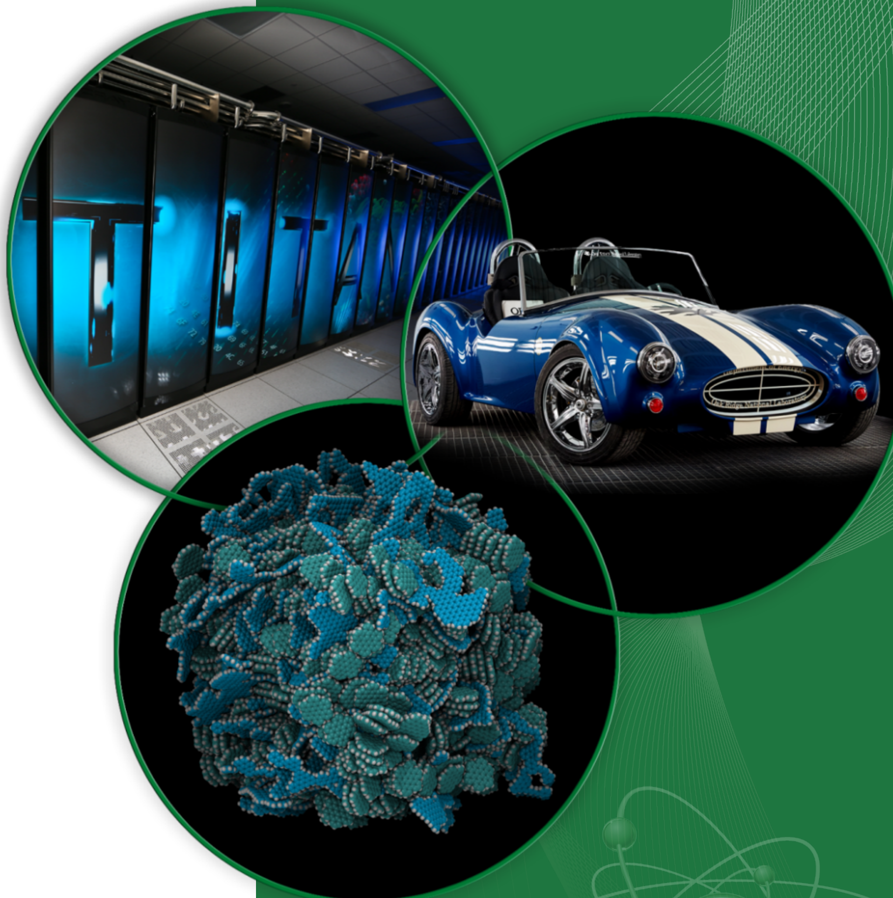
BigPanDA Tech Interchange Meeting (17 January 2018)



Experiments/Users:

Computational Biology

Manesh B. Shah

Sandra K. Huynh Truong

Ryan F. McCormick

Dr. Daniel A. Jacobson

Topics

- Computational Biology problem space
- Genetic Regulatory Networks - Background
- Two-dimensional genome scans using GBOOST
  – Initial Titan / PanDA implementation
  – Plans for comprehensive testing, profiling
- Future plans
  – Larger, diverse datasets
  – Incorporate multiple software packages

Computational Biology Problem Space

- 2D Genome Scans to capture genetic regulatory networks
  - GBOOST for discrete (categorical) data
  - epiGPU for continuous (quantitative) data
- Protein docking
  - AutoDock
  - MPI-Vina

2D genome scans using GBOOST software

Two-dimensional genome scans to capture genetic regulatory networks

Oak Ridge Leadership and Computing Facility (OLCF) Proposal

Sandra K. Huynh Truong

Ryan F. McCormick

Daniel A. Jacobson

Computational Biology, Biosciences Division,

Oak Ridge National Laboratory

2017-07-17

- Background
  - Genetic regulation of traits is determined by networks of genes that regulate overt phenotypes
  - To capture the organization of genetic regulatory networks, we employ quantitative genetic analyses that identify the combinations of genetic loci that explain the trait
- Determining significance thresholds to constrain model traversal is computationally intensive
  - Significance thresholds can be acquired from permutation testing involving two-dimensional genome scans
    - Scale as

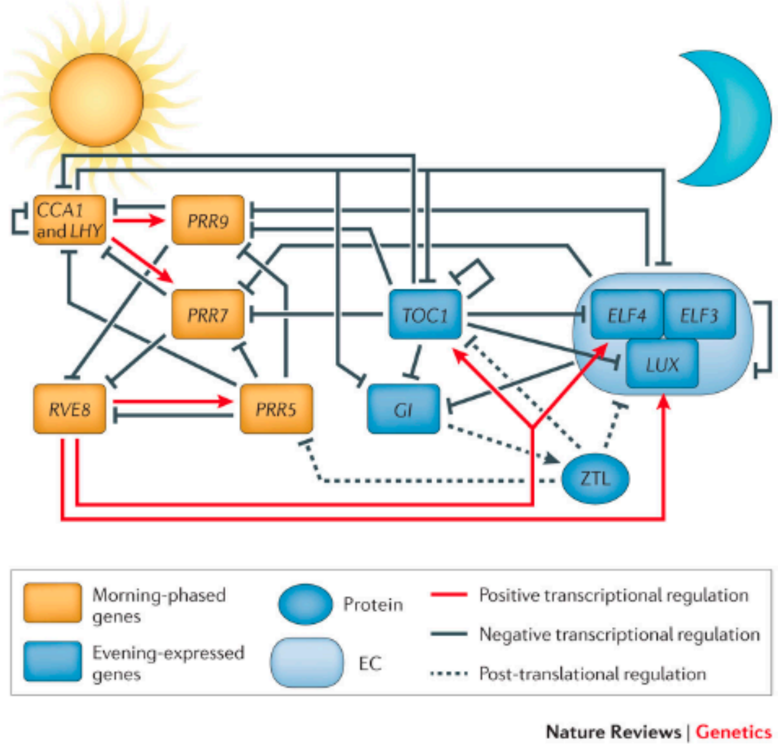
      $O(p(n(n-1)/2))$

      For p ~ 1,000 and n ~100,000
    - Highly parallel
- These analyses will enable characterization of the genetic networks underlying traits of economic importance, such as human health.

OAK RIDGE National Laboratory

GRN Background

Genetic regulatory networks

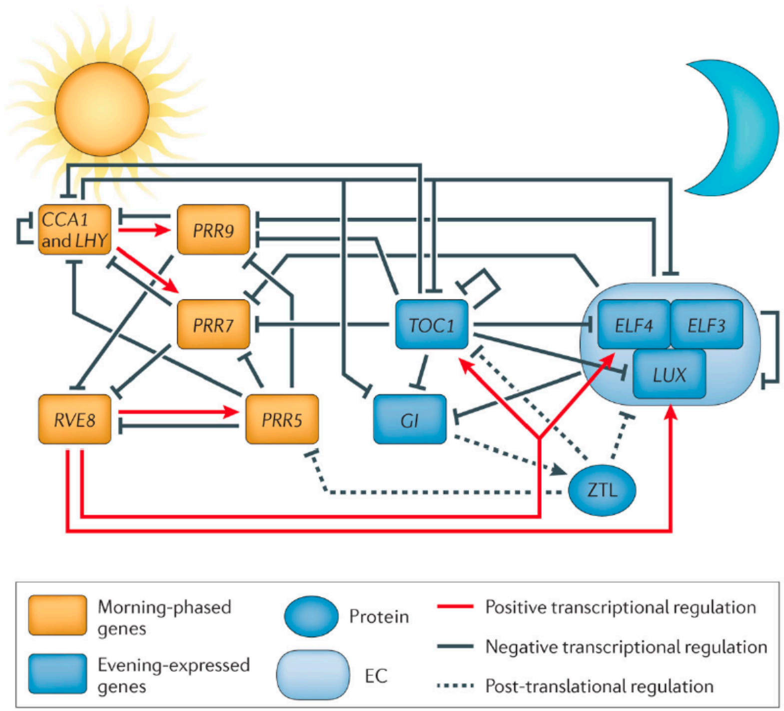


- Phenotypes, or traits, are determined by genetic regulatory networks.
- These genetic networks are composed of genes that are organized to coordinate overt phenotypes.
- In quantitative genetic analyses, genes underlying the basis of traits are formalized and studied as quantitative trait loci (QTL), or genomic loci whose alleles are correlated with trait differences.

Greenham & McClung. *Nature Reviews Genetics* (2015)

GRN Background

Genetic regulatory networks



Nature Reviews | Genetics

- For phenotype y and QTL Q_m, where m represents a locus in the genome, we want to determine $\beta_m \, \forall m$.

$$y = \sum_{i}^{m} \beta_i \boxed{Q_i} + \sum_{\substack{i,j \\ i \neq j}}^{m} \beta_{i,j} \boxed{Q_i} : \boxed{Q_j}$$

Greenham & McClung. *Nature Reviews Genetics* (2015)
Manichaikul et al. *Genetics* (2009)

GRN - Model selection

Model selection of QTL networks

$$y = \sum_{i}^{m} \beta_i \boxed{Q_i} + \sum_{\substack{i,j \\ i \neq j}}^{m} \beta_{i,j} \boxed{Q_i} : \boxed{Q_j}$$

- In the case of $\mathrm{m} = \{1, 2\}$ there are three different sized models to compare:

Terms	Model
1	$y = \bar{y}$
	$y = Q_1$
	$y = Q_2$
2	$y = Q_1 + Q_2$
3	$y = Q_1 + Q_2 + Q_1 : Q_2$

- Permutation tests of these models are used to determine empirical significance thresholds over which a locus or interaction between loci are considered to add sufficient information to the model.

Manichaikul et al. *Genetics* (2009)
Churchill & Doerge. *Genetics* (1994)

GRN - Permutation tests

Permutation tests with 2D genome scans

Input	Given data Y = $\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$ and `Q[m]` = $\begin{bmatrix} q_{m,1} \\ \vdots \\ q_{m,N} \end{bmatrix}$ where y_n is the phenotype of individual n $q_{m,n}$ is the genotype of individual n at marker m for individuals $n = 1, 2, \dots, N$ and markers $m = 1, 2, \dots, M$
Output	Generate `X[model]` = $\begin{bmatrix} \sigma_{\text{model}, 1} \\ \vdots \\ \sigma_{\text{model}, P} \end{bmatrix}$ where $\sigma_{\text{model}, p}$ is the best fit statistics of a model of permutation p for models = {one, add, full} and permutations $p = 1, 2, \dots, P$

Manichaikul et al. *Genetics* (2009)

OAK RIDGE National Laboratory

GBOOST

Original package: BOOST

BOolean Operation based Screening and Testing (BOOST)

A fast algorithm for detecting gene-gene interactions in genome-wide case-control studies by examining all pairwise interactions.

GBOOST is a GPU-implementation of BOOST based on the CUDA technology by Nvidia

URL: http://bioinformatics.ust.hk/BOOST.html#GBOOST

Hong Kong University of Science and Technology

GBOOST - Analysis details

Calculations are small and independent

- Analysis is made up of small, independent calculations that are readily subdivided
  - e.g., limit `PERMUTATIONS` per job, limit number of Markers per job
- Jobs are flexible with respect to the number of permutations and markers
  - i.e., all i, j pairs for a single permutation can be fit in the three models independently of other pairs

```
one = fit(y_perm = Q[i])
add = fit(y_perm = Q[i] + Q[j])
full= fit(y_perm = Q[i] + Q[j] +Q[i]:Q[j])
```

- Once a `PERMUTATION` is complete, the four resulting values for each i, j pair are compared and reduced to four values per permutation
  - `perm, one, add, full`
  - (one, add and full are each a statistic of the model fit)

Manichaikul et al. *Genetics* (2009)

GBOOST – Initial implementation

- Initial implementation on Titan / PanDA
  - Alzheimer Dataset
    - 359 individuals, 100K markers (in 23 chromosome files)
    - Tested for 10 test permutations (12 minutes per permutation)
    - Full test will perform (1000 permutations)
      - Experiment with multiple permutations per submission and multiple concurrent submissions
      - Each permutation requires creating a permuted copy of each chromosome file, output is a single file with interaction scores

Large datasets, software toolkit?

- Large publically available datasets
  - Alzheimer dataset
    - http://labs.med.miami.edu/myers/LFuN/data.html
  - gEUVADIS dataset (European 1000 Genomes project)
    - http://www.geuvadis.org/web/geuvadis/home
    - 412 individuals (278 cases, 143 controls)
    - Different Marker sizes (10K, 1M, 6M, 12M)
    - Discrete (categorical) features or phenotypes
      - analysis using GBOOST software
    - Continuous (quantitative) features
      - analysis using epiGPU software

gEUVADIS Project – 1000 Genomes

Oak Ridge National Laboratory

GBOOST analysis refinements

- Memory footprint
  - Original workflow loaded all data in memory, created shuffled (permuted) copies, then performed scatter / gather operations, for multiple permutations
    - Not scalable with larger datasets
    - Modified scheme splits up operations
      - stage_data operation preprocesses the input files, generates permuted files on disk
      - gboost_permutation operation submitted to backfill queue, one per permutation
- Modify GBOOST
  - for data streaming (shuffled files piped to GBOOST)
  - potential performance speedup

Generalized PanDA client interface

- Wrapper to incorporate all steps involved in analysis workflow
  - Check / Start / Stop / Restart pilot
  - Transformation file per analysis code
  - Set up client environment
  - Configure submission (code / transformation, inputs, parameters) – configuration file
  - Submit to PanDA backfill queue
  - Confirm success, parse results

Acknowledgements

- Ruslan Mashinistov
- Oral H. Sarp
- Veronica G. Vergara Larrea
- Dr. Jack Wells
- ORNL OLCF

OAK RIDGE National Laboratory