet must also be in the matrix of normalised expression values and vice versa. The gene IDs used must be the same type of ID as those found in the background file. To be included in Searchlight 2 analysis a gene must be in both this file (if supplied) and the background file. Genes that are present in one but not the other will be ignored. An example of this file can be found here: example_data/normexp.tsv.

ID	gut_r1	gut_r2	gut_r3	node_r1	node_r2	node_r3
ENSG00000148677	1.84	1.92	1.07	1.04	1.76	0.39
ENSG00000187908	79.04	85.19	76.16	23.24	21.59	43.6
ENSG00000185324	19.12	12.89	22.78	75.73	76.65	56.01

Differential Expression Table. A typical differential expression file. This can be produced by any differential expression tool (such as DESeq2, EdgeR, Cuffdiff, etc), provided it supplies the following information: gene ID, log2fold change, a p-value and some kind of adjusted p-value. The differential expression file must have a header row with the following headers in this order: ID, log2fold, p, p.adj. The gene IDs used must be the same type of ID as those found in the background file. Genes in this file that do not match a gene in the background file AND count matrix or normalised expression matrix (whichever was used) will be excluded. You should supply one of these files for each comparison that you wish to explore. An example of this file can be found here: example_data/LP_vs_ML.tsv.

ID	LOG2FOLD	P	P.ADJ
ENSG00000148677	0.598	0.0023	0.89
ENSG00000187908	1.443	0.00056	0.046
ENSG00000185324	-1.927	0.0000041	0.00023

Gene-set database. A typical gene-set database file in the Gene Matrix Transposed (GMT) format (https://software.broadinstitute.org/cancer/software/gsea/wiki/index.php/Data_formats#GMT:_Gene_Matrix_Transposed_file_format_.28.2A.gmt.29). This file should not contain a header line. The first column should include the gene set name (or ID), and the second column the source of the gene set (e.g. GO, String, etc). All subsequent columns should contain the genes in the gene set. **Note: the genes must be listed as gene symbols and not IDs.** Only gene symbols which are also in the background file will be included in the analysis. Gene sets should be listed by row. An example of this file type can be found here (obtained from GO): databases/gene_sets/

Upstream Regulator Database. The upstream regulator file lists known gene regulatory interactions. This file should not contain a header line. The first column should contain a regulator gene, the second a gene that the regulator regulates, and the third column the type of interaction. Valid interaction types are "Activation", "Repression", and "Unknown". Each row should list one interaction. There is no limit to the number of different lines that a regulator or regulated gene can appear on. **Note: the genes must be listed as gene symbols and not IDs.** An example of this file type can be found here (obtained from TRRUST): databases/gene_sets/databases/ureg/.

Additional Annotations File. A file of user defined additional annotation. The annotations file must have a header row with the first column as “ID”. The IDs in this column must be of the same type as those in the background file and normalised expression matrix. Any number of additional columns may be added, provided that they do not have incomplete cells.

List of Options

Most Searchlight 2 parameters include several sub-parameters these must be supplied as a comma separated string with no spaces. Sub-parameters listed with an asterisk next to them are optional and unless supplied will be set to the default value.

Annotations File. Specifies any additional gene annotations, supplied as an annotation file. This parameter is optional.

```
--anno file=path/to/annotations_file
```

Sub-parameter	description
file=	Full path to the annotation file

Background File. Specifies the background file to be used. This is a required parameter.

```
--bg file=path/to/background_file
```

Sub-parameter	description
file=	Full path to background file

Differential Expression Workflow. Runs a differential expression workflow (PDE) for a supplied DE file and sub-parameters. This option can be supplied more than once and should be supplied for each PDE file that you wish to run a PDE workflow on.

```
--pde file=path/to/pde_file,numerator=sample_group1,denominator=sample_group2,
p.adj=0.05,log2fold=1,order=sample_group1+sample_group2,gl=NONE
```

Sub-parameter	description
file=	Full path to pairwise differential expression file
numerator=	Name of the sample group that will show higher expression with a positive fold change. The sample group must match a group from the sample sheet.
denominator=	Name of the sample group that will show lower expression with a positive fold change. The sample group must match a group from the sample sheet.
p.adj=	* Adjusted p value threshold for differential gene significance [number][default 0.05]
log2fold=	* Absolute lo2fold enrichment threshold for differential gene significance [number][default 1]
order=	* + separated list of sample groups to plot in the order that you wish them to be plotted. Each sample group must match a group from the sample sheet. [+ separated string][default numerator+denominator]
gl=	* full path to a file containing a list of gene IDs. If supplied all genes not in this list are excluded from the analysis.

Ignore normalised expression workflow. Ignores the normalised expression workflow.

```
--ignore_normexp T
```

Ignore differential expression workflows. Ignores differential expression workflows.

```
--ignore_pde T
```

Ignore multiple differential expression workflows. Ignores multiple differential expression workflows.

```
--ignore_mpde T
```

Mulitple Differential Expression Workflow. Runs a multiple pairwise differential expression workflow (MPDE) for a series of PDE files. This option can be supplied more than once, and should be supplied for each combination of PDE file that you wish to run a MPDE workflow on.

```
--mpde name=name,numerator=sample_group_1*denominator=sample_group_2,order,scc=0.75,
order=sample_group_1+sample_group_2
```

Sub-parameter	description
name=	Name tag to give this MPDE. Can be any single word and is used only for identification.
numerator=*denominator=	Comma separated list of the PDEs to be included in the MPDE. Each PDE in this list must also be supplied as a separate PDE workflow. Each PDE is referenced in this list by stating the numerator and denominator in the following format: numerator=sample_group1denominator=sample_group2. E.g. the list might look like: numerator=sample_group1*denominator=sample_group2,numerator=sample_group3*denominator=sample_group4 There is no limit to the number of PDEs that can be supplied to a MPDE.
order=	* + separated list of sample groups to plot in the order that you wish them to be plotted. Each sample group must match a group from the sample sheet. [+ separated string][default all unique sample groups in MPDE in sample sheet order]
gl=	* Full path to a file containing a list of gene IDs. If supplied all genes not in this list are excluded from the analysis.
scc=	* Threshold used for merging the various differential expression profiles into differential expression signatures. Two profiles that have a Spearman Correlation Coefficient above this value will be merged. [default = 0.75]

Normalised Expression Matrix. Specifies the normalised expression file to be used. This is a required parameter.

```
--normexp file=path/to/noremxp_file,expressed=1,type=expression
```

Sub-parameter	description
file=	Full path to normalised expression matrix
expressed=	* Minimum value for a gene to be considered expressed in the spatial analysis. This is not used in any other stage of the SL2 software. [number][default=1]
type=	* Type of expression value, only used for the report text [any_single_word][default=expression]

Out Folder. Specifies the folder where results should be saved to. This is a required parameter.

```
--out path=output/folder/path
```

Sub-parameter	description
path=	Full path to SL2 output folder

Over-Representation Analysis. Specifies the gene-set file and settings for a hypergeometric gene set analysis to be performed for each relevant workflow. This is parameter is optional, and this can be included several times each with a different gene_set file, if desired.

```
--hgsea file=path/to/gene_set_file,type=GO,p.adj=0.05,log2fold=1,min_set_size=5,
max_set_size=250,network_overlap_ratio=0.5,network_overlap_size=5
```

Sub-parameter	description
file=	Full path to gene set file
type=	Database name, for naming purposes only
p.adj=	* Adjusted p value threshold for gene set significance [number][default 0.05]
log2fold=	* Absolute log2fold enrichment threshold for gene set significance [number][default 0]
min_set_size=	* To be included in the hypergeometric enrichment analysis step a gene set must have at least this many genes in it [number][default 5]
max_set_size=	* To be included in the hypergeometric enrichment analysis step a gene set must have less than this many genes in it [number][default 250]
network_overlap_ratio=	* All network edges must have at least this overlap ratio using the Szymkiewicz-Simpson coefficient [number][default 0.5]
network_overlap_ratio=	* All network edges must have at least this overlap size (number of genes) [number][default 5]

Sample Sheet. Specifies the sample sheet to be used. This is a required parameter.

```
--ss file=path/to/sample_sheet
```

Sub-parameter	description
file=	Full path to sample sheet

Upstream Regulator Analysis. Specifies the upstream regulator file and settings for upstream regulator analysis to be performed for each relevant workflow. This is parameter is optional, and this can be included several times each with a different upstream regulator file, if desired.

```
-- ureg file=path/to/ureg_file,type=trusst,zscore=2,p.adj=0.05,log2fold=1,min_set_size=5,
max_set_size=250,network_overlap_ratio=0.25,network_overlap_size=5
```

Sub-parameter	description
file=	Full path to the upstream regulator file
type=	database name, for naming purposes only
Zscore=	* Z-score threshold for upstream regulator activation [number][default 2]
p.adj=	* Adjusted p value threshold for gene set significance [number][default 0.05]
log2fold=	* Absolute log2fold enrichment threshold for gene set significance [number][default 0]
min_set_size=	* To be included in the upstream regulator analysis step a regulator must regulate at least this many genes [number][default 5]
max_set_size=	* To be included in the upstream regulator analysis step a regulator must regulate a maximum of this many genes [number][default 250]
network_overlap_ratio=	* All network edges must have at least this overlap ratio using the Szymkiewicz-Simpson coefficient [number][default 0.5]
network_overlap_ratio=	* All network edges must have at least this overlap size (number of genes) [number][default 5]

FAQs

Does Searchlight 2 check my input data? Yes. Searchlight 2 thoroughly checks the integrity and format of all input data. Files which are in the wrong format (for example don't include numbers where they should, or the correct headers) are reported. In addition, all input parameters are checked for format.

I am interested in exploring coding genes separately from non-coding genes. Can I do this? Yes, Searchlight 2 includes gene-types in the background file format, and automatically runs a separate analysis for each type. Simply supply a background file with the appropriate gene-types (see the list of input file formats for more information).

I have used linear modeling with interaction terms for my differential expression. Which workflow does this fall under? Though the interaction terms have been included in your model it is still a pairwise differential expression analysis (PDE). Such models simply consider a pairwise interaction taking into account the effect of the interaction terms. It can however be useful to visualise the interaction terms alongside the differential results. This can be achieved using the --order= sub-parameter

I ran Searchlight 2, but I can't see any plots. What's going on? Most likely the root cause is an unforeseen bug in one of the plots in the R script. This will have a knock-on effect. If you are missing plots please try running the R script for the plots in question (e.g. plots/workflow.r) in R and investigating the error directly. If you cannot fix this easily, please report it as an issue on the github page.

My computer is not connected to the web, does this matter? No. Searchlight 2 does not connect to anything external whatsoever, at any point. This is a deliberate design feature.

I want to use p values instead of adjusted p values. Can I do this? Yes, Searchlight 2 does not reference the web in any way, nor genome or transcriptome files not specified by the user. This is a deliberate design feature to allow the user control. If you wish to use P values instead of adjusted P simply "munge" your file. I.e. replace the adjusted p values with the p values in the differential expression file. The format of each file is fixed – to make sure the user understands what they are doing, but its up to the user what they put in it.

I am using a custom background file, and don't have both gene IDs and gene symbols, nor gene biotypes what can I do? As above though the format of each file is fixed it is up to the user what goes in. Try simply using the same values for IDs and symbols. Provided it matches the expression matrix and any differential expression files it doesn't actually matter what is in these columns. So long as you are comfortable with what you are inputting. The same logic applies to biotypes, simply set them all to "gene".

I have an issue that is not covered by the manual or FAQ. Don't we all? We would be extremely grateful if you could post it on the issues section of the Searchlight 2 github page. Though we are a small team we will try to respond ASAP.