Vignette for `SAINT`

Significance Analysis of INTeractome using Label-free Quantitative AP-MS Data

Hyungwon Choi[*]

January 15, 2010

Abstract

This vignette explains the software package `SAINT`. `SAINT` is based on a Markov chain Monte Carlo sampling algorithm written in C language and is thus suitable for use in a linux environment. The package contains a parser that converts a table-formatted data to a bait-prey matrix and vice versa, and estimation algorithm for two different situations: (1) analyzing a large-scale dataset without negative control IPs and (2) analyzing a small-scale dataset with negative control IPs.

Installation

A makefile will be added for automatic system-wide installation (coming soon). This software requires GNU C Scientific Library (any version is O.K.), freely downloadable from

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

For now, use `compile` script for a quick local installation in each folder. 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. In my case, the file `.bashrc` has the following lines:

```
PATH=/usr/local/bin/coplot:$PATH
PATH=/home/hwchoi/projects/pepCount/saint:$PATH
```

Input File Format

To use `SAINT`, the interaction dataset must be prepared in one of the two formats: (i) bait-prey data matrix format and (ii) table format.

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

Normalization Factors

Before we describe the file format, we first discuss the normalization factors (optionally) used in the spectral count data. As discussed in our paper [1], SAINT incorporates normalization by protein length, other baseline abundance measure such as PeptideAtlas counts [2], and bait coverage (equivalent to the spectral count of bait itself in its own IP). The first normalization factor is required, whereas the other 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 in the cell
- Bait coverage c_j corrects for the possible enrichment of spectral counts in the baits that are more compatible with the IP protocol, assuming that the bait count reflects the quality of IP

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

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

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

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

Bait-Prey Matrix Format

We describe the bait-prey matrix format first (in a tab-delimited file), which is used for the implementation for large-scale datasets. 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 followed when filling in the data.

- The first four lines of the input file must be the following: (i) unique name for each IP, (ii) unique bait names for each IP, (iii) tag type, and (iv) bait coverage. The third line was included for the kinome data analysis [3], where multiple tags were used to profile the overlapping set of baits. This field is a nuisance, and is planned to be deleted in the future (however you must keep this line for now).
- The rest of the table lists preys identified in the data, corresponding normalization factors (PeptideAtlas counts and length), and spectral counts for interactions.

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

Table 1: A sample bait-prey matrix format required for SAINT in large-scale datasets. By large-scale, we mean a minimum of 20∼30 baits.

- The 16 cells in the upper left corner of the data (spanning four rows and columns) can be filled anything as long as a tab delimiter is present between entries.

Table Format

Secondly, a more common data format is the table format. This format is divided into three files: (i) prey table, (ii) bait table, and (iii) interaction table.

In the current version of SAINT, it takes protein length only for the normalization of preys and 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. In this table, preys that appear in control IPs but not in experimental IPs should be excluded in the dataset. See Tables 2, 3, and 4 for an example that converts a portion of Table 1.

PROT1	188
PROT2	157
PROT3	723
PROT4	186
⋮	⋮

Table 2: A sample prey table format for SAINT in small-scale datasets.

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

Table 3: A sample bait table format for SAINT 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: A sample interaction table for SAINT in small-scale datasets. This table is a conversion of the first three rows of the bait-prey matrix data in Table 1. Interactions with zero count in the matrix data are not listed in this format, allowing an economical listing of data.

Large-scale Data without Control IPs

To run the implementation for a large scale dataset, prepare a bait-prey matrix data and use the following command line:

```
[hwchoi@gouda pepCount]$ saint-spc-large
usage: saint [data] [output] [nburn] [niter] [ff]
       saint [data] [output] [nburn] [niter] [ff] [abundance] [length] [coverage]
```

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

- `data`: bait-prey matrix formatted data
- `output`: the prefix for all output file names
- `nburn`: number of burn-in period in the Gibbs sampling
- `niter`: number of iterations in the Gibbs sampling
- `ff`: empirical frequency threshold
- `abundance`: 0/1 indicator for abundance normalization of each prey (a_i)
- `length`: 0/1 indicator for sequence length normalization of each prey (l_i)
- `coverage`: 0/1 indicator for bait coverage normalization of each bait (c_j)

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

Small-scale Data with Control IPs

In order to run SAINT for small-scale experiments, a quick intermediate step needs to be completed. The function `saint-append` adds zero counts for the following two cases. For one, it adds zero counts to those bait-prey pairs that were not replicated in all 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, whereas a zero count in the experiment IPs means absence of interaction, for which we do not have to calculate the probability score. The command line is as follows.

```
[hwchoi@gouda pepCount]$ saint-append
usage: saint-append [interactionfile] [baitfile]
```

- `interactionfile`: interaction table data

- `baitfile`: bait table data

Given this is done, one can run the main SAINT algorithm using the command line below.

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

- `interactionfile`: interaction table data
- `preyfile`: prey table data
- `baitfile`: bait table data
- `nburn`: number of burn-in period in the Gibbs sampling
- `niter`: number of iterations in the Gibbs sampling

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

Output from SAINT

Due to an issue related to the release of data, this section will be added upon publication of the paper.

References

[1] H. Choi et al. Probabilistic Scoring of Protein-Protein Interactions using Label-free Quantitative AP-MS Data with an Efficient Statistical Model. In Preparation.

[2] 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.

[3] A. Breitkreutz et al. Global architecture of the yeast kinome interaction network. Submitted.