geneXtendeR

Bohdan B. Khomtchouk

2016-05-13

geneXtendeR is designed to optimally annotate a histone modification ChIP-seq peak input file with functionally important genomic features (e.g., genes associated with peaks) based on optimization calculations. These optimization calculations automatically factor in experimental conditions such as the broadness of the histone peaks found in the specific tissue of the ChIP-seq peak file.

To accomplish this level of custom-tailored data-centric analysis, geneXtendeR first optimally extends the boundaries of every gene in a genome by some genomic distance (in DNA base pairs) for the purpose of flexibly incorporating cis-regulatory elements, such as enhancers and promoters, as well as downstream elements that are important to the function of the gene relative to an epigenetic histone modification ChIP-seq dataset. This action effectively transforms genes into "gene-spheres", a new term meant to emphasize the 3D-nature of heterochromatin. A gene-sphere is composed of cis-regulatory elements (e.g., proximal promoters +/- 3kb from TSS), distal regulatory elements (e.g., enhancers), transcription start/end sites (TSS/TES), exons, introns, and downstream elements of a gene. As such, geneXtender maximizes the signal-to-noise ratio of locating gene regions closest to and directly under peaks. By performing a computational expansion of this nature, ChIP-seq reads that would initially not map strictly to a specific gene can now be optimally mapped to the regulatory regions of the gene, thereby implicating the gene as a potential candidate, and thereby making the ChIP-seq experiment more successful. Such an approach becomes particularly important when working with epigenetic histone modifications that have inherently broad peaks with a diffuse range of signal enrichment (e.g., H3K9me1, H3K27me3).

A series of diagnostic criteria are used to compute optimal gene extensions tailored to the tissue-specific broadness of the specific epigenetic mark in the ChIP-seq peak input file:

First, install the geneXtendeR R package via `install.packages("geneXtendeR")` and load it in:

```
library(geneXtendeR)
```

```
## Loading required package: gtf
```

This automatically loads the `gtf` R package, which is a prerequisite data package containing the gene transfer format files of commonly studied model organisms in ChIP-seq analyses. Refer to the `gtf` R package for more details.

geneXtendeR also requires the installation of an external program called `bedtools`. This program must be pre-installed on your computer prior to using geneXtendeR. As described in the `README` file found in the top-level directory of this geneXtendeR R package, detailed installation instructions can be accessed here: http://bedtools.readthedocs.io/en/latest/content/installation.html. After `bedtools` has been installed, the geneXtendeR package is fully configured and setup for use.

First, create a series of geneXtendeR files at some user-specified interval:

```
generate(rat, 1000, 10000, 500)
```

This command generates 19 individual whole-genome files: 100, 1500, 2000, . . . , and 10000 bp upstream extension files for the rat (*Rattus norvegicus*) genome, each having an automatic 500 bp downstream extension. See `species()` for a list of available genomes from the `gtf` R package.

Next, the user must input their peak data from some peak caller (e.g., SICER, MACS2, etc). The peak data must contain only three tab-delimited columns: chromosome number, peak start, and peak end. See

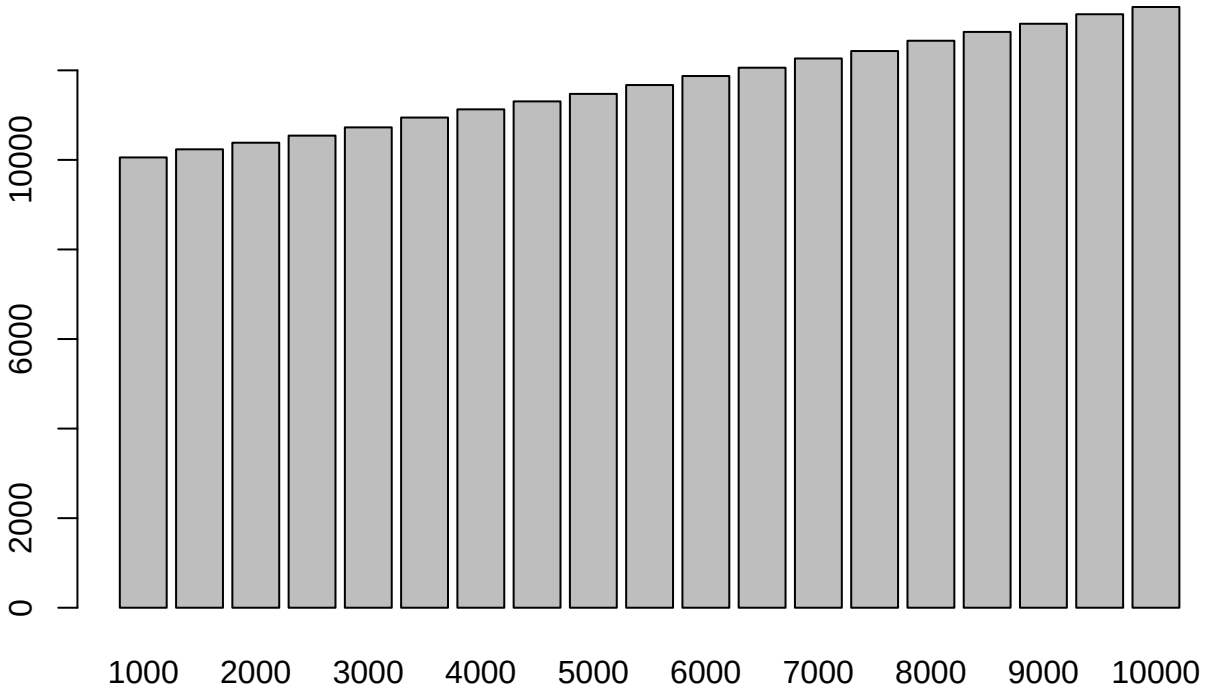
`?sample_peaks_input` for an example. Once the peak input data (e.g., "my_filename.txt") has been assembled properly, it must be properly formatted prior to the execution of various geneXtendeR analyses:

```
peaksInput("my_filename.txt")
```

This command properly formats the user's peak file in preparation for subsequent analyses. For sake of reproducibility, "my_filename.txt" is provided in the `/vignettes` directory of the geneXtendeR package to allow for an interactive session with the commands introduced in the rest of this vignette.

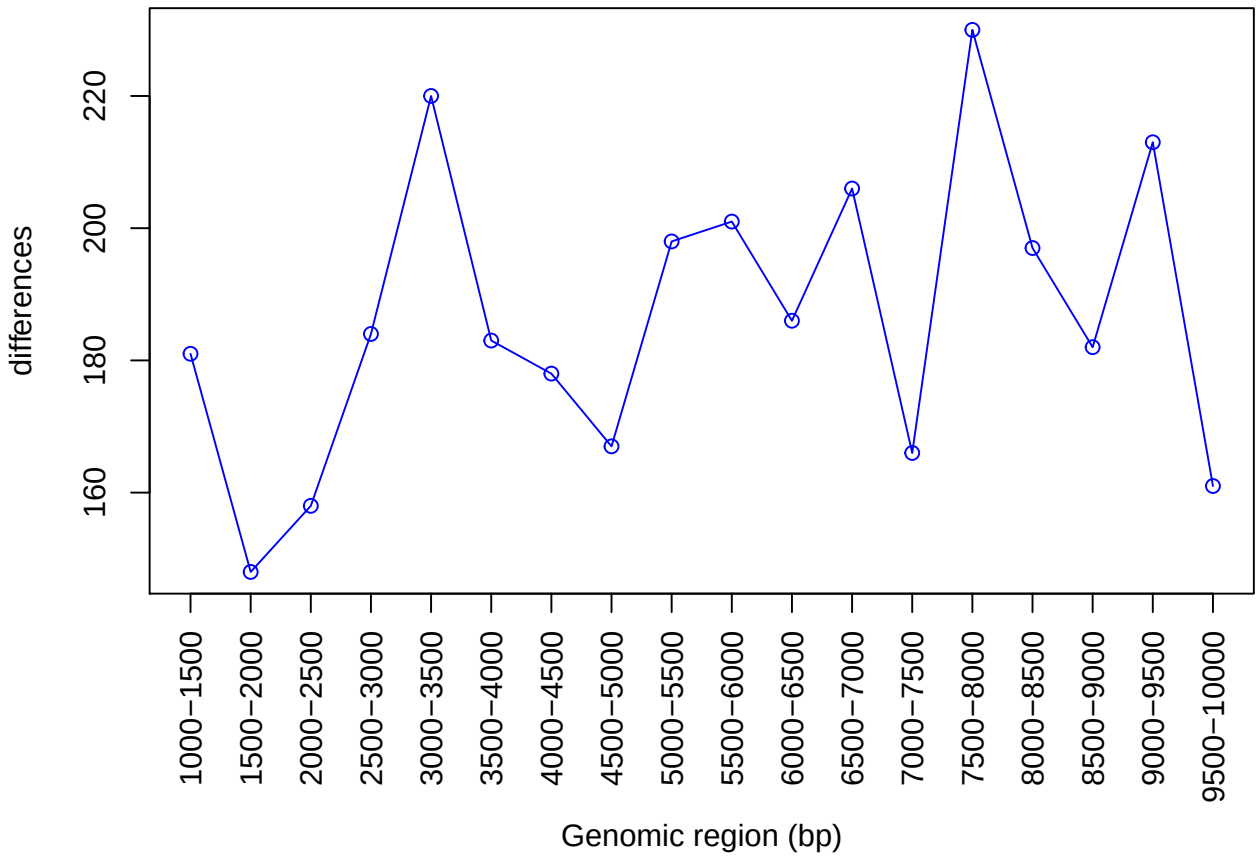
First, a raw count of the number of peaks that are sitting on top of a gene is calculated for each file.

```
barChart()
```



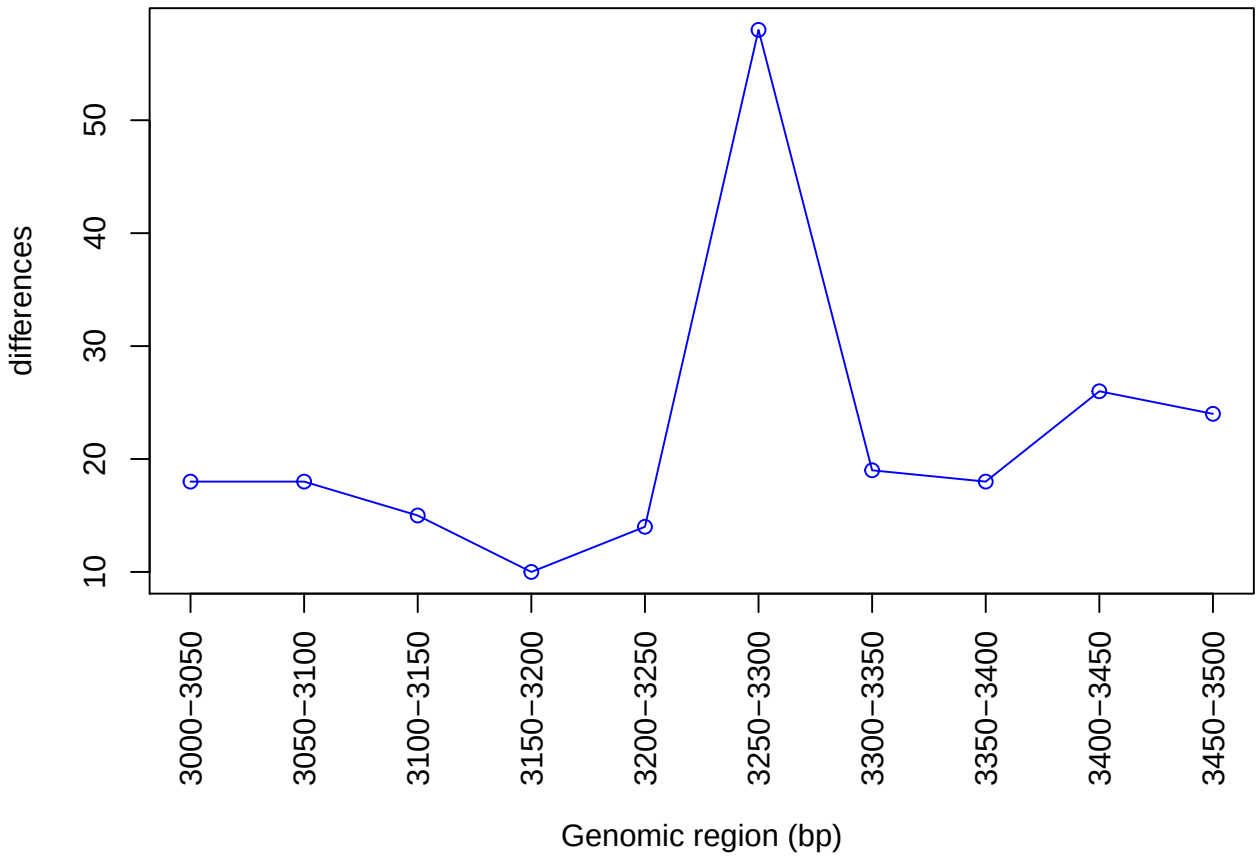
Clearly, the wider the gene-sphere, the more peaks-on-top-of-genes are found throughout the genome. However, the law of diminishing returns begins to kick in at increasing upstream extension levels (see `linePlot()` for a visual representation):

```
linePlot()
```



Clearly, there is a sharp rise in the number of peaks-on-top-of-genes from a 2000 bp upstream extension to a 3500 bp upstream extension. This rise is followed by a steady decline at subsequent extension levels followed by some fluctuations characteristic of noise. It may be interesting to investigate what is going on in the interval from 3000 bp to 3500 bp:

```
generate(rat, 3000, 3500, 50)
linePlot()
```



Clearly, there is a relatively sharp spike in the number of peaks-on-top-of-genes at the 3300 bp upstream extension (as compared to the 3250 bp extension). This spike then drops back down and stays approximately constant at subsequent extension levels, exhibiting fluctuating behavior characteristic of biological noise. It is also possible to identify the genes that are unique amongst the 3250 and 3300 bp upstream extension levels:

```
distinct(rat, 3250, 3300)
```

```
##       V1        V2        V3  V4        V5        V6                 V7
##  1:    1 209071800 209072999   1 209072955 209078098 ENSRNOG00000054210
##  2:    2  36973200  36974199   2  36966308  36973214 ENSRNOG00000053154
##  3:    2  80943600  80945399   2  80945358  81143947 ENSRNOG00000048363
##  4:    2 181779800 181782999   2 181782955 181787067 ENSRNOG00000058638
##  5:    3   9041200   9044399   3   9035873   9041242 ENSRNOG00000024846
##  6:    5  48439800  48441599   5  48419802  48439831 ENSRNOG00000007755
##  7:    5 117561400 117561799   5 117557543 117561417 ENSRNOG00000047347
##  8:    5 124292800 124297199   5 124297177 124338553 ENSRNOG00000007639
##  9:    6  28467400  28469199   6  28392516  28467418 ENSRNOG00000012950
## 10:    7  27968200  27970199   7  27970158  27974657 ENSRNOG00000059844
## 11:    8  13771400  13775399   8  13775367  13779286 ENSRNOG00000060932
## 12:    8  50542600  50544399   8  50538142  50542631 ENSRNOG00000059084
## 13:   10  12288000  12289599  10  12283275  12288036 ENSRNOG00000061166
## 14:   11  29703800  29705999  11  29698840  29703823 ENSRNOG00000058541
## 15:   15  18698400  18699599  15  18647344  18698433 ENSRNOG00000008167
## 16:   16  12665600  12666999  16  12658728  12665636 ENSRNOG00000056897
## 17:   16  13978400  13980799  16  13980778  13984684 ENSRNOG00000059118
## 18:  100  73074600  73095199 100  73081434  73090700 ENSRNOG00000051175
```

```
## 19: 100  73560400  73572799 100  73557042  73565346 ENSRNOG00000061703
## 20: 100  73560400  73572799 100  73560431  73564861 ENSRNOG00000051183
## 21: 100  73905800  73906999 100  73900180  73919850 ENSRNOG00000051053
## 22: 100  75172400  75173399 100  75150108  75295238 ENSRNOG00000002790
## 23: 100  77542800  77545199 100  77507314  77562648 ENSRNOG00000002451
## 24: 100 106055600 106056799 100 106048575 106069527 ENSRNOG00000023256
## 25: 100 111310400 111311199 100 111268990 111328486 ENSRNOG00000057622
## 26: 100 111942200 111944799 100 111941602 111946049 ENSRNOG00000060899
## 27: 100 112767800 112768599 100 112766345 112984185 ENSRNOG00000018951
## 28: 100 112769400 112770399 100 112766345 112984185 ENSRNOG00000018951
## 29: 100 112804400 112805599 100 112766345 112984185 ENSRNOG00000018951
## 30: 100 113568600 113570199 100 113547092 113583064 ENSRNOG00000053818
## 31: 100 113572800 113573399 100 113547092 113583064 ENSRNOG00000053818
## 32: 100 114086600 114086799 100 113945354 114110562 ENSRNOG00000012787
## 33: 100 115735800 115736399 100 115627153 115911993 ENSRNOG00000027233
## 34: 100 118161800 118162999 100 118081590 118318539 ENSRNOG00000030877
## 35: 100 118218200 118219599 100 118081590 118318539 ENSRNOG00000030877
## 36: 100 118347800 118348399 100 118347032 118350904 ENSRNOG00000041256
## 37: 100 118481800 118482999 100 118443323 118516361 ENSRNOG00000032973
## 38: 100 122632800 122633599 100 122504074 122690965 ENSRNOG00000013321
## 39: 100 123660600 123663799 100 123602742 123665714 ENSRNOG00000054788
## 40: 100 123660600 123663799 100 123659177 123664597 ENSRNOG00000049392
## 41: 100 123988000 123995399 100 123887488 123996842 ENSRNOG00000040013
## 42: 100 128177400 128178599 100 128161395 128271622 ENSRNOG00000007315
## 43: 100 128177400 128178599 100 128174529 128183851 ENSRNOG00000059506
## 44: 100 139429800 139432999 100 139353655 139468098 ENSRNOG00000002413
## 45: 100 139483400 139485599 100 139480884 139487024 ENSRNOG00000058144
## 46: 100 140884000 140888399 100 140874916 140889295 ENSRNOG00000000861
## 47: 100 142130800 142132799 100 142053154 142251669 ENSRNOG00000042753
## 48: 100 145875200 145876599 100 145875295 145879247 ENSRNOG00000054106
## 49: 100 150661400 150662999 100 150660243 150669609 ENSRNOG00000058521
## 50: 100 152538600 152538999 100 152414115 152645831 ENSRNOG00000056558
## 51: 100 153063200 153064999 100 153060728 153068000 ENSRNOG00000052022
## 52: 100 155341400 155343199 100 155340876 155344793 ENSRNOG00000057982
## 53: 100 157169800 157177399 100 157167644 157175368 ENSRNOG00000055185
## 54: 100 157590200 157591199 100 157588031 157594844 ENSRNOG00000060122
## 55: 100 159818600 159820599 100 159723366 159844372 ENSRNOG00000000869
## 56: 100 159821400 159823199 100 159723366 159844372 ENSRNOG00000000869
##      V1        V2        V3  V4        V5        V6                  V7
##                 V8 V9
##  1: AABR07005961.1  0
##  2: AABR07007980.1  0
##  3:          Dnah5  0
##  4:            7SK  0
##  5:          Ier5l  0
##  6:         Pm20d2  0
##  7: AABR07049336.1  0
##  8:            C8b  0
##  9:          Efr3b  0
## 10: AABR07056515.1  0
## 11:        SNORA55  0
## 12: AABR07073400.1  0
## 13: AABR07029192.1  0
## 14: AABR07033607.1  0
```

```
## 15:          Abhd6  0
## 16: AABR07024722.1  0
## 17:             U6  0
## 18:  Rn60_X_0740.2  0
## 19:        Gm14597  0
## 20:  Rn60_X_0744.7  0
## 21:  Rn60_X_0748.3  0
## 22:          Abcb7  0
## 23:        Fndc3c1  0
## 24:           Nxf7  0
## 25:          Rbm41  0
## 26:        Tsc22d3  0
## 27:         Col4a5  0
## 28:         Col4a5  0
## 29:         Col4a5  0
## 30: AABR07040906.1  0
## 31: AABR07040906.1  0
## 32:        Tmem164  0
## 33:          Trpc5  0
## 34:          Htr2c  0
## 35:          Htr2c  0
## 36: AABR07041066.1  0
## 37:        Il13ra2  0
## 38:         Dock11  0
## 39: AABR07041239.1  0
## 40:         Sowahd  0
## 41:          Rhox2  0
## 42:          Thoc2  0
## 43:     AC124926.2  0
## 44:           Gpc4  0
## 45: AABR07041778.3  0
## 46:           Zic3  0
## 47:          Fgf13  0
## 48:      5_8S_rRNA  0
## 49: AABR07042226.1  0
## 50:         Gabra3  0
## 51:          Pnma3  0
## 52:        Mir3585  0
## 53:          Dusp9  0
## 54: AABR07042465.1  0
## 55:        Arhgef6  0
## 56:        Arhgef6  0
##                V8 V9
```

`V1-V3` denote the chromosome/start/end positions of the peaks, `V4-V6` denote the respective values for the genes, `V7` is the gene ID (e.g., Ensembl ID), `V8` is the gene name, and `V9` is the distance of each respective peak to its nearest gene. Note that the X chromosome is designated by the integer 100, the Y chromosome by the integer 200, and the mitochondrial chromosome by the integer 300. This is done for sorting purposes (see `peaksInput()` for more details). The `distinct()` command finds what peaks-on-top-of-genes would be missed if a 3250 bp upstream extension is used instead of a 3300 bp extension. Of course, subsequent follow-up extensions naturally incorporate additional peaks-on-top-of-genes, since the concept of a gene is being expanded into an ever-widening gene-sphere.

However, even though these dynamics are to be expected, such extensions are unlikely to add significant value to the annotation of the peak file. Taking the example of the 1000-10000 bp line plot, an upstream extension

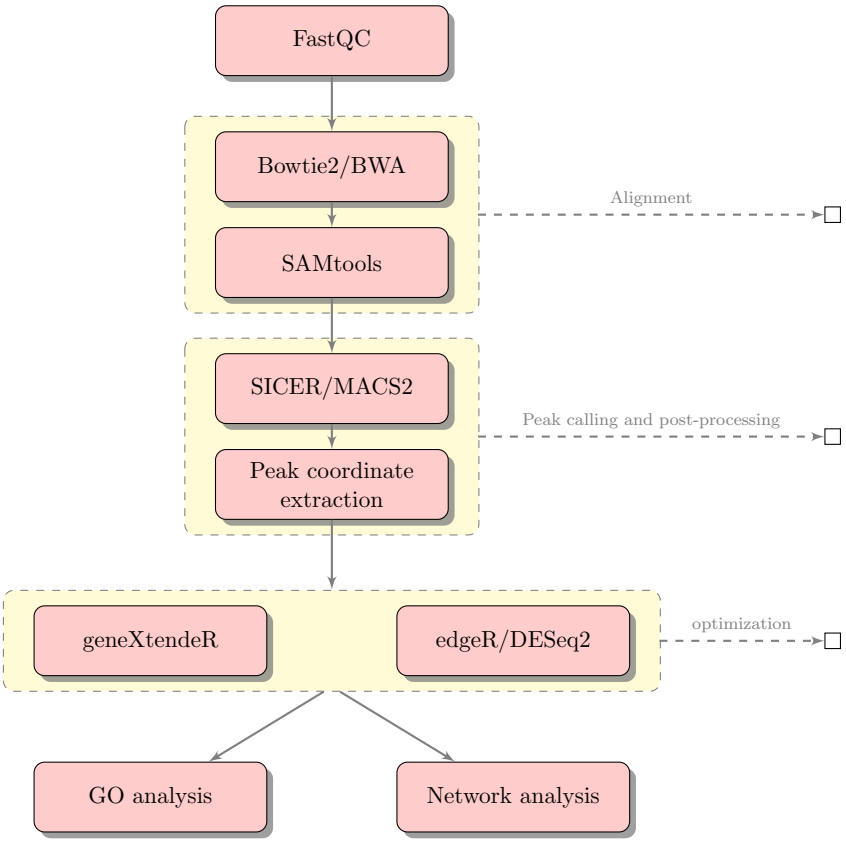
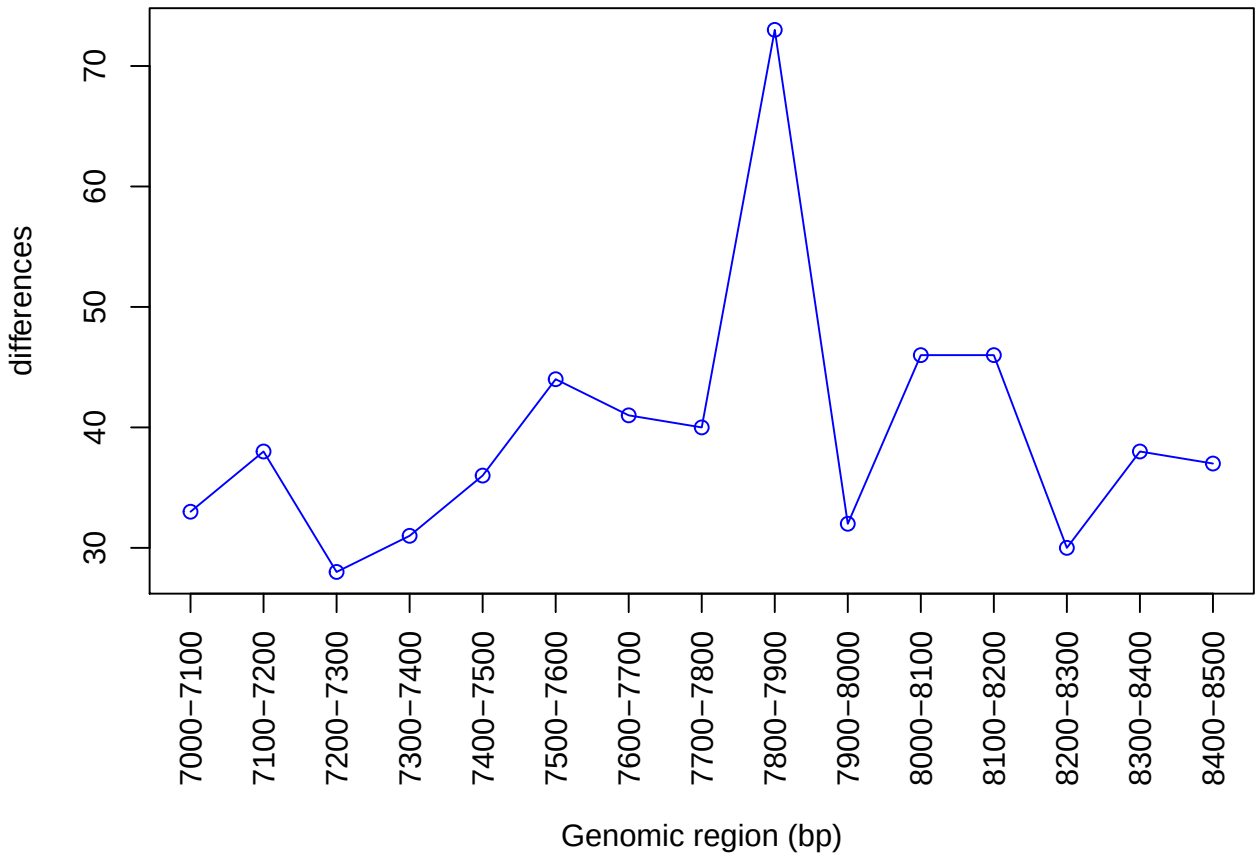


Figure 1: Sample biological workflow using geneXtendeR in combination with existing statistical software to analyze peak significance. Subsequent gene ontology or network analysis may be conducted on genes associated with statistically significant peaks.

beyond 3500 bp globally across every gene in a genome would most likely not accurately reflect the biology of the peak input file (since such large global upstream extensions are likely to reach considerably beyond known proximal promoter elements, especially for relatively narrow histone marks). Such assumptions may be validated directly by the user by investigating the p-value and FDR of specific peaks using a combination of HT-seq (to count the reads) and edgeR/DESeq (to assess statistical significance). As such, geneXtendeR is designed to be used as part of a biological workflow involving subsequent statistical analysis:

It is entirely possible (and probable) for significant peaks to be present at relatively high upstream extension levels (i.e., large gene-spheres), albeit these significant peaks may be associated with biology not directly relevant to the study at-hand, due mainly to the sheer magnitude of the distance of the peak from traditional gene boundaries (+/- 3kb from TSS and +/- 0.5kb from TES). Consequently, it is normal for peaks-on-top-of-genes to exhibit higher levels of noise at higher upstream extension levels. However, this does not mean that potential enhancer activity should be discounted. For instance, it is not uncommon to see a steady rise or even a surge in the number of peaks-on-top-of-genes at higher upstream extension levels:

```
generate(rat, 7000, 8500, 100)
linePlot()
```



In far-out cases like this, it is particularly recommended to examine the statistical significance of peaks to get a sense for potential enhancer activity. Assessment of such statistical significance values is beyond the scope of geneXtendeR, in order to allow the user freedom to choose the respective statistical package/technique. As before, first use the `distinct()` command to create a table of unique genes located under peaks between the two upstream extension levels:

```
distinct(rat, 7800, 7900)
```

```
##       V1        V2        V3  V4        V5        V6                 V7
##  1:    1 105025000 105027799   1 105027749 105041820 ENSRNOG00000056203
##  2:    1 250308800 250310399   1 250300153 250308852 ENSRNOG00000052057
##  3:    1 282629800 282630199   1 282630117 282702924 ENSRNOG00000036571
##  4:    2 116265800 116267599   2 116242873 116265880 ENSRNOG00000027995
##  5:    3  58521200  58522999   3  58522970  58558527 ENSRNOG00000001517
##  6:    3  61409200  61412199   3  61412161  61420640 ENSRNOG00000041393
##  7:    3 109832000 109837599   3 109823460 109832021 ENSRNOG00000061411
##  8:    4 127154200 127156599   4 127156553 127177977 ENSRNOG00000013390
##  9:    4 148454200 148455799   4 148398392 148454203 ENSRNOG00000012972
## 10:    5 163239400 163239999   5 163185849 163239478 ENSRNOG00000016782
## 11:    6 109454800 109458199   6 109458160 109505661 ENSRNOG00000008224
## 12:    6 138860400 138861599   6 138851641 138860471 ENSRNOG00000058426
## 13:    9 121733600 121734999   9 121706479 121733616 ENSRNOG00000049882
## 14:   12  19448400  19450999  12  19428627  19448401 ENSRNOG00000031343
## 15:   14  19167800  19169799  14  19141255  19167823 ENSRNOG00000002889
## 16:   15  62013800  62017599  15  62017594  62026090 ENSRNOG00000040381
## 17:   16  64336000  64337399  16  64337334  64346007 ENSRNOG00000058948
```

```
## 18:  17   9358600   9362999  17   9362965   9371481 ENSRNOG00000050396
## 19:  18  22413600  22415399  18  22315318  22413621 ENSRNOG00000051417
## 20: 100  87372800  87372999 100  87371289  87379795 ENSRNOG00000054668
## 21: 100 100639400 100643999 100 100631671 100640462 ENSRNOG00000046638
## 22: 100 106055600 106056799 100 106048575 106074127 ENSRNOG00000023256
## 23: 100 111310400 111311199 100 111268990 111333086 ENSRNOG00000057622
## 24: 100 111942200 111944799 100 111941602 111950649 ENSRNOG00000060899
## 25: 100 112767800 112768599 100 112761745 112984185 ENSRNOG00000018951
## 26: 100 112769400 112770399 100 112761745 112984185 ENSRNOG00000018951
## 27: 100 112804400 112805599 100 112761745 112984185 ENSRNOG00000018951
## 28: 100 113568600 113570199 100 113542492 113583064 ENSRNOG00000053818
## 29: 100 113572800 113573399 100 113542492 113583064 ENSRNOG00000053818
## 30: 100 114086600 114086799 100 113940754 114110562 ENSRNOG00000012787
## 31: 100 115735800 115736399 100 115627153 115916593 ENSRNOG00000027233
## 32: 100 118161800 118162999 100 118076990 118318539 ENSRNOG00000030877
## 33: 100 118218200 118219599 100 118076990 118318539 ENSRNOG00000030877
## 34: 100 118347800 118348399 100 118347032 118355504 ENSRNOG00000041256
## 35: 100 118481800 118482999 100 118443323 118520961 ENSRNOG00000032973
## 36: 100 119207400 119208999 100 119193937 119208675 ENSRNOG00000054795
## 37: 100 122632800 122633599 100 122499474 122690965 ENSRNOG00000013321
## 38: 100 123660600 123663799 100 123602742 123670314 ENSRNOG00000054788
## 39: 100 123660600 123663799 100 123654577 123664597 ENSRNOG00000049392
## 40: 100 123988000 123995399 100 123887488 124001442 ENSRNOG00000040013
## 41: 100 123988000 123995399 100 123976246 123988417 ENSRNOG00000031534
## 42: 100 128177400 128178599 100 128161395 128276222 ENSRNOG00000007315
## 43: 100 128177400 128178599 100 128169929 128183851 ENSRNOG00000059506
## 44: 100 139429800 139432999 100 139353655 139472698 ENSRNOG00000002413
## 45: 100 139483400 139485599 100 139475140 139483645 ENSRNOG00000047856
## 46: 100 139483400 139485599 100 139475155 139484106 ENSRNOG00000034150
## 47: 100 139483400 139485599 100 139476284 139487024 ENSRNOG00000058144
## 48: 100 140884000 140888399 100 140870316 140889295 ENSRNOG00000000861
## 49: 100 142130800 142132799 100 142053154 142256269 ENSRNOG00000042753
## 50: 100 145875200 145876599 100 145875295 145883847 ENSRNOG00000054106
## 51: 100 150661400 150662999 100 150660243 150674209 ENSRNOG00000058521
## 52: 100 152538600 152538999 100 152414115 152650431 ENSRNOG00000056558
## 53: 100 153063200 153064999 100 153056128 153068000 ENSRNOG00000052022
## 54: 100 155341400 155343199 100 155336276 155344793 ENSRNOG00000057982
## 55: 100 156203400 156204599 100 156202102 156211440 ENSRNOG00000058820
## 56: 100 157169800 157177399 100 157167644 157179968 ENSRNOG00000055185
## 57: 100 157590200 157591199 100 157588031 157599444 ENSRNOG00000060122
## 58: 100 159075400 159075999 100 159073545 159101012 ENSRNOG00000037452
##      V1        V2        V3  V4        V5        V6                 V7
##                 V8 V9
##  1: AABR07003310.2  0
##  2: AABR07006727.1  0
##  3:          Ces2c  0
##  4:          Samd7  0
##  5:           Pdk1  0
##  6: AABR07052554.1  0
##  7:             U1  0
##  8:         Kbtbd8  0
##  9:          Alox5  0
## 10:        Tnfrsf8  0
## 11:           Jdp2  0
```

```
## 12: AABR07065656.9  0
## 13:        Adcyap1  0
## 14:          Nxpe4  0
## 15:            Afp  0
## 16:         Mir759  0
## 17:            7SK  0
## 18: rno-mir-3542-1  0
## 19: AABR07031612.1  0
## 20:             U6  0
## 21:   LOC102549291  0
## 22:           Nxf7  0
## 23:          Rbm41  0
## 24:        Tsc22d3  0
## 25:         Col4a5  0
## 26:         Col4a5  0
## 27:         Col4a5  0
## 28: AABR07040906.1  0
## 29: AABR07040906.1  0
## 30:        Tmem164  0
## 31:          Trpc5  0
## 32:          Htr2c  0
## 33:          Htr2c  0
## 34: AABR07041066.1  0
## 35:        Il13ra2  0
## 36: AABR07041096.1  0
## 37:         Dock11  0
## 38: AABR07041239.1  0
## 39:         Sowahd  0
## 40:          Rhox2  0
## 41:          Rhox7  0
## 42:          Thoc2  0
## 43:     AC124926.2  0
## 44:           Gpc4  0
## 45: AABR07041778.1  0
## 46:         Ftl1l1  0
## 47: AABR07041778.3  0
## 48:           Zic3  0
## 49:          Fgf13  0
## 50:      5_8S_rRNA  0
## 51: AABR07042226.1  0
## 52:         Gabra3  0
## 53:          Pnma3  0
## 54:        Mir3585  0
## 55:        Olr1768  0
## 56:          Dusp9  0
## 57: AABR07042465.1  0
## 58:   LOC100364989  0
##                V8 V9
```

Then, assess the statistical significance of these peaks using a combination of HT-seq and edgeR, or HT-seq and DESeq2, or some other appropriate combination of existing software tools. Genes associated with the resultant statistically significant peaks may then be further assessed with gene ontology analysis or network analysis to help answer a variety of interesting research questions.

Even though geneXtendeR is designed to compute (and analyze/display) optimal gene extensions tailored

to the characteristics of a specific peak input file, geneXtendeR will not explicitly impose on the user the optimal extension to use, since this information is highly study-dependent and, as such, is ultimately reserved to the user's discretion. For example, a user may choose a conservatively lower upstream extension (e.g., for studies investigating narrow peaks such as H3K4me3 or H3K9ac that exhibit a compact and localized enrichment pattern, where high upstream extensions may lose biological meaning). Likewise, a user may also investigate the statistical significance of specific peaks of interest at varying upstream cutoffs via the help of external software (e.g., HT-seq/edgeR, HT-seq/DESeq2, etc). Once the user has chosen the specific upstream extension to be used, the peak file is ready to be fully annotated:

```
annotate(rat, 3300)
```

which generates a fully annotated peaks file containing various genomic features and labeled headers.