Granger Causality Analysis of Sparse Hodgkin Huxley Neural Network and Its Relatives

1 Pairwise Regression v.s. Joint Regression: Numerical Results

Table 1: Common Parameters

S^{EE}, S^{IE}	S^{EI}, S^{II}	μ	f	T	Sample Rate
0.05 (0.85 mV)	0.09 (−0.45 mV)	1.0 kHz	0.03 (0.93 mV)	1×10^6	2.0 kHz

All edge judgements are made by $P = 1 \times 10^{-5}$.

Table 2: Basic statistics, n=50

n	edges	firing rate Hz	data	od	aveGC0 / Ideal (10^{-4})		Correctness
				bic/aic	joint	pairwise	Joint / pair
30+20	129 / 2450	11.8 ± 0.4	volt	27 / 31	0.15 / 0.15	0.15 / 0.15	100% / 100%
			ST	1 / 40	0.20 / 0.20	0.20 / 0.20	100% / 100%
30+20	270 / 2450	11.9 ± 0.4	volt	27 / 31	0.15 / 0.15	0.16 / 0.15	100% / 100%
			ST	1 / 40	0.20 / 0.20	0.20 / 0.20	100% / 100%
30+20	398 / 2450	12.0 ± 0.6	volt	27 / 31	0.15 / 0.15	0.16 / 0.15	100% / 99.9%
			ST	1 / 40	0.20 / 0.20	0.20 / 0.20	100% / 100%
30+20	527 / 2450	12.2 ± 0.6	volt	27 / 31	0.15 / 0.15	0.18 / 0.15	100% / 99.3%
			ST	1 / 40	0.19 / 0.20	0.22 / 0.20	100% / 99.8%

Table 3: Basic statistics, n=100

n	edges	firing rate Hz	data	od	aveGC0 / Ideal (10^{-4})		Correctness
				bic/aic	joint	pairwise	Joint / pair
80+20	515 / 9900	12.5 ± 0.5	volt	20 / 30	0.15 / 0.15	0.15 / 0.15	100% / 99.9%
			ST	1 / 40	0.20 / 0.20	0.20 / 0.20	100% / 100%
80+20	1010 / 9900	13.2 ± 0.8	volt	21 / 30	0.15 / 0.15	0.17 / 0.15	100% / 99.6%
			ST	1 / 40	0.20 / 0.20	0.21 / 0.20	100% / 99.9%
80+20	1507 / 9900	14.1 ± 1.0	volt	21 / 30	0.15 / 0.15	0.24 / 0.15	100% / 91.9%
			ST	1 / 40	0.20 / 0.20	0.25 / 0.20	100% / 98.2%
80+20	1987 / 9900	14.9 ± 1.2	volt	21 / 30	0.15 / 0.15	0.51 / 0.15	100% / 47.2%
			ST	1 / 40	0.20 / 0.20	0.39 / 0.20	100% / 77.6%

Table 4: Basic statistics, n=200

n	edges	firing rate Hz	data	od	aveGC0 / Ideal (10^{-4})		Correctness
				bic/aic	joint	pairwise	Joint / pair
180+20	2080 / 39800	13.9 ± 0.9	volt	7 / 30	0.15 / 0.15	0.16 / 0.15	100% / 99.9%
			ST	1 / 1	0.01 / 0.00	0.01 / 0.00	94.8% / 94.8%
			ST	40	0.20 / 0.20	0.21 / 0.20	100.0% / 100.0%
180+20	4077 / 39800	16.8 ± 1.5	volt	12 / 29	0.15 / 0.15	0.45 / 0.15	100% / 46.5%
			ST	1 / 1	0.01 / 0.00	0.02 / 0.00	89.1% / 88.7%
			ST	40	0.20 / 0.20	0.37 / 0.20	100.0% / 84.8%
180+20	6086 / 39800	23.3 ± 2.3	volt	16 / 30	0.22 / 0.15	22.96 / 0.15	96.2% / 15.3%
			ST	1 / 1	0.11 / 0.00	0.61 / 0.00	52.0% / 15.5%
			ST	40	0.24 / 0.20	12.65 / 0.20	99.4% / 15.3%
180+20	8047 / 39800	33.0 ± 1.3	volt	23 / 39	0.33 / 0.20	176.05 / 0.20	84.8% / 20.2%
			ST	1 / 1	0.23 / 0.00	6.75 / 0.00	38.5% / 20.2%
			ST	40	0.33 / 0.20	102.30 / 0.20	85.3% / 20.2%

Note: ST 40 use pval=1e-6

Table 5: Basic statistics, n=400

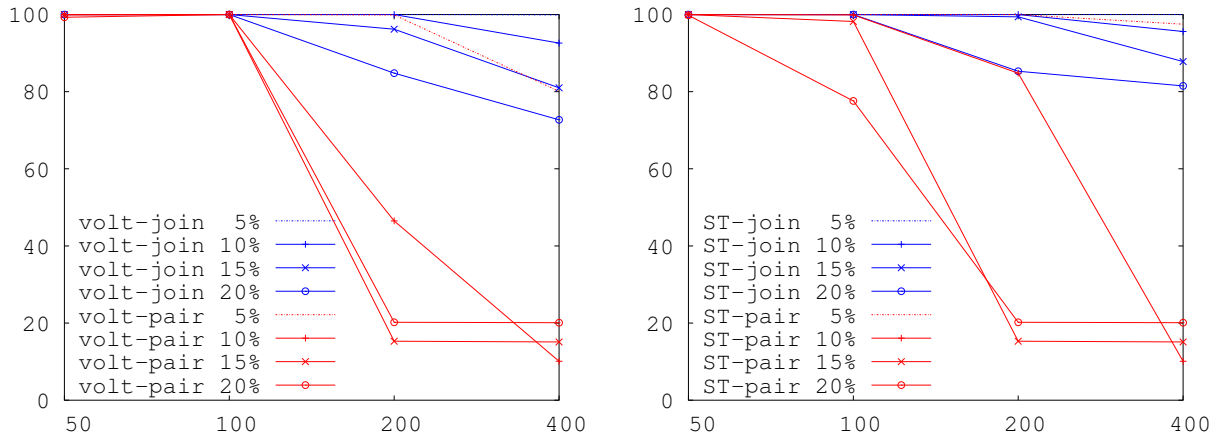
n	edges	firing rate Hz	data	od	aveGC0 / Ideal (10^{-4})		Correctness
				bic/aic	joint	pairwise	Joint / pair
380+20	8113 / 159600	17.8 ± 1.5	volt	7 / 29	0.15 / 0.15	0.29 / 0.15	99.9% / 80.1%
			ST	1 / 1	0.01 / 0.00	0.01 / 0.00	94.8% / 94.7%
			ST	40	0.20 / 0.20	0.28 / 0.20	100.0% / 97.5%
380+20	16111 / 159600	35.5 ± 0.4	volt	19 / 35	0.23 / 0.18	260.50 / 0.18	92.6% / 10.1%
			ST	1 / 1	0.10 / 0.00	11.84 / 0.00	62.0% / 10.1%
			ST	40	0.23 / 0.20	157.21 / 0.20	95.6% / 10.1%
380+20	24098 / 159600	37.04 ± 0.08	volt	17 / 33	0.32 / 0.17	426.39 / 0.17	81.0% / 15.1%
			ST	1 / 1	0.16 / 0.00	43.36 / 0.00	56.0% / 15.1%
			ST	40	0.30 / 0.20	274.47 / 0.20	87.8% / 15.1%
380+20	32148 / 159600	37.17 ± 0.06	volt	16 / 31	0.41 / 0.16	539.83 / 0.16	72.7% / 20.1%
			ST	1 / 1	0.22 / 0.00	76.86 / 0.00	52.4% / 20.1%
			ST	40	0.37 / 0.20	358.26 / 0.20	81.5% / 20.1%

Table 6: parameters for $n = 1000$

S^{EE}, S^{IE}	S^{EI}, S^{II}	μ	f	T	Sample Rate
0.04 (0.68 mV)	0.09 (-0.35 mV)	0.7 kHz	0.03 (0.93 mV)	1×10^6	2.0 kHz

Table 7: Basic statistics, n=1000,pval=1e-7

n	edges	firing rate Hz	data	od	aveGC0 / Ideal (10^{-4})		Correctness
				bic/aic	joint	pairwise	Joint / pair
750+250	24958 / 999000=2.5%	8.33 ± 0.49	volt	40 (1/1)	0.20 / 0.20	0.20 / 0.20	100.0% / 100.0%
			ST	40	0.20 / 0.20	0.20 / 0.20	99.6% / 99.6%
750+250	50138 / 999000=5%	10.1 ± 0.9	volt	40 (1/1)	0.20 / 0.20	0.35 / 0.20	100.0% / 85.4%
			ST	40	0.20 / 0.20	0.25 / 0.20	99.6% / 99.1%
750+250	99815 / 999000=10%	32.1 ± 0.6	volt	40 (7/27)	0.20 / 0.20	308.79 / 0.20	98.0% / 10.0%
			ST	40	0.20 / 0.20	208.90 / 0.20	98.3% / 10.0%
750+250	150235 / 999000=15%	33.6 ± 0.1	volt	40 (7/26)	0.22 / 0.20	480.68 / 0.20	93.7% / 15.0%
			ST	40	0.22 / 0.20	374.05 / 0.20	93.5% / 15.0%
750+250	200013 / 999000=20%	33.92 ± 0.08	volt	40 (7/25)	0.26 / 0.20	586.02 / 0.20	88.8% / 20.0%
			ST	40 (1/1)	0.24 / 0.20	480.63 / 0.20	86.5% / 20.0%



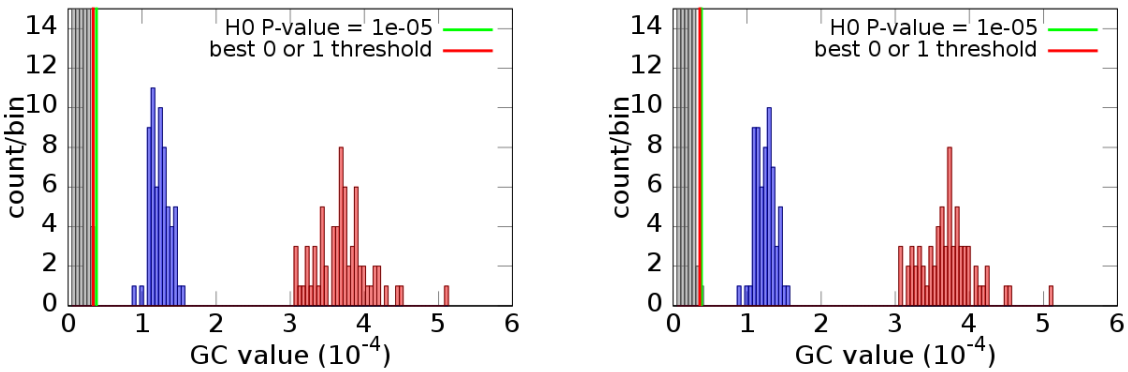
(a) Volt

(b) Spike Train

Figure 1: Summary: correctness comparison

The drop of spike train join GC can be largely improved by manually set fitting order (to 40).

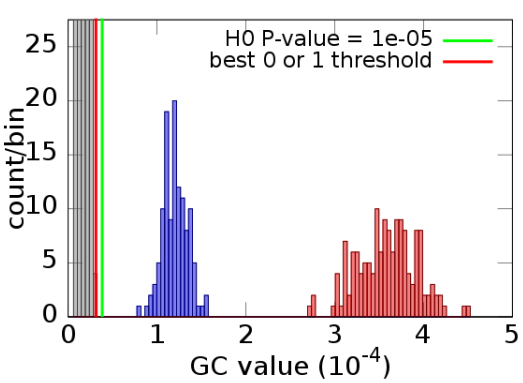
1.1 n=30E+20I, volt



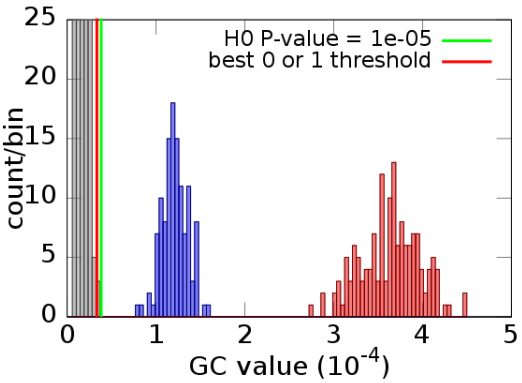
(a) Joint. over guess: 0, lack guess: 0.

(b) Pairwise. over guess: 0, lack guess: 0.

Figure 2: n=30E+20I, sparseness: 5%, volt

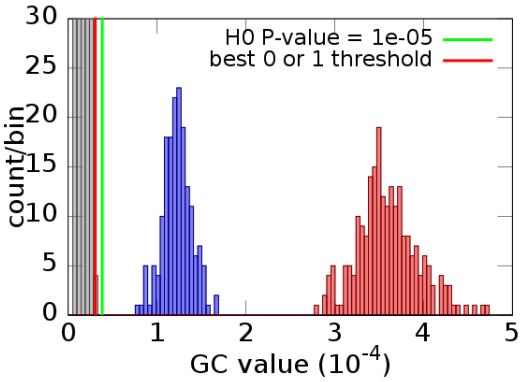


(a) Joint. over guess: 0, lack guess: 0.

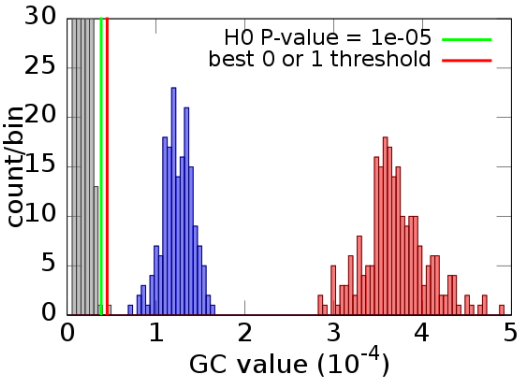


(b) Pairwise. over guess: 0, lack guess: 0.

Figure 3: n=30E+20I, sparseness: 10%, volt

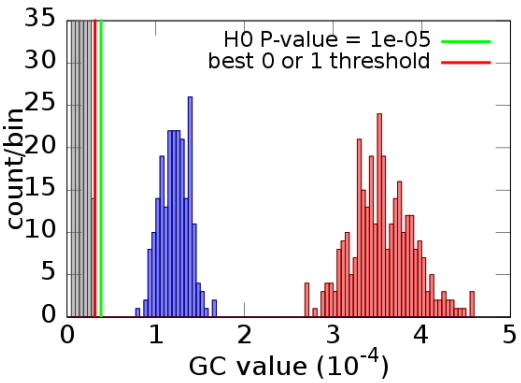


(a) Joint. over guess: 0, lack guess: 0.

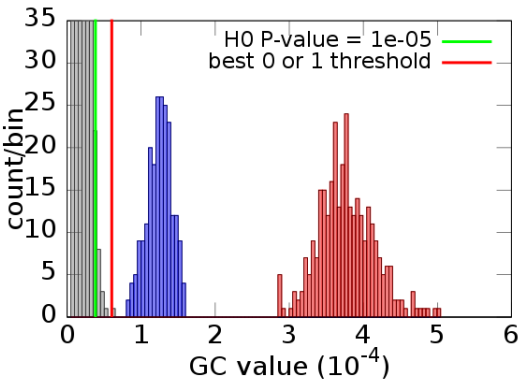


(b) Pairwise. over guess: 1, lack guess: 0.

Figure 4: n=30E+20I, sparseness: 15%, volt



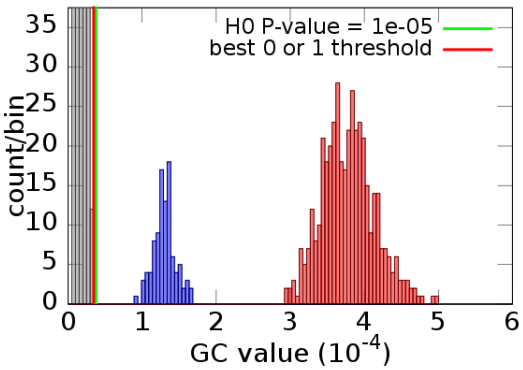
(a) Joint. over guess: 0, lack guess: 0.



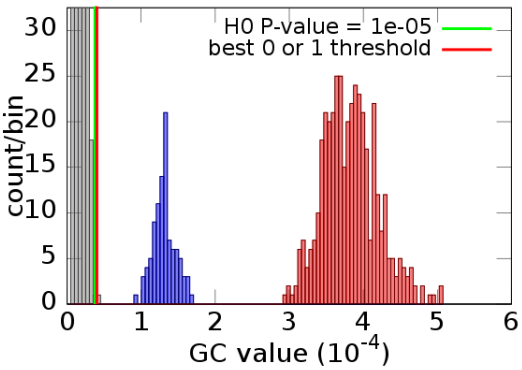
(b) Pairwise. over guess: 18, lack guess: 0.

Figure 5: n=30E+20I, sparseness: 20%, volt

1.2 n=80E+20I, volt

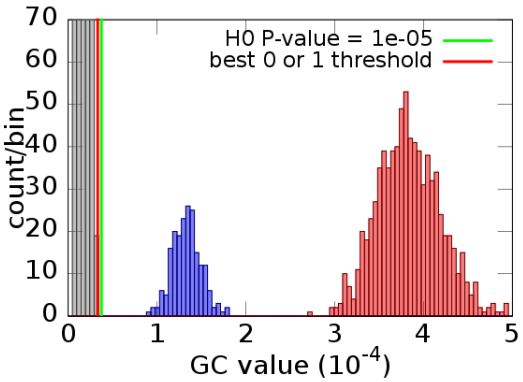


(a) Joint. over guess: 0, lack guess: 0.

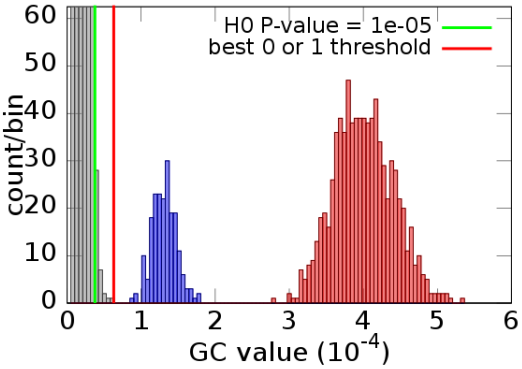


(b) Pairwise. over guess: 1, lack guess: 0.

Figure 6: n=80E+20I, sparseness: 5%, volt

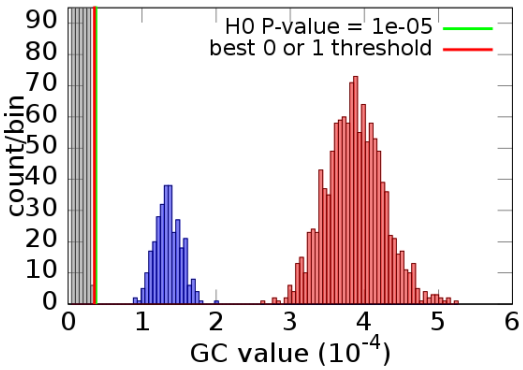


(a) Joint. over guess: 0, lack guess: 0.

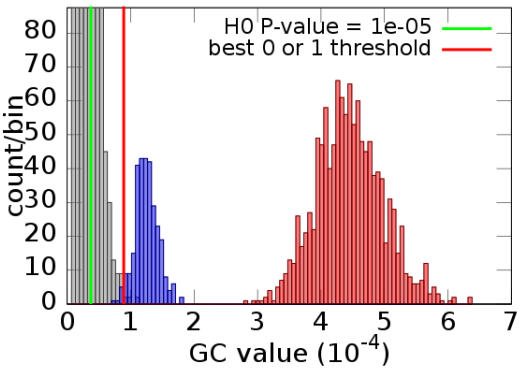


(b) Pairwise. over guess: 36, lack guess: 0.

Figure 7: n=80E+20I, sparseness: 10%, volt

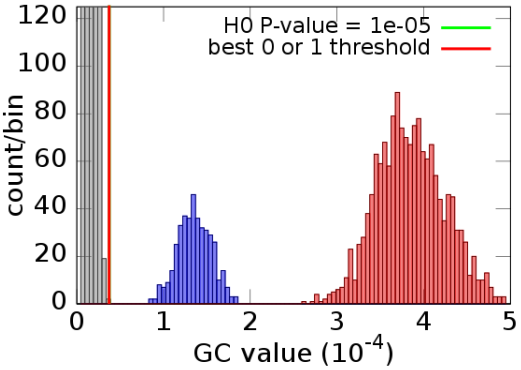


(a) Joint. over guess: 0, lack guess: 0.

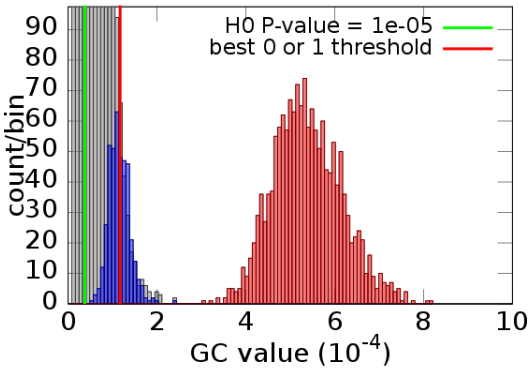


(b) Pairwise. over guess: 806, lack guess: 0.

Figure 8: n=80E+20I, sparseness: 15%, volt



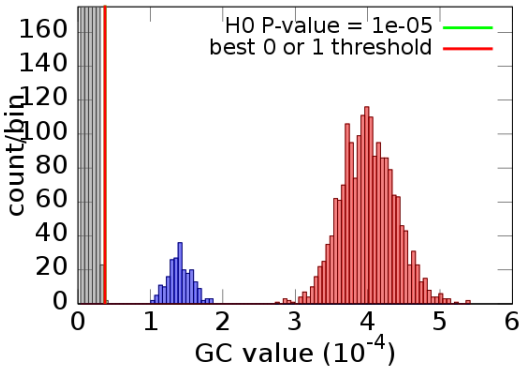
(a) Joint. over guess: 0, lack guess: 0.



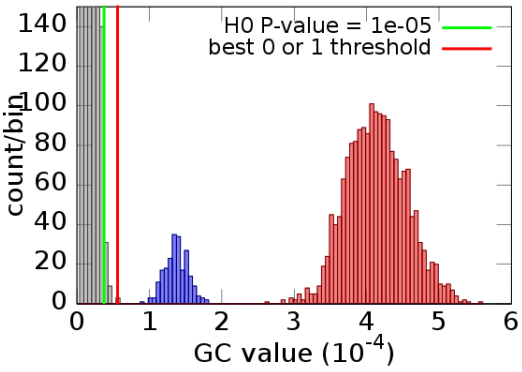
(b) Pairwise. over guess: 5221, lack guess: 0.

Figure 9: n=80E+20I, sparseness: 20%, volt

1.3 n=180E+20I, volt

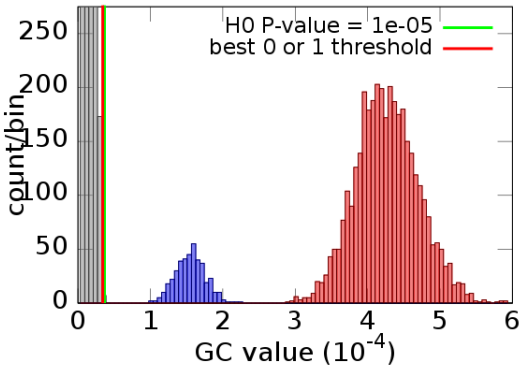


(a) Joint. over guess: 0, lack guess: 0.

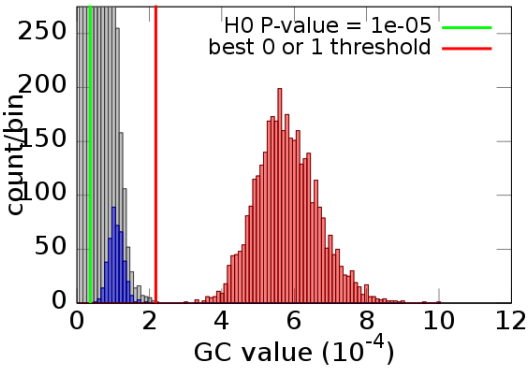


(b) Pairwise. over guess: 37, lack guess: 0.

Figure 10: n=180E+20I, sparseness: 5%, volt

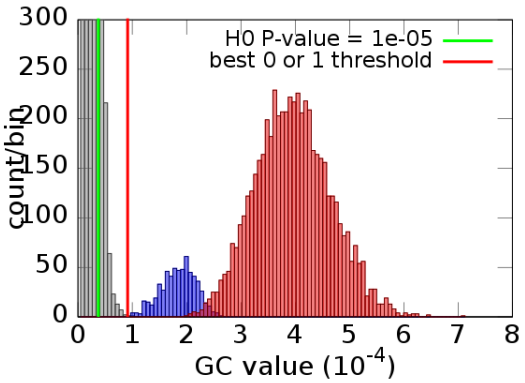


(a) Joint. over guess: 0, lack guess: 0.

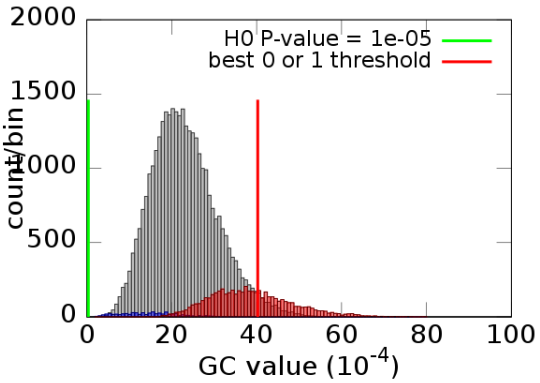


(b) Pairwise. over guess: 21290, lack guess: 0.

Figure 11: n=180E+20I, sparseness: 10%, volt

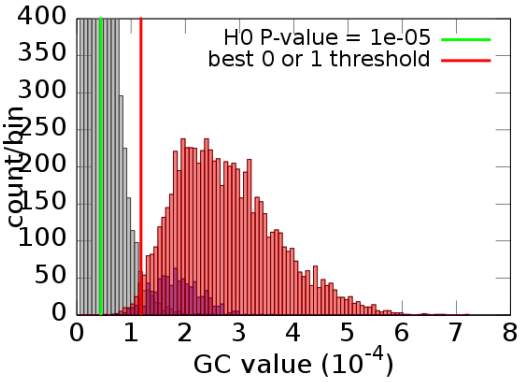


(a) Joint. over guess: 1479, lack guess: 0.

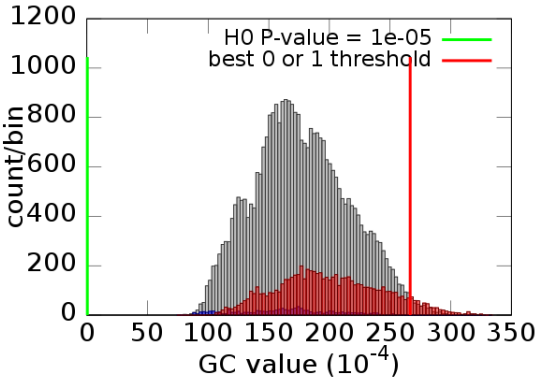


(b) Pairwise. over guess: 33714, lack guess: 0.

Figure 12: n=180E+20I, sparseness: 15%, volt



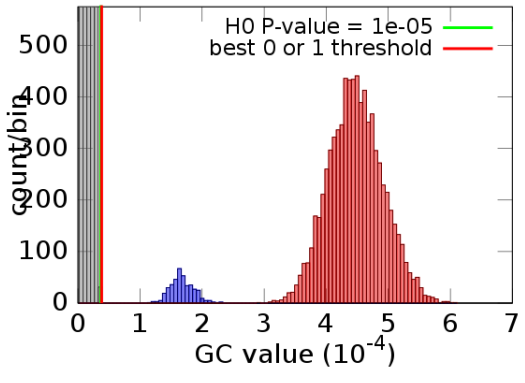
(a) Joint. over guess: 6062, lack guess: 0.



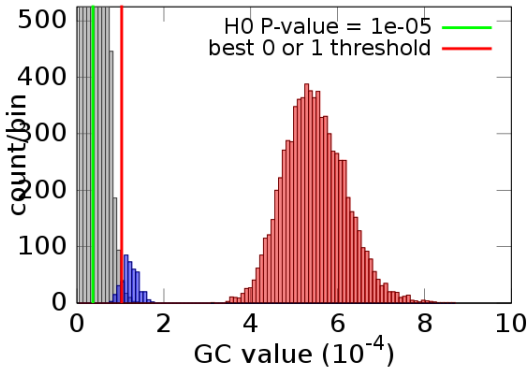
(b) Pairwise. over guess: 31753, lack guess: 0.

Figure 13: n=180E+20I, sparseness: 20%, volt

1.4 n=380E+20I, volt

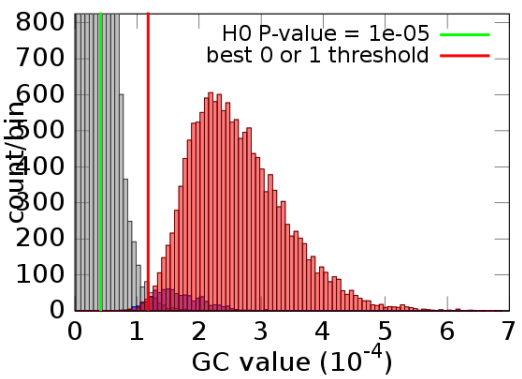


(a) Joint. over guess: 3, lack guess: 0.

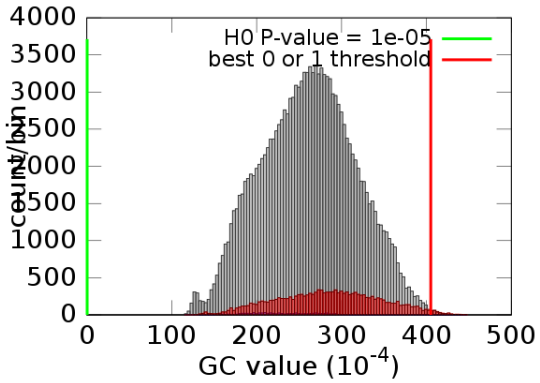


(b) Pairwise. over guess: 31805, lack guess: 0.

Figure 14: n=380E+20I, sparseness: 5%, volt

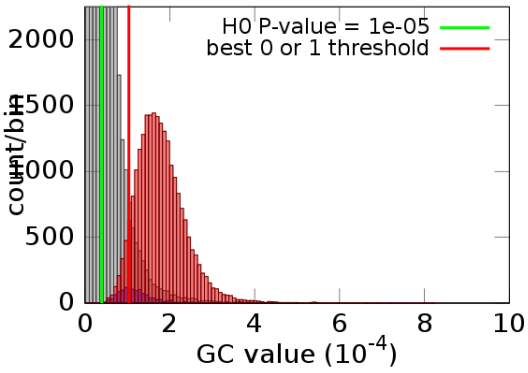


(a) Joint. over guess: 11857, lack guess: 0.

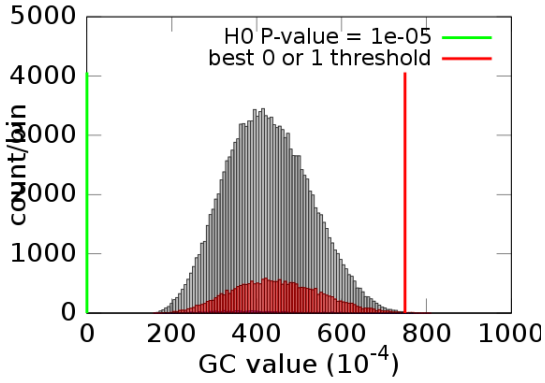


(b) Pairwise. over guess: 143489, lack guess: 0.

Figure 15: n=380E+20I, sparseness: 10%, volt

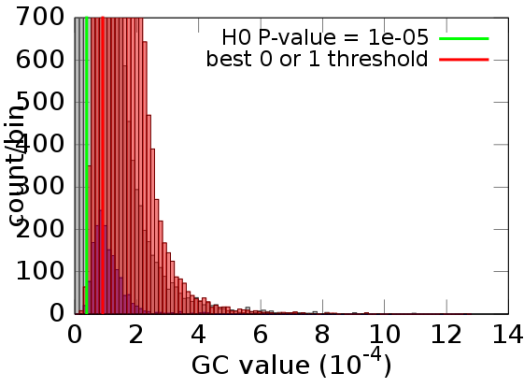


(a) Joint. over guess: 30247, lack guess: 4.

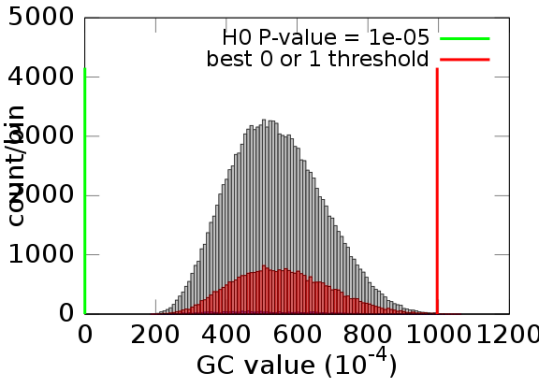


(b) Pairwise. over guess: 135502, lack guess: 0.

Figure 16: n=380E+20I, sparseness: 15%, volt



(a) Joint. over guess: 43422, lack guess: 71.



(b) Pairwise. over guess: 127452, lack guess: 0.

Figure 17: n=380E+20I, sparseness: 20%, volt

2 Pairwise Regression v.s. Joint Regression: Formula Results

2.1 Expression for Three-Variable GC

Assume $\mathrm{E}(x_t) = \mathrm{E}(y_t) = \mathrm{E}(y_t) = 0$, as usual, let's do a 3-var AR of order m:

$$\begin{cases} x_t = \sum_{j=1}^{m} a_j^{(11)} x_{t-j} + \sum_{j=1}^{m} a_j^{(12)} y_{t-j} + \sum_{j=1}^{m} a_j^{(13)} z_{t-j} + \epsilon_t^{(1)} & \text{(a)} \\ y_t = \sum_{j=1}^{m} a_j^{(21)} x_{t-j} + \sum_{j=1}^{m} a_j^{(22)} y_{t-j} + \sum_{j=1}^{m} a_j^{(23)} z_{t-j} + \epsilon_t^{(2)} & \text{(b)} \\ z_t = \sum_{j=1}^{m} a_j^{(31)} x_{t-j} + \sum_{j=1}^{m} a_j^{(32)} y_{t-j} + \sum_{j=1}^{m} a_j^{(33)} z_{t-j} + \epsilon_t^{(3)} & \text{(c)} \end{cases} \tag{1}$$

where variables x_t, y_t, z_t might be column vectors, coefficient $a^{(ij)}$ might be matrix.

In order to solve Eq.(1), left multiply x_{t-k}^T, y_{t-k}^T, z_{t-k}^T to each of them, then take expectation. e.g. left multiply x_{t-k}^T to (a) of Eq.(1), then take expectation:

$$\mathrm{E}(x_t x_{t-k}^T) = \sum_{j=1}^{m} a_j^{(11)} \mathrm{E}(x_{t-j} x_{t-k}^T) + \sum_{j=1}^{m} a_j^{(12)} \mathrm{E}(y_{t-j} x_{t-k}^T) + \sum_{j=1}^{m} a_j^{(13)} \mathrm{E}(z_{t-j} x_{t-k}^T) + \mathrm{E}(\epsilon_t^{(1)} x_{t-k}^T) \tag{2}$$

For convenience, define $x^{(1)} = x$, $x^{(2)} = y$, $x^{(3)} = z$, and

$$R^{(uv)} = (b_{jk}) = \left(\mathrm{E}(x_{t-j}^{(u)} x_{t-k}^{(v)T}) \right), \ (j, k = 1 \dots m).$$

For example, $R^{(xy)} = (b_{jk}) = \left(\mathrm{E}(x_{t-j} y_{t-k}^T) \right)$, its j-row, k-column element is $\mathrm{E}(x_{t-j} y_{t-k}^T)$. b is only a placeholder variable to indicate row and column index. $R^{(uv)T} = R^{(vu)}$. Define

$$\vec{a}^{(uv)} = (b_j) = \left(a_j^{(uv)} \right), \ (j = 1 \dots m).$$

For example, $\vec{a}^{(12)} = (b_j)^T = \left(a_j^{(12)} \right)^T$, $(j = 1 \dots m)$, it's a row vector (also might be partitioned matrix). Define

$$\vec{v}^{(u|v)} = (b_k)^T = \left(\mathrm{E}(x_t^{(u)} x_{t-k}^{(v)T}) \right)^T, \ (k = 1 \dots m).$$

For example, $\vec{v}^{(x|y)} = (b_k)^T = \left(\mathrm{E}(x_t y_{t-k}^T) \right)^T$, $(k = 1 \dots m)$. It's row vector.

By use above symbols, Eq.(1)(a) now satisfy

$$\begin{cases} \vec{v}^{(x|x)} = \vec{a}^{(11)} R^{(xx)} + \vec{a}^{(12)} R^{(yx)} + \vec{a}^{(13)} R^{(zx)} \\ \vec{v}^{(x|y)} = \vec{a}^{(11)} R^{(xy)} + \vec{a}^{(12)} R^{(yy)} + \vec{a}^{(13)} R^{(zy)} \\ \vec{v}^{(x|z)} = \vec{a}^{(11)} R^{(xz)} + \vec{a}^{(12)} R^{(yz)} + \vec{a}^{(13)} R^{(zz)} \end{cases} \tag{3}$$

note $\mathrm{E}(\epsilon_t^{(u)} x_{t-k}^{(v)T}) = 0, \ \forall k = 1 \dots m$. And similarly for (b), (c) of Eq.(1).

We need to solve $\vec{a}^{(uv)}$, $\mathrm{var}(\epsilon_t^{(u)}) \left(= \mathrm{E}(\epsilon_t^{(u)} x_t^{(u)T}) \right)$, then 2-var AR, then finally GC.

Further assume x_t, y_t, z_t are white noise each, hence $R^{(uu)} = v^{(u)} I$, $v^{(u)} \triangleq \mathrm{var}(x_t^{(u)})$ [1], and $\vec{v}^{(x|x)} = 0$.

Rewrite Eq.(3) as:

$$\begin{bmatrix} \vec{a}^{(11)} & \vec{a}^{(12)} & \vec{a}^{(13)} \end{bmatrix} \begin{bmatrix} v^{(x)} I & R^{(xy)} & R^{(xz)} \\ R^{(yx)} & v^{(y)} I & R^{(yz)} \\ R^{(zx)} & R^{(zy)} & v^{(z)} I \end{bmatrix} = \begin{bmatrix} 0 & \vec{v}^{(x|y)} & \vec{v}^{(x|z)} \end{bmatrix}. \tag{4}$$

[1] I denote the identical matrix. For vector $x^{(u)}$, if $x^{(u)}$ is not whitened element-wise, $R^{(uu)}$ is a blocked diagonal matrix.

In order to solve the variance of residual, in Eq.(2), let $k = 0$, get

$$\mathrm{var}(x_t^{(1)}) = \vec{a}^{(12)}\vec{v}^{(x|y)T} + \vec{a}^{(13)}\vec{v}^{(x|z)T} + \mathrm{var}(\epsilon_t^{(1)}). \tag{5}$$

Due to Eq.(4), it's

$$\mathrm{var}(\epsilon_t^{(1)}) = \mathrm{var}(x_t^{(1)}) - \begin{bmatrix} 0 & \vec{v}^{(x|y)} & \vec{v}^{(x|z)} \end{bmatrix} \begin{bmatrix} v^{(x)}I & R^{(xy)} & R^{(xz)} \\ R^{(yx)} & v^{(y)}I & R^{(yz)} \\ R^{(zx)} & R^{(zy)} & v^{(z)}I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vec{v}^{(x|y)T} \\ \vec{v}^{(x|z)T} \end{bmatrix}. \tag{6}$$

Or in correlation form, $S^{(xy)} \triangleq R^{(uv)}/\sqrt{v^{(u)}v^{(v)}}$, $\vec{s}^{(u|v)} \triangleq \vec{v}^{(u|v)}/\sqrt{v^{(u)}v^{(v)}}$ (row vector)[2]:

$$\mathrm{var}(\epsilon_t^{(1)}) = \mathrm{var}(x_t^{(1)}) \left(I - \begin{bmatrix} 0 & \vec{s}^{(x|y)} & \vec{s}^{(x|z)} \end{bmatrix} \begin{bmatrix} I & S^{(xy)} & S^{(xz)} \\ S^{(yx)} & I & S^{(yz)} \\ S^{(zx)} & S^{(zy)} & I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vec{s}^{(x|y)T} \\ \vec{s}^{(x|z)T} \end{bmatrix} \right). \tag{7}$$

Note that in this expression, Z need a full regression.
Through similar process, get 2-var AR solution without variable y:

$$\mathrm{var}(\epsilon_t^{A(1)}) = \mathrm{var}(x_t^{(1)}) \left(I - \begin{bmatrix} 0 & \vec{s}^{(x|z)} \end{bmatrix} \begin{bmatrix} I & S^{(xz)} \\ S^{(zx)} & I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vec{s}^{(x|z)T} \end{bmatrix} \right) \tag{8}$$

For simplicity, assume x_t, y_t are scalars below.
Finally get GC

$$F_{y\to x|z} = \ln\left(\frac{1 - \begin{bmatrix} 0 & \vec{s}^{(x|z)} \end{bmatrix} \begin{bmatrix} I & S^{(xz)} \\ S^{(zx)} & I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vec{s}^{(x|z)T} \end{bmatrix}}{1 - \begin{bmatrix} 0 & \vec{s}^{(x|y)} & \vec{s}^{(x|z)} \end{bmatrix} \begin{bmatrix} I & S^{(xy)} & S^{(xz)} \\ S^{(yx)} & I & S^{(yz)} \\ S^{(zx)} & S^{(zy)} & I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vec{s}^{(x|y)T} \\ \vec{s}^{(x|z)T} \end{bmatrix}} \right). \tag{9}$$

2.2 Approximation of Three-Variable GC

For $\vec{s}^{(u|v)}$ and $S^{(uv)}$ $(u \neq v)$ small, view the inverse matrix in Eq.(9) as identity matrix, get approximation

$$F_{y\to x|z} = \vec{s}^{(x|y)}\vec{s}^{(x|y)T} \approx \sum_{k=1}^{m} \vec{s}_k^{(x|y)}\vec{s}_k^{(x|y)} = \sum_{k=1}^{m} \frac{\mathrm{E}(x_t y_{t-k})^2}{\mathrm{var}(x_t)\mathrm{var}(y_t)} \tag{10}$$

Since z_t in Eq.(1) might be a vector, Eq.(10) is also the n-var approximate GC expression.
Similarly through Eq.(4), get coefficients (row (blocked) vector)

$$\begin{cases} \vec{a}^{(11)} \approx 0 \\ \vec{a}^{(12)} \approx \vec{v}^{(x|y)} \left(v^{(y)}\right)^{-1} = (b_k) = (\mathrm{E}(x_t y_{t-k})/\mathrm{var}(y_t)) \quad k = 1, \ldots, m \\ \vec{a}^{(13)} \approx \vec{v}^{(x|z)} \left(v^{(z)}\right)^{-1} = (b_k) = \left(\mathrm{E}(x_t z_{t-k}^T)/\mathrm{var}(z_t)\right) \end{cases} \tag{11}$$

Note that Eq.(10) is actually the same as the approximation for bivariate GC. So we need to correct it to reflect effect from other variables.

We have

$$1 - \frac{1-d}{1-c} \approx d - c, \quad \text{for } d, c \ll 1,$$

[2]Here the definition of $S^{(xy)}$ and $\vec{s}^{(u|v)}$ are for scalar $v^{(u)}$. See section 4.3 for derivation detail.

$$\ln(\frac{1}{1-b}) \approx b, \quad \text{for } b \ll 1.$$

So Eq.(9) become

$$F_{y\to x|z} = \ln\left(\frac{1-c}{1-d}\right) \approx d - c$$

where

$$c = \begin{bmatrix} 0 & \vec{s}^{(x|z)} \end{bmatrix} \begin{bmatrix} I & S^{(xz)} \\ S^{(zx)} & I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vec{s}^{(x|z)T} \end{bmatrix}$$

$$d = \begin{bmatrix} 0 & \vec{s}^{(x|y)} & \vec{s}^{(x|z)} \end{bmatrix} \begin{bmatrix} I & S^{(xy)} & S^{(xz)} \\ S^{(yx)} & I & S^{(yz)} \\ S^{(zx)} & S^{(zy)} & I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vec{s}^{(x|y)T} \\ \vec{s}^{(x|z)T} \end{bmatrix}$$

Under the small $S^{(uv)}$ assumption,

$$(I-B)^{-1} \approx I + B, \quad \text{for } \|B\| \ll 1, \tag{12}$$

$$B_2 = -\begin{bmatrix} O & S^{(xz)} \\ S^{(zx)} & O \end{bmatrix}, \quad B_3 = -\begin{bmatrix} O & S^{(xy)} & S^{(xz)} \\ S^{(yx)} & O & S^{(yz)} \\ S^{(zx)} & S^{(zy)} & O \end{bmatrix}$$

$$c \approx \begin{bmatrix} 0 & \vec{s}^{(x|z)} \end{bmatrix} (I + B_2) \begin{bmatrix} 0 \\ \vec{s}^{(x|z)T} \end{bmatrix}$$

$$d \approx \begin{bmatrix} 0 & \vec{s}^{(x|y)} & \vec{s}^{(x|z)} \end{bmatrix} (I + B_3) \begin{bmatrix} 0 \\ \vec{s}^{(x|y)T} \\ \vec{s}^{(x|z)T} \end{bmatrix}.$$

Get the corrected GC (compared with Eq.(10))

$$F_{y\to x|z} \approx \vec{s}^{(x|y)} \vec{s}^{(x|y)T} - 2\vec{s}^{(x|y)} S^{(yz)} \vec{s}^{(x|z)T}. \tag{13}$$

In covariance form (Still assume x_t, y_t are scalars) (See section (4.3) for definition of $\Lambda^{(z)}$)

$$F_{y\to x|z} \approx \frac{\vec{v}^{(x|y)} \vec{v}^{(x|y)T} - 2\vec{v}^{(x|y)} R^{(yz)} \left(\Lambda^{(z)}\right)^2 \vec{v}^{(x|z)T}}{\text{var}(x_t)\text{var}(y_t)}.$$

For vector z, Eq.(13) can be writen in component form

$$F_{y\to x|z} \approx \vec{s}^{(x|y)} \vec{s}^{(x|y)T} - 2\vec{s}^{(x|y)} \sum_{k=1}^{n_z} S^{(yz[k])} \vec{s}^{(x|z[k])T}. \tag{14}$$

$$F_{y\to x|z} \approx \frac{1}{\text{var}(x_t)\text{var}(y_t)} \left(\vec{v}^{(x|y)} \vec{v}^{(x|y)T} - 2\vec{v}^{(x|y)} \sum_{k=1}^{n_z} \frac{R^{(yz[k])} \vec{v}^{(x|z[k])T}}{\text{var}(z[k]_t)} \right). \tag{15}$$

But Eq.(14) is not suitable for large n (say $n > 10$), due to $\|B_2\| \ll 1$ and $\|B_3\| \ll 1$ not satisfies any more (hence the approximation Eq.(12) not valid).

For large n, we may use

$$F_{y\to x|z} \approx d - c = \vec{s}^{(x|y)} \vec{s}^{(x|y)T} + \begin{bmatrix} 0 & \vec{s}^{(x|y)} & \vec{s}^{(x|z)} \end{bmatrix} \left(\begin{bmatrix} I & S^{(xy)} & S^{(xz)} \\ S^{(yx)} & I & S^{(yz)} \\ S^{(zx)} & S^{(zy)} & I \end{bmatrix}^{-1} - \begin{bmatrix} I & 0 & S^{(xz)} \\ 0 & I & 0 \\ S^{(zx)} & 0 & I \end{bmatrix}^{-1} \right) \begin{bmatrix} 0 \\ \vec{s}^{(x|y)T} \\ \vec{s}^{(x|z)T} \end{bmatrix}$$

where $F_{y\to x} \approx \vec{s}^{(x|y)} \vec{s}^{(x|y)T}$, so we get the relationship between conditional GC and pairwise GC.

Formula ($\vec{a}^{(12)}$ is from standard whitened data):

$$d - c = d - c = \vec{a}^{(12)} \left(Q^{(yy)}\right)^{-1} \vec{a}^{(12)T}$$

$$\left(Q^{(yy)}\right)^{-1} = I - S^{(yz)} S^{(zy)} - \left(S^{(yx)} - S^{(yz)} S^{(zx)}\right) \left(I - S^{(xz)} S^{(zx)}\right)^{-1} \left(S^{(xy)} - S^{(xz)} S^{(zy)}\right)$$

3 Simulator `raster_tuning_HH` changes

Compared current version (2014-10-27, changeset 49:20ba04b94c86) to JYL version (2014-05-01 just after branch created, changeset 27:9efee6777a85)[3].

3.1 Speed improvement

Eliminate Exponential calculation

Recall that there are 7 exponent calculation for each neuron at each time step

$$\alpha_n(V_i) = \frac{0.1 - 0.01V_i}{\exp(1 - 0.1V_i) - 1} \qquad \beta_n(V_i) = 0.125\exp(-V_i/80)$$
$$\alpha_m(V_i) = \frac{2.5 - 0.1V_i}{\exp(2.5 - 0.1V_i) - 1} \qquad \beta_m(V_i) = 4\exp\left(-V_i/18\right)$$
$$\alpha_h(V_i) = 0.07\exp(-V_i/20) \qquad \beta_h(V_i) = \frac{1}{\exp(3 - 0.1V_i) + 1}$$

$$g(V_j^{\text{pre}}) = 1/\left(1 + \exp(-(V_j^{\text{pre}} - 85\,\text{mV})/2)\right).$$

Reformulate the exponentials as following ($g(V_j^{\text{pre}})$ has no change)

$$e_1 = \exp(-0.1V_i), \quad e_2 = \sqrt{e_1}$$

$$\exp(1 - 0.1V_i) = e_1\exp(1) \qquad \exp(-V_i/80) = \sqrt{\sqrt{e_2}}$$
$$\exp(2.5 - 0.1V_i) = e_1\exp(2.5) \qquad \exp\left(-V_i/18\right) = \exp\left(-V_i/18\right)$$
$$\exp(-V_i/20) = e_2 \qquad \exp(3 - 0.1V_i) = e_1\exp(3)$$

So we only need to compute 3 exponentials, instead of 7 (the constant numbers like exp(1) will be calculated at compile time, instead of runtime).

Also note that there some expressions like $\frac{\exp(x)}{\exp(x)+a}$ in the origin code. They have been changed to $\frac{1}{1+a\exp(-x)}$, which is faster (if the compiler failed to optimize it), simpler and no precision lose.

The exponential elimination make the program 30% faster for 100 neuron case.

Eliminate unnecessary synaptic calculation

Neurons are not always firing. So don't do it at all if not firing. The original code contain this type of optimization, but only halfway to that, like this

```
for neuron_passive = 1 to n
  for neuron_driving = 1 to n
    if (neuron_driving is firing)
      synaptic[neuron_passive] += v[neuron_driving]
                                * matrix[neuron_passive][neuron_driving]
```

In whatever condition, there are $O(n^2)$ calculations there (also bad for CPU pipeline). Instead, write it this way

```
for neuron_driving = 1 to n
  if (neuron_driving is firing)
    for neuron_passive = 1 to n
      synaptic[neuron_passive] += v[neuron_driving]
                                * matrix[neuron_passive][neuron_driving]
```

Which cost much less if it's spare network.

With the above two eliminations, the program runs 1 times faster. (And synaptic calculation no more a hot spot for 100 neuron case)

[3] https://bitbucket.org/bewantbe/ifsimu branch HH_jyl

Utilize SSE/AVX instructions

Thanks to the pipeline and Streaming SIMD (Single instruction, multiple data) Extensions (SSE) and Advanced Vector Extensions (AVX), modern CPU can calculate 2 or 4 (or even 8) multiplications (and potentially 2 or 4 or more additions at the same time) in ever clock cycle.

Use SSE/AVX instructions directly would need hand write Assembly code or equivalences (compiler intrinsics), or fine tune of compiler options, and probably in the end, no big success.

So use SSE instructions through third party package is a good compromise. I choose Eigen[4] here. Eigen performs automatic parallel also.

After analyze the hot spot of the program, and rewrite the hot spot in vector form (use Eigen), get 30%~40% more faster.

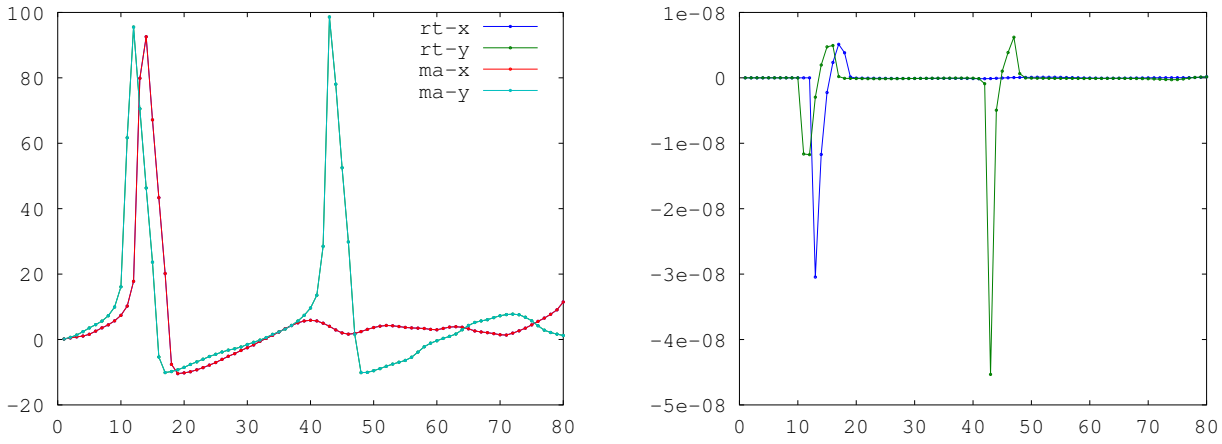
Further possible speed optimization

- Use highly optimized math function library for exp() calculation. (e.g. mkl, acml) The related code is still a hot spot.
- Use sparse representation for the adjacency matrix.

3.2 Program Correctness Verification

See matcode/prj_GC_clean/HH/test_HH.m for parameter details.

See matcode/prj_GC_clean/HH/HH_cal.nb for code to generate accurate results.



(a) voltage trace

(b) voltage trace differences

Figure 18: Accuracy verification. x-axis is time in millisecond, y-axis is voltage in millivolt(-65 mV shifted). "rt" curves is generated by `raster_tuning_HH`, "ma" curve is generated by Mathematica using implicit Runge-Kutta with max time step 1/16 ms.

We can see that the error is below 10^{-9} (relatively) in short time.

4 Appendix

4.1 Few Basic Properties of The HH Model Used

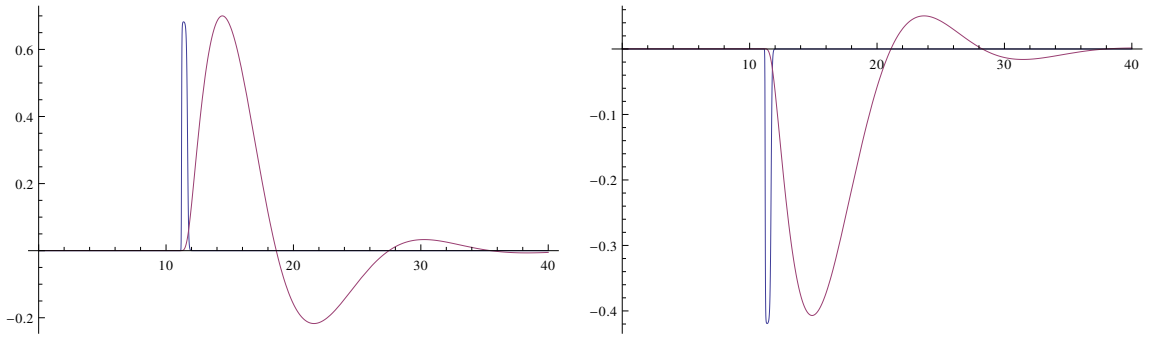
For current input.

Slowest firing, ISI: 19.547 ms, rate: 51.159 Hz. (input current: 6.27)

Fast firing, ISI=8.544 ms, rate 117.03 Hz. (input current: 50)

[4] http://eigen.tuxfamily.org/

Critical current input that firing will automatically disappeared: 6.264.



(a) EPSP ($s^{EE} = 0.05$) (b) IPSP ($s^{II} = 0.1$)

Figure 19: The blue curve is the synaptic input

Note that in this model, EPSP and IPSP will be affect by pre-synaptic neuron dynamics.

4.2 The HH model Used

It is the classical Hodgkin-Huxley (HH) neuron model. For neuron i, its membrane potential V_i obey

$$\begin{cases} C\dfrac{\mathrm{d}V_i}{\mathrm{d}t} = -(V_i - V_{\mathrm{Na}})G_{\mathrm{Na}}h_i m_i^3 - (V_i - V_{\mathrm{K}})G_{\mathrm{K}}n_i^4 - (V_i - V_{\mathrm{L}})G_{\mathrm{L}} + I_i^{\mathrm{input}} \\ \dfrac{\mathrm{d}m_i}{\mathrm{d}t} = (1-m_i)\alpha_m(V_i) - m_i\beta_m(V_i) \\ \dfrac{\mathrm{d}h_i}{\mathrm{d}t} = (1-h_i)\alpha_h(V_i) - h_i\beta_h(V_i) \\ \dfrac{\mathrm{d}n_i}{\mathrm{d}t} = (1-n_i)\alpha_n(V_i) - n_i\beta_n(V_i) \end{cases}$$

where

$$\begin{aligned} \alpha_n(V_i) &= \frac{0.1 - 0.01V_i}{\exp(1 - 0.1V_i) - 1} & \beta_n(V_i) &= 0.125\exp(-V_i/80) \\ \alpha_m(V_i) &= \frac{2.5 - 0.1V_i}{\exp(2.5 - 0.1V_i) - 1} & \beta_m(V_i) &= 4\exp\left(-V_i/18\right) \\ \alpha_h(V_i) &= 0.07\exp(-V_i/20) & \beta_h(V_i) &= \frac{1}{\exp(3 - 0.1V_i) + 1} \end{aligned}$$

V_i, m_i, n_i, h_i, I_i^{input} are functions of t, and others are constants: $V_{\mathrm{Na}} = 115\,\mathrm{mV}$, $V_{\mathrm{K}} = -12\,\mathrm{mV}$, $V_{\mathrm{L}} = 10.6\,\mathrm{mV}$ (resting potential set to $0\,\mathrm{mV}$), $G_{\mathrm{Na}} = 120\,\mathrm{mS}\cdot\mathrm{cm}^{-2}$, $G_{\mathrm{K}} = 36\,\mathrm{mS}\cdot\mathrm{cm}^{-2}$, $G_{\mathrm{L}} = 0.3\,\mathrm{mS}\cdot\mathrm{cm}^{-2}$ and membrane capacity $C = 1\,\mu\mathrm{F}\cdot\mathrm{cm}^{-2}$.

The interaction between neurons and external inputs come from I_i^{input}

$$I_i^{\mathrm{input}} = I_i^{\mathrm{E}} + I_i^{\mathrm{I}}, \quad I_i^{\mathrm{E}} = -(V_i - V_G^{\mathrm{E}})G_i^{\mathrm{E}}, \quad I_i^{\mathrm{I}} = -(V_i - V_G^{\mathrm{I}})G_i^{\mathrm{I}}$$

I_i^{E}, I_i^{I} are excitatory and inhibitory input respectively, and V_G^{E}, V_G^{I} is their reversal potential. The conductances G_i^Q ($Q \in \{\mathrm{E}, \mathrm{I}\}$) evolves according to

$$\frac{\mathrm{d}G_i^Q}{\mathrm{d}t} = -\frac{G_i^Q}{\sigma_d^Q} + H_i^Q, \quad \frac{\mathrm{d}H_i^Q}{\mathrm{d}t} = -\frac{H_i^Q}{\sigma_r^Q} + \sum_k F_i^Q\delta(t - T_{i,k}^F) + \sum_{j\neq i} S_{ij}g(V_j^{\mathrm{pre}})$$

$$g(V_j^{\mathrm{pre}}) = 1/\left(1 + \exp(-(V_j^{\mathrm{pre}} - 85\,\mathrm{mV})/2)\right)$$

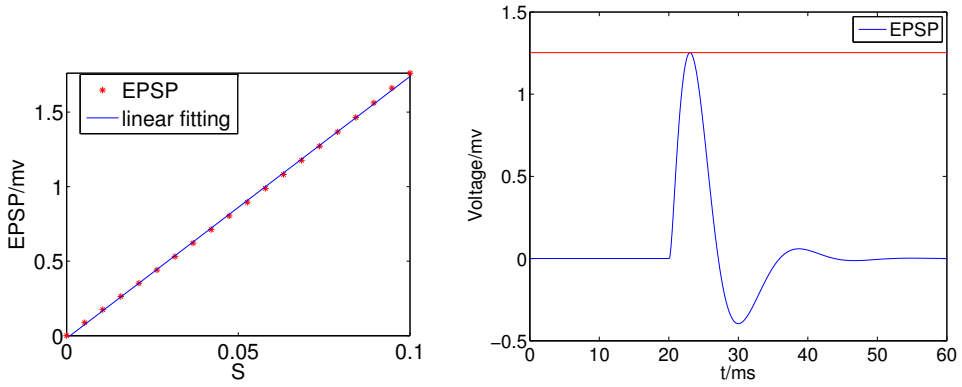
where F_i^Q is the strength of external input to neuron i, $T_{i,k}^F$ is its time of k-th input event, which is a Poisson process with rate μ_i. We call this term the Poisson input. S_{ij} is the coupling strength from j-th neuron to i-th neuron. V_j^{pre} is the (presynaptic) membrane potential of j-th neuron. σ_r^Q, σ_d^Q are the fast rising and slow decaying timescales in the α function. $V_G^{\text{E}} = 65\,\text{mV}$, $V_G^{\text{I}} = -15\,\text{mV}$, $\sigma_d^E = 0.5$, $\sigma_r^E = 3.0$, $\sigma_d^I = 0.5$, $\sigma_r^I = 7.0$.

We use adjacency matrix $A = (A_{ij})$ to denote the neural network structure, i.e. $S_{ij} = A_{ij} S^{Q_i Q_j}$, and $S^{Q_i Q_j}$ is one of S^{EE}, S^{EI}, S^{IE}, S^{II}, depends on the type of corresponding neuron pair (E for excitatory, I for inhibitory). $A_{ij} \neq 0$ means there is a direct affection to i-th neuron from j-th neuron. When we talk about "homogeneous coupling", we mean A_{ij} equals either 1 or 0.

F, μ, A, $S^{Q_i Q_j}$, σ_r^Q, σ_d^Q are parameters relate to synaptic and input to neurons. For all neurons $F_i^{\text{E}} = F$, $F_i^{\text{I}} = 0$, $\mu_i = \mu$. During one simulation, these parameters are all constant.

The time delay due to long dendrite or axion are neglected.

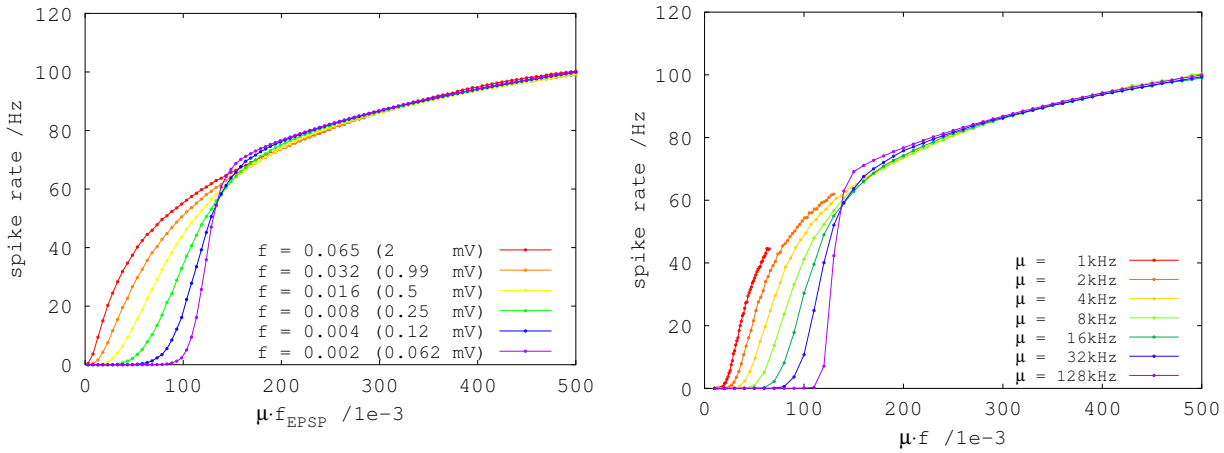
In numerical simulation, we use explicit fourth-order Runge-Kutta method with time step 1/32 ms. The data samples (i.e. x_t, y_t) we used are voltages obtained in sampling rate 2 kHz. When we talk about spike train data, we mean $x_t = 1$ if $V_i(t)$ just pass through the threshold (10 mV in our case) from low to high, otherwise $x_t = 0$.



(a) (b)

Figure 20: (a) Excitatory postsynaptic potential (EPSP) - S^E relation. $V_{\text{EPSP}} \approx 17\,S^{\text{EE}}$ mV (b) typical EPSP voltage trace in this model.

For inhibitory input and Poisson input, the curves are roughly the same (up to scalings) as Fig.(20). The relation between peak respond and model parameter is $V_{\text{IPSP}} \approx -5.0\,S^{\text{II}}$ mV, $V_{\text{EPSP-Poisson}} \approx 31\,F^{\text{E}}$ mV.



(a) Scan input rate μ.

(b) Scan input strength f. Some curves terminate because EPSP exceed 2 mV.

Figure 21: Gain function under Poisson pulse drive.

4.3 Derivation Details of Eq.(7)

Define

$$S^{(uv)} = (b_{jk}) = \left(D_{t-j}^{(u)} \mathrm{E}(x_{t-j}^{(u)} x_{t-k}^{(v)T}) D_{t-k}^{(v)} \right), \tag{16}$$

$$\vec{s}^{(u|v)} = (b_k)^T = \left(D_t^{(u)} \mathrm{E}(x_t^{(u)} x_{t-k}^{(v)T}) D_{t-k}^{(v)} \right), \tag{17}$$

$$D_t^{(u)} = \begin{bmatrix} 1/\sqrt{\mathrm{var}(x_t^{(u[1])})} & & & \\ & 1/\sqrt{\mathrm{var}(x_t^{(u[2])})} & & \\ & & \ddots & \\ & & & 1/\sqrt{\mathrm{var}(x_t^{(u[n_u])})} \end{bmatrix}, \tag{18}$$

where $x_t^{(u[j])}$ means the j-th component of $x_t^{(u)}$, n_u is the dimension of $x_t^{(u)}$. This definition is consist with the $R^{(uv)}$ after normalize all components of $x_t^{(u)}$.

Define

$$\Lambda^{(u)} = \begin{bmatrix} D_t^{(u)} & & & \\ & D_{t-1}^{(u)} & & \\ & & \ddots & \\ & & & D_{t-m}^{(u)} \end{bmatrix}, \tag{19}$$

so

$$S^{(uv)} = \Lambda^{(u)} R^{(uv)} \Lambda^{(v)}, \tag{20}$$

$$\vec{s}^{(u|v)} = D_t^{(u)} \vec{v}^{(u|v)} \Lambda^{(v)}. \tag{21}$$

Accordingly

$$\begin{bmatrix} v^{(x)}I & R^{(xy)} & R^{(xz)} \\ R^{(yx)} & v^{(y)}I & R^{(yz)} \\ R^{(zx)} & R^{(zy)} & v^{(z)}I \end{bmatrix}^{-1} = \begin{bmatrix} \Lambda^{(x)} & & \\ & \Lambda^{(y)} & \\ & & \Lambda^{(z)} \end{bmatrix} \begin{bmatrix} S^{(xx)} & S^{(xy)} & S^{(xz)} \\ S^{(yx)} & S^{(yy)} & S^{(yz)} \\ S^{(zx)} & S^{(zy)} & S^{(zz)} \end{bmatrix}^{-1} \begin{bmatrix} \Lambda^{(x)} & & \\ & \Lambda^{(y)} & \\ & & \Lambda^{(z)} \end{bmatrix}$$

$$\begin{aligned} & \mathrm{var}(x_t^{(1)}) - \mathrm{var}(\epsilon_t^{(1)}) \\ = & \begin{bmatrix} 0 & \vec{v}^{(x|y)} & \vec{v}^{(x|z)} \end{bmatrix} \begin{bmatrix} v^{(x)}I & R^{(xy)} & R^{(xz)} \\ R^{(yx)} & v^{(y)}I & R^{(yz)} \\ R^{(zx)} & R^{(zy)} & v^{(z)}I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vec{v}^{(x|y)T} \\ \vec{v}^{(x|z)T} \end{bmatrix} \\ = & \left(D_t^{(x)} \right)^{-1} \left(\begin{bmatrix} 0 & \vec{s}^{(x|y)} & \vec{s}^{(x|z)} \end{bmatrix} \begin{bmatrix} S^{(xx)} & S^{(xy)} & S^{(xz)} \\ S^{(yx)} & S^{(yy)} & S^{(yz)} \\ S^{(zx)} & S^{(zy)} & S^{(zz)} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vec{s}^{(x|y)T} \\ \vec{s}^{(x|z)T} \end{bmatrix} \right) \left(D_t^{(x)} \right)^{-1}. \end{aligned} \tag{22}$$

If x_t, y_t are scalar, then $S^{(xx)} = S^{(yy)} = I_{m \times m}$. And the middle (parenthesized, note as ♦ below) part of Eq.(22) is also scalar, hence

$$\mathrm{var}(x_t^{(1)}) - \mathrm{var}(\epsilon_t^{(1)}) = \left(D_t^{(x)} \right)^{-2} \blacklozenge = \mathrm{var}(x_t^{(1)}) \blacklozenge \tag{23}$$

Further, If $\mathrm{E}(x_t^{(z[j])} x_t^{(z[k])}) = 0,\ \forall j \neq k$, then $S^{(zz)} = I_{(p-2)m \times (p-2)m}$, which lead to Eq.(7).

4.4 Time Cost in Computation

Only for reference. They are computed on different computers.

Table 8: GC analysis time cost. There are ranges because there are different networks.

n	len/ms	Fr Rate(Hz)	HH simu	od_{max}	GC (sec)
50	10^6	32.0	0.838 h	40	74.2
50	10^6	12.2	0.509 h	40	81.6
100	10^6	33.7	2.200 h	40	117.0
100	10^6	12.5	1.203 h	40	120.3
200	10^6	36.0	11.97 h	40	1391~1608
200	10^6	13.9	5.51 h	40	1562
400	10^6	35.4~37.1	25.36 h~30.52 h	40	7256~8063

.
.
.
.
.