SAINT: Significance Analysis of INTeractome

Hyungwon Choi[*]

September 16, 2011

Abstract

SAINT implements Bayesian statistical inference for mixture modeling of label-free quantitative protein-protein interaction data [1, 2]. Model parameters are estimated by Markov chain Monte Carlo sampling algorithm written in C language and is thus suitable for use in a linux environment. The package was written for two different scenarios: (1) analysis without control IPs and (2) analysis with control IPs. Because of the history of the software development, we present two different implementations (with different data formats) for scenario (1).

1 Installation

A makefile is available for automatic installation. This software requires GNU Scientific Library for C language (any version is O.K.), freely downloadable from

`http://www.gnu.org/software/gsl/`

You will need to add the current installation directory to your shell login files such as `.cshrc` or `.bashrc` in order to run the command line at any location you want. One way to do this is to add the following lines to `.bashrc` in the home directory (∼/):

```
PATH=/home/hwchoi/projects/pepCount/saint:$PATH
```

2 Input File Format

To use SAINT, the interaction dataset must be prepared in one of the two formats: (i) matrix format or (ii) table format. The first format is for running the earlier implementation used in [1], and the latter is for more generic applications described in [2].

[*] For troubleshooting, contact the author at hwchoi@med.umich.edu.

2.1 Normalization Factors

Before we describe the file format, we first discuss the normalization factors (optionally) used in the spectral count data. SAINT incorporates normalization of spectral counts by protein length, other baseline abundance measure such as PeptideAtlas counts [3], and bait coverage (equivalent to the spectral count of bait itself in its own IP). The first normalization factor is required, whereas the latter two are optional (by a flag in the input command line). The following describes what each factor normalizes the spectral count for:

- Protein length l_i corrects the bias in spectral counting for the enrichment of sequencible peptides in longer proteins
- PeptideAtlas count a_i corrects the bias for naturally abundant proteins
- Bait coverage c_j corrects for the possible boosting of spectral counts in IPs where baits are extremely over-expressed

where i and j index preys and baits respectively. These factors are reflected in the SAINT model by division of counts. Specifically, the spectral count of interaction between prey i and bait j is expressed as:

$$\log X_{ij} = \log l_i + \log a_i + \log c_j + \beta_0 + \alpha_{ij} + \epsilon_{ij} \tag{1}$$

with ϵ_{ij} following a certain error distribution. This equation is equivalent to

$$X'_{ij} = \frac{X_{ij}}{l_i \cdot a_i \cdot c_j} = \beta_0 + \alpha_{ij} + \epsilon_{ij}. \tag{2}$$

2.2 Matrix Data Format

We describe matrix format first (in one consolidated, tab-delimited file). See Table 1 for an example with 3 baits, each purified twice. If each bait was IP'ed once, then the first two lines shall be identical. When there are replicates of the same bait IP, the column labels must be carefully named when filling in the data.

- The first three lines of the input file must be the following: (i) unique name for each IP, (ii) unique bait names for each IP, and (iii) bait coverage.
- The rest of the table lists preys identified in the data, corresponding normalization factors for preys (PeptideAtlas counts and length), and spectral counts for interactions.
- The first twelve cells (3×4) in the upper left corner of the data (spanning three rows and four columns) can be filled with any values as long as a tab delimiter is placed between the entries.
- For each prey, the first four entries are: (i) unique name for prey, (ii) Peptide Atlas counts, (iii) length of prey (amino acid count), and (iv) prey type. If Peptide Atlas counts are unknown, fill number 1 for it. As these numbers are logged later,

			IP	A1	A2	B1	B2	C1	C2
			Bait	A	A	B	B	C	C
Preys	PepAtlas	Length	PreyType\BaitCov	26	16	167	54	140	153
PROT1	7	188	C	19	12	4	7	24	16
PROT2	40	157	N	1	0	0	0	1	0
PROT3	9	723	C	47	9	21	18	57	24
PROT4	9	186	R	29	6	10	7	14	15
PROT5	1564	988	N	1	0	0	0	0	0
PROT6	10463	417	N	2	1	0	0	9	0
PROT7	386	175	N	23	19	0	0	3	2
PROT8	1459	166	N	1	0	0	0	4	3
PROT9	433	200	N	1	0	1	1	12	5
PROT10	2658	363	N	3	0	7	1	27	4
PROT11	44	1179	N	25	29	0	0	0	0
PROT12	58	373	N	9	10	0	0	0	0
PROT13	36	279	N	4	5	0	0	0	0
PROT14	173	259	N	6	3	0	0	0	0
PROT15	101	808	N	0	0	0	1	0	0
PROT16	47	412	N	0	0	0	0	99	101
PROT17	17	393	N	0	0	0	0	15	70

Table 1: Sample matrix format required for SAINT.

having non-positive number incurs `NaN`'s in the calculation. Prey type can be one of the three choices: `C` for known contaminants, `R` for known non-contaminants (especially hubs), and `N` for all other proteins. This option is critical for differentiating real hub proteins from frequently appearing contaminants.

2.3 Table Data Format

Table formatted data must be prepared in three files: (i) prey table, (ii) bait table, and (iii) interaction table (all tab or space delimited).

The current version of `SAINT` using the table formatted data takes protein length only for normalization purposes. Therefore, the prey table should contain two columns, prey names and their sequence length. The bait table should list three columns, IP name, bait name, and the indicator for experimental and control IPs (T = experimental, C = control). The interaction table should contain four fields, IP name, bait name, prey name, and spectral count. Note that, if the same pair of interactions appears in replicate IPs, then the pair appears in multiple lines. In this table, preys that appear in control IPs only (not in experimental IPs) should be excluded in the dataset (There is a command for data preprocessing which can do this for you!). See Tables 2, 3, and 4 below for an example.

PROT1	188
PROT2	157
PROT3	723
PROT4	186
⋮	⋮

Table 2: Sample prey table format.

A1	A	T
A2	A	T
B1	B	T
B2	B	T
⋮	⋮	⋮
ctrl1	ctrl1	C
ctrl2	ctrl2	C
ctrl3	ctrl3	C

Table 3: Sample bait table format in small-scale datasets. The last three rows are shown in case that the control IP data are available.

A1	A	PROT1	19
A1	A	PROT2	1
A1	A	PROT3	47
A2	A	PROT1	12
A2	A	PROT3	9
B1	B	PROT1	4
B1	B	PROT3	21
B2	B	PROT1	7
B2	B	PROT3	18
C1	C	PROT1	24
C1	C	PROT2	1
C1	C	PROT3	57
C2	C	PROT1	16
C2	C	PROT3	24

Table 4: Sample interaction table in small-scale datasets. Interactions with zero count in the matrix data are not listed in this format, allowing an economical listing of data.

3 Data without control IPs - old version [1]

To run `SAINT` for a (large-scale, matrix-formatted) dataset without control purification, use the following command line:

```
[hwchoi@gouda pepCount]$ saint-spc-noctrl-matrix
usage: saint-spc-noctrl-matrix [data] [output] [nburn] [niter] [ff]
       saint-spc-noctrl-matrix [data] [output] [nburn] [niter] [ff]
                                              [abun] [len] [cov]
```

The first five arguments are required, and the last three are optional. We describe each argument below.

- `data`: matrix formatted data
- `output`: the prefix for all output file names
- `nburn`: number of burn-in period in the Gibbs sampling, normally $1,000 \sim 2,000$
- `niter`: number of iterations in the Gibbs sampling normally $10,000 \sim 20,000$
- `ff`: empirical frequency threshold ($[0,1]$, e.g. 0.1 (10%) in the kinome data [1])
- `abun`: 0/1 indicator for abundance normalization of each prey (a_i)
- `len`: 0/1 indicator for sequence length normalization of each prey (l_i)
- `cov`: 0/1 indicator for bait coverage normalization of each bait (c_j)

`SAINT` reports probabilities in the same matrix format. This new matrix, however, lists unique baits in the columns because the algorithm computes the probability for a unique bait-prey pair averaging over the evidence in the replicates.

4 Data without control IPs [2]

Before running the new version, a quick data reformatting step is required. The function `saint-reformat` adds zero counts for the following two cases. For one, it adds zero counts to those bait-prey pairs not reproduced in all replicate IPs. If a prey was found in one of three replicate IPs for a bait, then two zeros will be added. For the other, it adds zero counts to preys in control IPs. A zero count in control IPs is an important piece of information. Likewise a zero count in the experiment IPs means absence of interaction, which gives information for reproducibility in calculating the probability score.

Furthermore, `saint-reformat` removes redundant information from user-prepared input datasets. For example, if the user provides preys that do not appear in the interaction data, then those preys will be removed. Also, if interaction list (IP-prey pair) is duplicated, the first entry from the data will be taken and the rest will be discarded. However, if IP names are not unique, then the program quits and prompts the user to fix the bait file. The same shall happen when interaction file contains prey, bait, IP names that are not in the prey and bait files.

The command line is as follows.

```
[hwchoi@gouda pepCount]$ saint-reformat
usage: saint-reformat [interactionfile] [baitfile] [preyfile]
usage: saint-reformat [interactionfile] [baitfile] [preyfile] [# control IPs]
```

- `interactionfile`: interaction table data
- `baitfile`: bait table data
- `preyfile`: prey table data
- `# control IPs`: not relevant for datasets without control IPs

As a result of this run, three new files shall be generated: `interaction.new`, `prey.new`, and `bait.new`. Then `SAINT` can be run:

```
[hwchoi@gouda pepCount]$ saint-spc-noctrl
usage: saint-spc-noctrl [interactionfile] [preyfile] [baitfile] [nburn] [niter]
                          [fthres] [fgroup] [var]
```

- `interactionfile`: interaction table data (`interaction.new`)
- `preyfile`: prey table data (`prey.new`)
- `baitfile`: bait table data (`bait.new`)
- `nburn`: number of burn-in period in the Gibbs sampling (at least 2,000 suggested)
- `niter`: number of iterations in the Gibbs sampling (at least 10,000 suggested).
- `fthres`: frequency threshold for preys above which probability is set to 0 in all IPs
- `fgroup`: frequency boundary dividing high and low frequency groups
- `var`: binary [0/1] indicating whether variance of count data distributions should be modeled or not

For successful filtering, the choice of `fthres` and `fgroup` is critical. We are currently working on a guideline to choose this value.

5 Data with control IPs [2]

To run SAINT with control data, one can run the reformat operation described in the previous section. In this step, datasets with excessively many control IPs can be reformatted in a compact form. Having too many control IPs leads to poor filtering because many contaminants do not appear consistently over as many experiments (while some do). Hence one can take k largest spectral counts for each prey from control IPs by specifying k as the last argument in saint-reformat command (default $k = 5$).

Once reformatting is done, type in the command below:

```
[hwchoi@gouda pepCount]$ saint-spc-ctrl
usage: saint-spc-ctrl [interactionfile] [preyfile] [baitfile] [nburn] [niter]
                                                 [lowMode] [minFold] [normalize]
```

- interactionfile: interaction table data
- preyfile: prey table data
- baitfile: bait table data
- nburn: number of burn-in period in the Gibbs sampling (at least 2,000 suggested)
- niter: number of iterations in the Gibbs sampling (at least 10,000 suggested).
- lowMode: minimize the impact of extremely high count interactions on the scoring of low count interactions (recommended 1).
- minFold: force separation of positive and negative distribution when there are few data (recommended 1)
- normalize: divide spectral counts by the total spectral counts of each IP (recommended 0).

SAINT reports probabilities in the table format as well, next to the field of spectral counts.

6 Other types of quantitative measures in SAINT

The description so far has been limited to the datasets with spectral counts, and thus the software cannot be applied to other types of quantitative measures, such as MS1 intensity measurement of precursor ions. In order to address this, a version for continuous measurements (non-discrete as counts) was recently added. The data are log-transformed and 0 values are treated as missing data (missing at random, or MAR). Missing data for repeatedly measured interactions are imputed according to their posterior distribution, under simple prior distribution concentrated in the low abundance range.

Input format is identical to the spectral count data. The only exception is that the prey file should not include anything other than the prey names (no length). Command lines are two steps again:

```
[hwchoi@gouda pepCount]$ saint-reformat
usage: saint-reformat [interactionfile] [baitfile] [preyfile]
usage: saint-reformat [interactionfile] [baitfile] [preyfile] [# control IPs]
```

which first imputes missing data and cleans up duplicate entries, followed by

```
[hwchoi@gouda pepCount]$ saint-int-ctrl
usage: saint-int-ctrl [interactionfile] [preyfile] [baitfile] [nburn] [niter]
```

7 Output from **SAINT**

Main output files can be found in the folder "RESULT" for most of the commands (except `saint-spc-noctrl-matrix`). In the folder, there are `interaction` and `unique_interaction` files. These files list the raw input data with an additional column of estimated probability of true interaction. As the file names indicate, the former lists all observed interactions repeating over the replicates, and the latter lists only the unique bait-prey pairs. In most cases, the users are presumably interested in the latter file. Having both lists, however, saves the effort for parsing unique list into replicate-level data and vice versa. In order to facilitate the global view of the data, we also provide matrix format data arranged.

References

[1] A. Breitkreutz *et al.* A Global Protein Kinase and Phosphatase Interaction Network in Yeast. *Science*, 328:1043–1046, 2010.

[2] H. Choi *et al.* Significance Analysis of INTeractome (SAINT): Probabilistic Scoring of Affinity Purification - Mass Spectrometry Data. *Nat. Methods*, 8:70–73, 2011.

[3] F. Desiere, E.W. Deutsch, N.L. King, A.I. Nesvizhskii, P. Millick, J. Eng, S. Chen, J. Eddes, S.N. Loevenich, and R. Aebersold. The PeptideAtlas project. *Nucleic Acids Res.*, 34:D655–658, 2006.