Reference Manual for Genomic Random Interval (GRIN) Analysis

Zhifa Liu, Stan Pounds

February 28, 2014

1 Introduction

Genomic random interval (GRIN) analysis [1] attempts to infer the biological significance of a genomic lesion data set by statistically modeling the genomic lesions as a set of genomic random intervals. Briefly, the statistical model represents lesions as genomic random intervals of fixed length that may occur at any position along the same chromosome with uniform probability. The frequencies at which lesions overlap with specific genomic loci, specific genes, or gene sets are statistics that measure the extent to which lesions 'target' those loci, genes, or gene sets. The observed frequencies are compared to the probability distribution derived by modeling the lesions as genomic random intervals. A genomic random interval (GRIN) model is used as a null model to evaluate the statistical significance of the pattern of overlap of genomic lesions with the loci of individual genes, predefined sets of genes, and each base-pair locus in the genome. The null model represents each lesion as an interval of fixed length and random location that may occur at any location along the chromosome with equal probability. The null model defines null distributions for the total number of overlapping gene-lesion pairs, the number of lesions that overlap any locus in a gene set, and the number of tumors with a lesion that overlaps any locus in a gene set. The significance of each overlap statistic may be determined by comparing its observed value to its null distribution defined by the GRIN model.

In this document, we describe how to perform GRIN procedure using example data sets provided with the package.

2 Installation

The latest verion of grin package can be download from http://www.stjuderesearch.org/site/depts/biostats/software and R CRAN version will be available soon.

```
> # Install packages
> # Load the package
> library(grin)
> # Load the example data set
> data(HYPO.lesion.data) #hg19
> data(ETP.lesion.data) #hg18
```

3 Overview the grin package

The grin package uses the genomic random interval (GRIN) model to identify particular loci, genes, or gene-sets that are significantly targeted by a set of observed genomic lesions. The package defines the following user-level functions and utility data sets:

3.1 Data Analysis Functions

- grin.genes: screen each individual gene for a significant abundance of genomic lesions.
- grin.genomes: identify specific loci with a significant abundance of lesions (loci may be intragenic or not)
- grin.gsets: identify specific gene-sets with a significant abundance of lesions
- grin.analysis: perform all three of the above analyses and save the results.
- robust.fdr: use the robust FDR method [2] to compute q-values for any of the above analyses.

3.2 Visualization Functions

- seg.heatmap: generate the heatmap for the lesion data
- seg.man.plot: generate the Manhattan plot

3.3 Annotation Data Sets

- hg18.chrom.sizes: gives the size of each chromosome under the hg18 version of the human genome
- hg18.gene.annotation.data: gives gene locations for the hg18 version of the human genome
- hg19.chrom.sizes: gives the size of each chromosome under the hg19 version of the human genome
- hg19.gene.annotation.data: gives gene locations for hg19 version of the human genome
- KEGG.gset.data: defines multiple gene-sets for an example analysis. Each row assigns one gene to a gene-set. This data set includes the locations of a few selected genes as an example. The selected genes are members of the KEGG acute myeloid leukemia pathway and a list of genes known to be important for T-cell development.

3.4 Genomic Lesion Data Sets

- ETP.lesion.data [3]
- HYPO.lesion.data [4]

4 Prepare the Lesion dataset

The grin package includes two example datasets: HYPO (hg19) and ETP (hg18). The lesion data have the following data format:

```
> library(xtable)
> print(xtable(HYPO.lesion.data[1:6,]))
```

Each row of this data gives information about one genomic lesion. The *subj.ID* gives the identifier of the subject (or tumor) in which the lesion was observed and the colums *chrom, loc.start, and loc.end* give the locus of the lesion. These four columns are required for the grin analysis functions *grin.genes, grin.genome, and grin.gsets*. The column *type* gives the type of genomic lesion (such as copy number gain, copy number loss, loss of heterozygosity, etc). The *type* column is required for the *seg.heatmap* function.

	subj.ID	chrom	loc.start	loc.end	type
1	SJHYPO002	1	10101	249240500	loss
2	SJHYPO002	11	133701	134946400	loss
3	SJHYPO002	12	60701	133841500	loss
4	SJHYPO002	13	19020701	115109800	loss
5	SJHYPO002	14	106330472	106382690	loss
6	SJHYPO002	14	106382691	106405607	loss

In preparing the data set, be sure that the lesion loci and the gene loci are based on the same version of the genome assembly (hg18, hg19, etc). Additionally, note that the data uses the same convention for the X, Y (MT chromosomes need to be removed in our GRIN analysis). In some analyses, the convention that represents X as 23, Y as 24, may be necessary.

Also, note that some reformatting of copy number analysis results may be necessary. For example, a homozygous deletion should be represented as two rows in the genomic lesion data because there are two lesions (deletion on each chromosome). In our work, we represent structural rearrangements by the set of breakpoint loci (single base pair if identified from sequencing data using CREST [5].

5 GRIN Analysis

After the genomic lesion data has been prepared and formatted as described above, we are ready to perform an analysis.

5.1 Gene level analysis

The following code performs the *grin.genome* analysis:

```
> data(ETP.lesion.data)
> data(hg18.gene.annotation.data) # gene annotation data frame
> data(hg18.chrom.sizes.data)  # chrom size data frame
> # hg18 example
> ETP.gene.res<-grin.genes(ETP.lesion.data, hg18.gene.annotation.data, hg18.chrom.sizes.data)
> # hg19 example
> HYPO.gene.res<-grin.genes(HYPO.lesion.data)
```

The second analysis uses the *hg19.gene.annotation.data* and *hg19.chrom.sizes.data* by default.

The result file is formatted as following. Each row contains the result for one gene. The first set of columns recapitulate the contents of the gene location data. The next set of columns provide statistical analysis results. The names and meanings of those main columns are listed below.

- chrom: chrom number
- gene.id: NCBI Entry gene ID
- gene.label: gene name
- loc.start: start location of the gene
- loc.end: end location of the gene
- n.overlaps: total number of lesion loci that overlap the locus of the gene
- p.overlaps: p-value for n.overlaps from the convolution model

- q.overlaps: estimated false discovery rate for p.overlaps [2]
- n.subjects: lists the number of subjects with a lesion overlapping the gene
- p.subjects: p-value for n.subjects from the convolution model
- q.subjects: false discovery rate for p.subjects

After we finished the grin.genes analysis, we can display the top 10 affected genes based on *p.subjects* in ETP example :

chrom	gene.label	n.overlaps	p.subjects	q.subjects	p.overlaps	q.overlaps
21	RUNX1	4	3.14e-09	9.15e-05	3.14e-09	8.33e-05
19	JAK3	4	1.56e-08	9.15e-05	1.82e-08	1.06e-03
17	SUZ12	4	8.65e-07	9.15e-05	1.30e-08	1.18e-04
2	FLJ41327	2	1.49e-05	9.15e-05	6.10e-05	1.20e-01
22	EP300	3	1.55e-05	9.15e-05	1.85e-05	7.15e-02
2	FLJ38379	2	1.73e-05	9.15e-05	6.95e-05	1.20e-01
2	FLJ40712	2	1.83e-05	1.75e-04	7.42e-05	1.35e-01
23	PHF6	3	2.20e-05	5.29e-03	4.63e-05	1.12e-01
2	LOC285095	2	2.93e-05	3.64e-02	1.12e-04	1.60e-01
2	LOC100131763	2	3.02e-05	5.36e-02	1.17e-04	1.63e-01

5.2 Genome level analysis

In *grin.genome* analysis, the lesion loci are used to empirically define distinct interval loci and then determine the significance of the abundance of lesions overlapping each of those empirically defined loci. The following code performs the *grin.genome* analysis:

```
> data(ETP.lesion.data)
> data(hg18.gene.annotation.data) # gene annotation data frame
> data(hg18.chrom.sizes.data)  # chrom size data frame
> # hg18 example
> ETP.genome.res<-grin.genome(ETP.lesion.data, hg18.gene.annotation.data, hg18.chrom.sizes.data)
> # hg19 example
> HYPO.genome.res<-grin.genome(HYPO.lesion.data)
```

In the output of the *grin.genome*, each row provides results for one of the intervals defined by the endpoints of the lesion loci. The rows are ordered first by prevalence of lesions and then statistical significance. The meanings of the main columns are explained below.

- interval.id: gives a numeric identifier to the interval
- chrom: chrom number
- loc.start: location of start of each gene
- loc.end: location of end of each gene
- size: gives the size of the interval in base pairs
- n.gene: gives the number of genes in the segment
- gene.list: lists the gene names in the segment
- n.overlaps: total number of lesion loci that overlap the locus of the interval

- p.overlaps: p-value for n.overlaps from the convolution model
- q.overlaps: false discovery rate for p.overlaps
- n.subjects: lists the number of subjects with a lesion overlapping the interval
- p.subjects: p-value for n.subjects from the convolution model
- q.subjects: false discovery rate for p.subjects

The following table displays the the output of the *grin.genome*. This may be saved in csv (using *write.csv*), tab-delimited (using *write.table*), or Rdata format (using *save*).

interval.id	chrom	loc.start	gene.list	p.subjects	q.subjects	p.overlaps	q.overlaps
653	19	17810108	JAK3	1.50e-21	2.92e-08	1.75e-21	1.30e-18
681	21	35174736	RUNX1	7.99e-10	3.93e-07	7.99e-10	9.62e-08
683	21	35174741	RUNX1	7.99e-10	3.93e-07	7.99e-10	9.62e-08
685	21	35174860	RUNX1	7.99e-10	3.93e-07	7.99e-10	9.62e-08
583	15	39011256	DLL4	1.13e-08	1.90e-06	2.52e-03	3.02e-02
585	15	40774080	STARD9	1.13e-08	1.90e-06	2.52e-03	3.02e-02
675	20	43366789	MATN4 RBPJL	4.80e-08	4.08e-06	4.80e-08	3.37e-06
677	20	62126384	PRPF6	4.80e-08	4.08e-06	4.80e-08	3.37e-06
689	22	15980477	CECR6	8.05e-08	5.05e-06	8.05e-08	5.30e-06

5.3 Gene-Set level analysis

Please use the code below to perform the grin.gsets analysis.

```
> data(ETP.lesion.data)
> data(hg18.gene.annotation.data)
> data(hg18.chrom.sizes.data)
> data(KEGG.gset.data)
> # hg18 example
> gset.res<-grin.gsets(ETP.lesion.data, hg18.gene.annotation.data,
+                      KEGG.gset.data, hg18.chrom.sizes.data)
> # hg19 example
> gset.res.HYPO<-grin.gsets(HYPO.lesion.data,  gset.data= KEGG.gset.data)
```

In the package, we have provided an example of gset data (KEGG.gset.data). You will need to generate your own gset data for your application. Please represent your gset data in the following format.

```
> data(KEGG.gset.data)
> names(KEGG.gset.data)
```

- gset.source: a character vector that provides the source of the gene-set definition (e.g., KEGG)
- gset.id: a character vector that gives the identifier for the gene-set
- gset.label: a character vector that gives the label (meaningful name) of the gene-set
- gene.id: the official NCBI Entry ID that gives the identifier of gene assigned to the gene-set.
- `NCBI_Gene`: the official NCBI gene label which gives the identifier of gene assigned to the gene-set.

The KEGG example are assembled as the following:

	gset.source	gset.id	gset.name	gset.genes	gene.id	NCBI_Gene
1	Mullighan	T-cell pathway	T-cell pathway	ETV6	2120	ETV6
2	Mullighan	T-cell pathway	T-cell pathway	NOTCH1	4851	NOTCH1
3	Mullighan	T-cell pathway	T-cell pathway	IKZF1	10320	IKZF1
4	Mullighan	T-cell pathway	T-cell pathway	GATA3	2625	GATA3
5	Mullighan	T-cell pathway	T-cell pathway	RUNX1	861	RUNX1
6	Mullighan	T-cell pathway	T-cell pathway	GFI1	2672	GFI1

```
> library(xtable)
> print(xtable(KEGG.gset.data[1:6,1:6]))
```

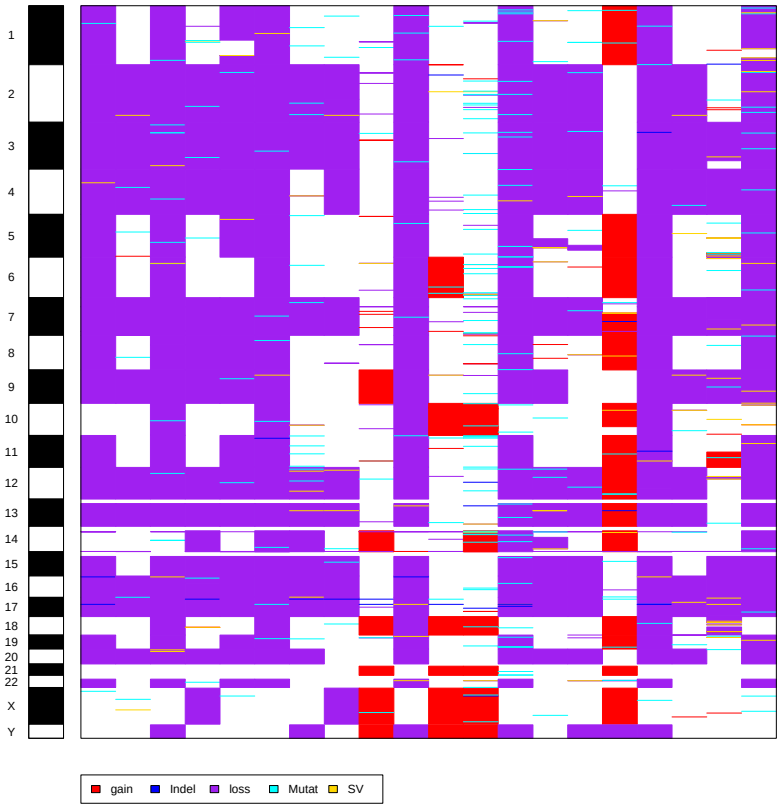
In the output of *grin.gsets*, each row of data.frame contains the results for one gene set. The definition of main columns are listed below:

- gset.source lists the source of the gene set definition
- gset.id lists the identifier of the gene set
- gset.label lists the label of the gene set
- ngenes.gset gives the number of genes assigned to the gene set
- gset.gene.ids lists the genes Entry ID(NCBI) assigned to the gene set
- ngenes.with.locus.data gives the number of genes in the set that were successfully matched to the gene location data
- all.genes.matched indicates (TRUE/FALSE) whether all genes in the set were successfully matched to the gene location data
- n.overlaps total number of lesion loci that overlap the locus of the gene set
- p.overlaps p-value for n.overlaps from the convolution model
- q.overlaps false discovery rate for p.overlaps
- n.subjects lists the number of subjects with a lesion overlapping the interval.
- p.subjects p-value for n.subjects from the convolution model
- q.subjects false discovery rate for p.subjects
- n.lesions the total number of lesions that overlap at least one gene in the gene set
- p.lesions p-value for n.lesion from the convolution model
- q.lesions false discovery rate for p.lesions

6 Generate the Lesion Heatmap

The grin package provides the *seg.heatmap* function to visualize the lesion data set as a heatmap. The lesion type variable is a required column to define type of lesion dataset. Here is the example:

```
> # hg18 example
> # seg.heatmap(ETP.lesion.data,hg18.chrom.sizes.data )
> # hg19 example
> seg.heatmap(HYPO.lesion.data)
```

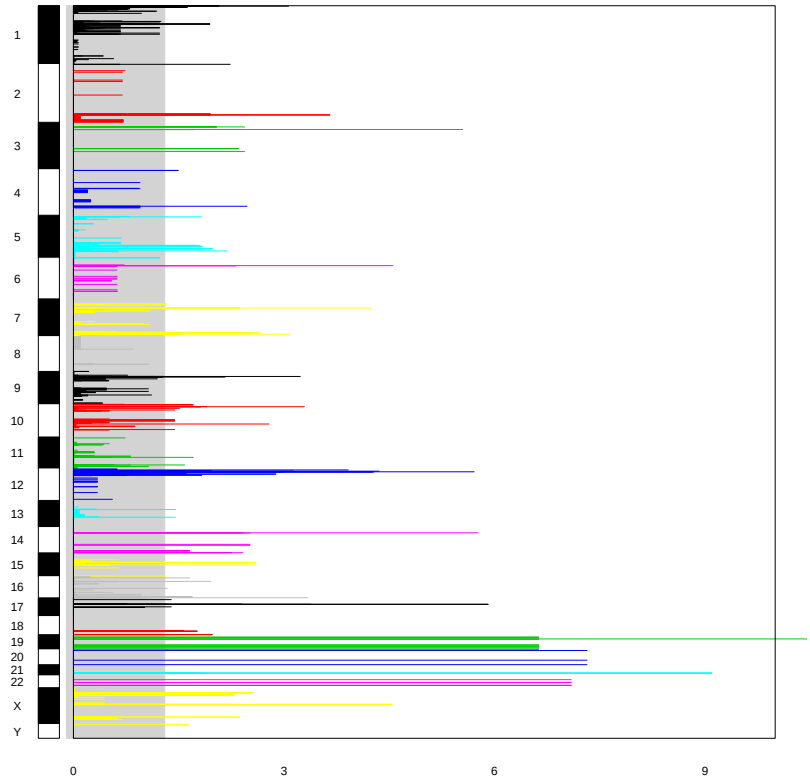


If your lesion data uses hg19 coordinates, then you don't need to specify the *chrom.size* parameter since the default is hg19.

7 Generate the Manhattan plot

The grin package provides the *seg.man.plot* function to generate the Manhattan plot for the output of grin.genes, grin.genome and grin.gsets. Here is the example:

```
> data(ETP.lesion.data)
> data(hg18.gene.annotation.data)
> data(hg18.chrom.sizes.data)
> genome.res<-grin.genome(ETP.lesion.data, hg18.gene.annotation.data, hg18.chrom.sizes.data)
> seg.man.plot(genome.res,hg18.chrom.sizes.data,"p.overlaps")
```



8 Contact Information

If you have any technique questions about using the grin package, please email us at: zhifa.liu@stjude.org or stanley.pounds@stjude.org.

References

[1] S. Pounds, C. Cheng, S. Li, Z. Liu, J. Zhang, and C. Mullighan, "A Genomic Random Interval Model for Statistical Analysis of Genomic Lesion Data," *Bioinformatics*, Jul 2013.

[2] S. Pounds and C. Cheng, "Robust estimation of the false discovery rate," *Bioinformatics*, vol. 22, pp. 1979–1987, Aug 2006.

[3] J. Zhang, L. Ding, L. Holmfeldt, G. Wu, S. L. Heatley, D. Payne-Turner, J. Easton, X. Chen, J. Wang, M. Rusch, C. Lu, S. C. Chen, L. Wei, J. R. Collins-Underwood, J. Ma, K. G. Roberts, S. B. Pounds, A. Ulyanov, J. Becksfort, P. Gupta, R. Huether, R. W. Kriwacki, M. Parker, D. J. McGoldrick, D. Zhao, D. Alford, S. Espy, K. C. Bobba, G. Song, D. Pei, C. Cheng, S. Roberts, M. I. Barbato, D. Campana, E. Coustan-Smith, S. A. Shurtleff, S. C. Raimondi, M. Kleppe, J. Cools, K. A. Shimano, M. L. Hermiston, S. Doulatov, K. Eppert, E. Laurenti, F. Notta, J. E. Dick, G. Basso, S. P. Hunger, M. L. Loh, M. Devidas, B. Wood, S. Winter, K. P. Dunsmore, R. S. Fulton, L. L. Fulton, X. Hong, C. C. Harris, D. J. Dooling, K. Ochoa, K. J. Johnson, J. C.

Obenauer, W. E. Evans, C. H. Pui, C. W. Naeve, T. J. Ley, E. R. Mardis, R. K. Wilson, J. R. Downing, and C. G. Mullighan, "The genetic basis of early T-cell precursor acute lymphoblastic leukaemia," *Nature*, vol. 481, pp. 157–163, Jan 2012.

[4] L. Holmfeldt, L. Wei, E. Diaz-Flores, M. Walsh, J. Zhang, L. Ding, D. Payne-Turner, M. Churchman, A. Andersson, S. C. Chen, K. McCastlain, J. Becksfort, J. Ma, G. Wu, S. N. Patel, S. L. Heatley, L. A. Phillips, G. Song, J. Easton, M. Parker, X. Chen, M. Rusch, K. Boggs, B. Vadodaria, E. Hedlund, C. Drenberg, S. Baker, D. Pei, C. Cheng, R. Huether, C. Lu, R. S. Fulton, L. L. Fulton, Y. Tabib, D. J. Dooling, K. Ochoa, M. Minden, I. D. Lewis, L. B. To, P. Marlton, A. W. Roberts, G. Raca, W. Stock, G. Neale, H. G. Drexler, R. A. Dickins, D. W. Ellison, S. A. Shurtleff, C. H. Pui, R. C. Ribeiro, M. Devidas, A. J. Carroll, N. A. Heerema, B. Wood, M. J. Borowitz, J. M. Gastier-Foster, S. C. Raimondi, E. R. Mardis, R. K. Wilson, J. R. Downing, S. P. Hunger, M. L. Loh, and C. G. Mullighan, "The genomic landscape of hypodiploid acute lymphoblastic leukemia," *Nat. Genet.*, vol. 45, pp. 242–252, Mar 2013.

[5] J. Wang, C. G. Mullighan, J. Easton, S. Roberts, S. L. Heatley, J. Ma, M. C. Rusch, K. Chen, C. C. Harris, L. Ding, *et al.*, "Crest maps somatic structural variation in cancer genomes with base-pair resolution," *Nature methods*, vol. 8, no. 8, pp. 652–654, 2011.