Epistasis Testing on the Cloud

Limsoon Wong

5 October 2011

Plan

- **Empirical Comparison of Recent Epistasis Analysis Methods**
- **Epistasis Analysis on the Cloud**

Acknowledgements

- Wang Yue
- Feng Mengling
- Liu Guimei

- Wang Zhengkui
- Tan Kian Lee
- Divyakant Agrawal

- A*STAR SERC PSF grant

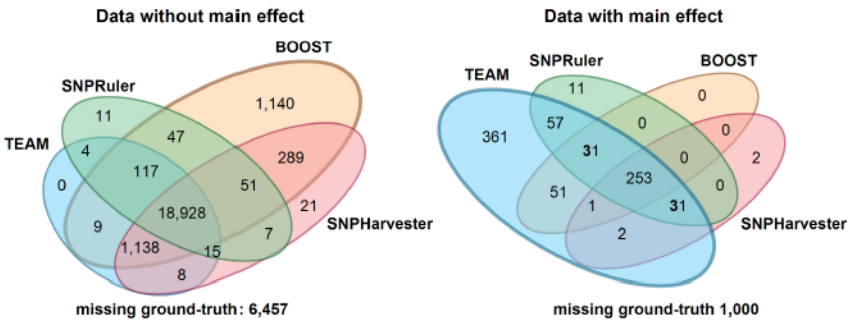
Empirical Comparison of Recent Epistasis Analysis Methods

Epistasis Analysis Methods

- **BOOST (B), TEAM (T), SNPRuler (SR), SNPHarvester (SH), Screen and Clean (SC)**

	B	T	SR	SH	SC
Exhaustive Search	×	√	×	×	×
Logit Model Assumed	√	×	×	√	√
Multi-Stage	×	×	×	×	√
Permutation Test	×	√	×	×	×
Bonferroni correction	√	×	√	√	√
Programming language	C	C++	Java	Java	R

Power



- **BOOST detected most of the ground-truth epistatic interactions**
- **TEAM detected most of the ground-truth epistatic interactions**

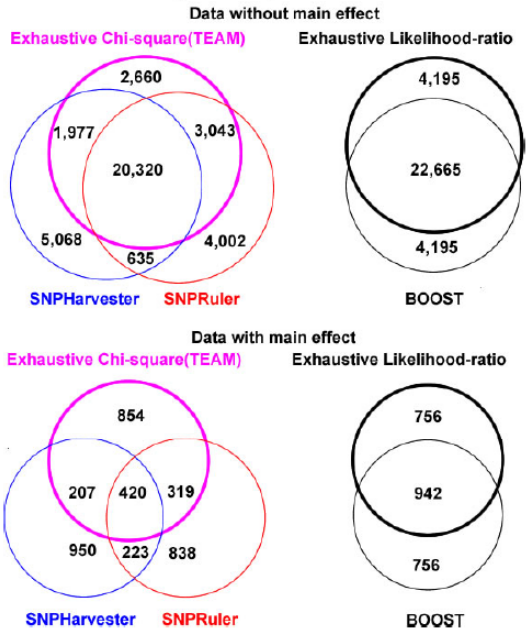
Type-1 Error Rate

- **Type-1 error rate**
  - Proportion of datasets that a method reports existence of significant epistatic interactions, out of the 1,000 datasets in which no epistatic interactions are embedded

Method	Type-1 Error Rate
TEAM	0.018
BOOST	0.065
SNPRuler	0.003
SNPHarvestor	0.003
Screen and Clean	0.860

Completeness

- **Non-exhaustive methods wrongly pruned many top significant epistatic interactions**



Performance

Table 3. Running time comparison of the five methods. SR is short for SNPRuler, SH is SNPHarvester, SC is Screen and Clean.

# SNPs	TEAM	BOOST	SR	SH	SC
100	58.23s	0.162s	2.43s	2.29s	7.389s
1000	353s.20s	2.47s	21.73s	22.33s	55.475s
10000	7406.29s	156.16s	1097.65s	224.24s	626.96s
100000	~36 days	15010.42s	NA	6616.65s	5858.34s

Remarks

- **Methods that are more "exhaustive" in testing of SNP pairs (i.e., BOOST, TEAM)**
  - Are more sensitive
  - Have good type-1 error rate
  - But take long time to compute

⇒

- **Stick to exhaustive methods**
- **Need large-scale parallelism**

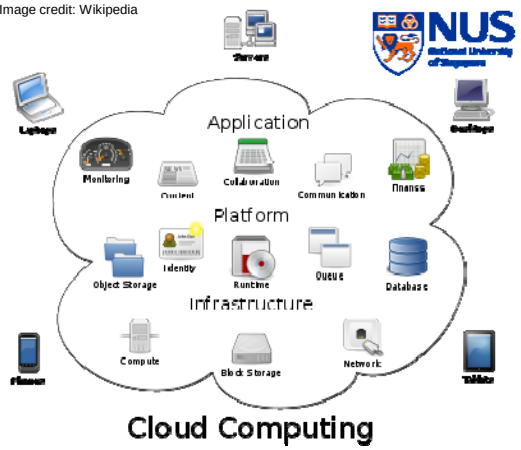
⇒ **Cloud computing!**

Epistasis Analysis on the Cloud

NUS National University of Singapore

Cloud Computing: What

Image credit: Wikipedia



- **Cloud computing is the delivery of computing as a service, whereby shared resources, software and information are provided to computers and other devices as a utility over the Internet**

Cloud Computing: Why

- **Reduced Cost**
  - Cloud computing is paid incrementally, saving money
- **Increased Storage**
  - Can store more data than on private computer systems
- **Highly Automated**
  - No need to worry about keeping software up to date
- **Flexibility**
  - More flexible than past computing methods
- **More Mobility**
  - Access info from anywhere
- **Allows IT to Shift Focus**
  - No worry about server updates and other computing issues

  ⇒Free to concentrate on innovation

Zhengkui Wang, Yue Wang, Kian-Lee Tan, Limsoon Wong, Divyakant Agrawal. **eCEO: An efficient Cloud Epistasis cOmputing model in genome-wide association study**. *Bioinformatics*, 27(8):1045--1051, April 2011

Cloud Computing: How

Raw Data

Sample Id	1	2	3	4	5	6	PT
1	1	2	1	2	2	2	1
2	0	0	2	0	1	0	0
3	2	1	0	1	0	2	1
4	1	1	1	0	1	0	0
5	1	0	1	1	1	2	1
6	1	0	2	0	0	1	0
7	1	0	0	2	0	0	0
8	0	2	0	1	1	2	0

(a)

Data after Preprocessing

Snp	PT	GT list
1	0	0 10000010
1	0	1 01101000
1	1	0 00010000
1	1	1 00000001
1	1	2 00000100
2	0	0 00101010
.		
.		
.		
6	1	2 00010101

(b2)

Boolean Operation

01101000
AND 00101010
00101000

(c2)

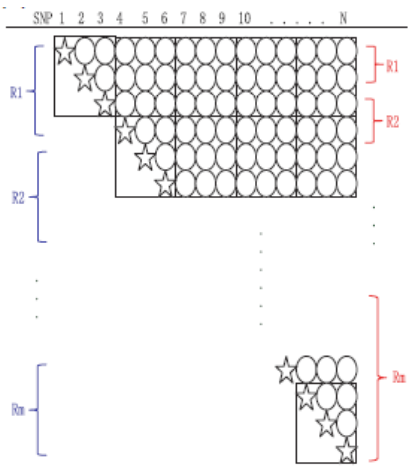
- **eCEO uses bit string data representation**

Zhengkui Wang, Yue Wang, Kian-Lee Tan, Limsoon Wong, Divyakant Agrawal. **eCEO: An efficient Cloud Epistasis cOmputing model in genome-wide association study**. *Bioinformatics*, 27(8):1045--1051, April 2011

Cloud Computing: How

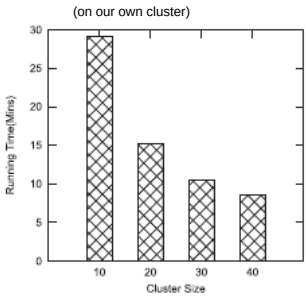
- **eCEO uses**
  - Map-reduce processing frame work
  - Load–balanced distribution of SNP pairs to 2M reducers

Greedy model: ideally, each `reducer` should process $\frac{N(N-1)}{2M}$ SNP pairs. Therefore, starting from the first row, we seek to allocate consecutive rows to a `reducer` such that the total number of SNP pairs for these rows is closest to $\frac{N(N-1)}{2M}$. In Figure 3a, the square brackets on the RHS show that, under the greedy scheme, each `reducer` may be assigned different number of rows to process. However, the computation task in each `reducer` is about the same. From our experimental results, we can see that our Greedy model has almost linear speed up when adding more resources which shows that our Greedy model is nearly load balanced.

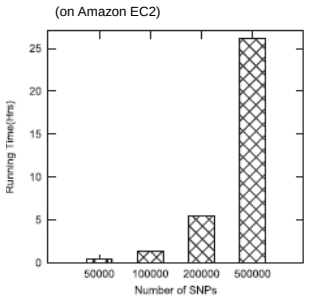


SNP-pairs representation and distribution to reducers

eCEO Performance

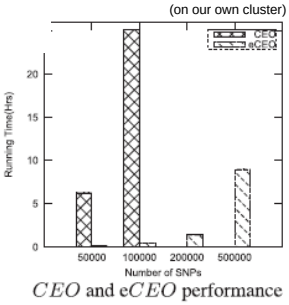


eCEO Scalability on different clusters



eCEO performance on EC2

It takes 1.2 years to do pairwise epistasis testing of 500 000 SNPs using a serial program on a 2.66 GHz single processor w/o parallel processing. Our eCEO can do this in < 9h using only a 43-node cluster



CEO and *eCEO* performance comparison

Any Question?