Gloria-Soria (ddRAD58)

Gus Dunn

2014-12-29

Contents

1 Functional Annotations **1**
1.1 Sequences Used . . . 1
1.2 Argot2 Analysis . . . 1

2 Linkage Analysis **2**
2.1 Source of SNPs . . . 2
2.2 Linkage measurements . . . 2
2.3 LD-based filtering of SNP-pairs . . . 2
2.3.1 Scaling of binned r^2 distributions . . . 2
2.3.2 Bayesian parameter estimation using the binned r^2 distributions . . . 3

Bibliography **4**

1 Functional Annotations

1.1 Sequences Used

Glossina fuscipes fuscipes: All putative peptides annotated for *G. f. fuscipes* in the GfusI1.1 gene-build were obtained from VectorBase [1]

Other: The sequences used to compare the *G. f. fuscipes* peptides against well known/annotated sequences were obtained from UNIPROT/SwissProt [2] (used with `blastp`) and PFAM [3] (used with `hmmscan`) as required by ARGOT2 [4–6].

1.2 Argot2 Analysis

The `blastp` and `hmmscan` results submitted to ARGOT2 were obtained by performing local searches on the *G. f. fuscipes* peptides against the UNIPROT peptide database (obtained on 2014-09-08) and the hidden Markov models (HMM) of the combined protein-domain sets om the Pfam databases (Pfam-A and Pfam-B: obtained on 2014-09-08), respectively. Settings used were as dictated by the ARGOT2 site.

The `blastp` and `hmmscan` results were uploaded to ARGOT2 servers for analysis after being split into 10 groups (roughly 2330 peptides per group) to prevent overloading the remote ARGOT2 cluster. The functional annotations were then downloaded and joined back together.

2 Linkage Analysis

2.1 Source of SNPs

SNPs were obtained as described earlier in this manuscript.

2.2 Linkage measurements

Plink version 1.9 [7] was used to calculate pairwise linkage disequilibrium (LD) as r for all SNP-pairs located on common supercontigs. The `--allow-extra-chr` option was required to handle the number of supercontigs. Unless stated otherwise, all subsequent analysis pertaining to LD used r^2.

2.3 LD-based filtering of SNP-pairs

The LD values of SNP-pairs were compared after binning SNP-pairs by base pair separation to control for unknown rates of recombination in *G. f. fuscipes*. Bin length was set at 50 bp with the lower bound inclusive and higher bound exclusive. In other words the bins were defined as $[i, i + 49)$ where $i \in \{1, 1(50), 2(50), 3(50)...n(50)\}$ and n is a positive integer.

To identify SNP-pairs for further investigation in an arbitrary set of binned SNP-pairs, we needed to assign probabilities to each SNP-pair in the bin. The distributions of binned SNP-pairs are bounded by 0 and 1 and do not appear to be Normal in shape. Additionally, in many cases, the data appear to exhibit peaks at both the lower and upper r^2 range. This suggested that the data may be modeled well using the probability density function (PDF) of the Beta distribution:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha, \beta)}$$

with shape parameters α and β and where x represents an observed value generated by the distribution and $\mathrm{B}()$ is the Beta function **{{CITE_ME}}**.

The Beta's cumulative distribution function (CDF)

$$F(x; \alpha, \beta) = \frac{\mathrm{B}(x; \alpha, \beta)}{\mathrm{B}(\alpha, \beta)}$$

can be used to describe the probability that a binned SNP-pair will be observed to have an $r^2 \leq x$. It follows that $1 - \mathrm{CDF}$ represents the probability of observing a more extreme value.

For each set of binned r^2 values, the SNP-pairs deemed worthy of further investigation were defined as those where $1 - \mathrm{CDF} \leq 0.01$ after Benjamini-Hochberg (BH) correction for multiple testing [8].

2.3.1 Scaling of binned r^2 distributions

The Beta distribution is bounded on the non-inclusive interval between 0 and 1. However, there are data in each bin that may have been assigned values of exactly 0 or 1. It is likely that these values are not truly 0 or 1 in the discrete binary since that a coin-flip is *either* heads or tails. Therefore, the all data for each bin were scaled according to the following scheme

$$((x_i - 0.5) \cdot \theta) + 0.5$$

in order to symmetrically shrink the distribution of values to fit within the open interval $(0, 1)$. In the scheme above, let x_i stand for the r^2 of each SNP-pair in an arbitrary bin set and θ stand for the scaling factor. The relevant results in this paper used $\theta = 0.999$.

2.3.2 Bayesian parameter estimation using the binned r^2 distributions

In order to use the CDF of the Beta distribution to assign significances to each observed SNP-pair in a bin, it is necessary to learn the distribution parameters (α and β) for each bin given the specific data in each bin. For this we used custom python code that made heavy use of the following third-party data analysis modules: Pandas [9], NumPy [10], SciPy [11], pyMC [12], and StatsModels [13].

We used pyMC to build the model of the Beta distribution and use it to exploit the bin-specific data to estimate the bin-specific values for the α and β parameters of the model [Figure fig_betamod]. The values of α and β were modeled with separate Uniform prior distributions from 0.01 to 10. This model topology was used to create pyMC "model" objects and initialized with the r^2 data from each bin. The parameters of each Beta distribution were then estimated by *maximum a posteriori* (MAP) and used to calculate the $1 - \text{CDF}$ for each SNP-pair in each bin [10,11]. The $1 - \text{CDF}$ values were then BH corrected by bin and filtered with a threshold of $(1 - CDF)_{BH} \leq 0.01$ using statsmodels [8].

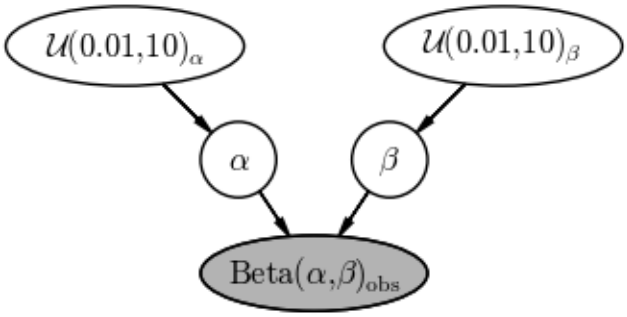

Figure 1: **[fig_betamod]Network representation of the LD Beta model**: Ovals represent modeled probability distributions. Circles represent learned parameters. Grey shading indicates use of observed data.

Bibliography

1. Giraldo-Calderon GI, Emrich SJ, MacCallum RM, Maslen G, Dialynas E, Topalis P, et al. VectorBase: an updated bioinformatics resource for invertebrate vectors and other organisms related with human diseases. Nucleic Acids Research. 2014;43: D707–D713. doi:10.1093/nar/gku1117

2. Boeckmann B, Blatter M-C, Famiglietti L, Hinz U, Lane L, Roechert B, et al. Protein variety and functional diversity: Swiss-Prot annotation in its biological context. Comptes rendus biologies. 2005;328: 882–99. doi:10.1016/j.crvi.2005.06.001

3. Finn RD, Bateman A, Clements J, Coggill P, Eberhardt RY, Eddy SR, et al. Pfam: the protein families database. Nucleic acids research. 2014;42: D222–30. doi:10.1093/nar/gkt1223

4. Radivojac P, Clark WT, Oron TR, Schnoes AM, Wittkop T, Sokolov A, et al. A large-scale evaluation of computational protein function prediction. Nature methods. Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved. 2013;10: 221–7. doi:10.1038/nmeth.2340

5. Falda M, Toppo S, Pescarolo A, Lavezzo E, Di Camillo B, Facchinetti A, et al. Argot2: a large scale function prediction tool relying on semantic similarity of weighted Gene Ontology terms. BMC bioinformatics. BioMed Central Ltd; 2012;13 Suppl 4: S14. doi:10.1186/1471-2105-13-S4-S14

6. Gillis J, Pavlidis P. Characterizing the state of the art in the computational assignment of gene function: lessons from the first critical assessment of functional annotation (CAFA). BMC Bioinformatics. BioMed Central Ltd; 2013;14: S15. doi:10.1186/1471-2105-14-S3-S15

7. Chang CC, Chow CC, Tellier LC, Vattikuti S, Purcell SM, Lee JJ. Second-generation PLINK: rising to the challenge of larger and richer datasets. GigaScience. BioMed Central Ltd; 2015;4: 7. doi:10.1186/s13742-015-0047-8

8. Benjamini Y, Hochberg Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. Journal of the Royal Statistical Society Series B (Methodological). Wiley for the Royal Statistical Society; 1995;57: pp. 289–300. Available: http://www.jstor.org/stable/2346101

9. McKinney W. Data Structures for Statistical Computing in Python. In: Walt S van der, Millman J, editors. Proceedings of the 9th python in science conference. 2010. pp. 51–56.

10. Van Der Walt S, Colbert SC, Varoquaux G. The NumPy array: A structure for efficient numerical computation. Computing in Science and Engineering. 2011;13: 22–30. doi:10.1109/MCSE.2011.37

11. Jones E, Oliphant T, Peterson P, Others. SciPy: Open source scientific tools for Python [Internet]. 2001. Available: http://www.scipy.org/

12. Patil A, Huard D, Fonnesbeck CJ. PyMC: Bayesian Stochastic Modelling in Python. Journal of statistical software. 2010;35: 1–81. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3097064/&tool=pmcentrez/&rendertype=abstract

13. Statsmodels-development-team. StatsModels: Statistics in Python (v0.6.1) [Internet]. Available: http://statsmodels.sourceforge.net/stable/