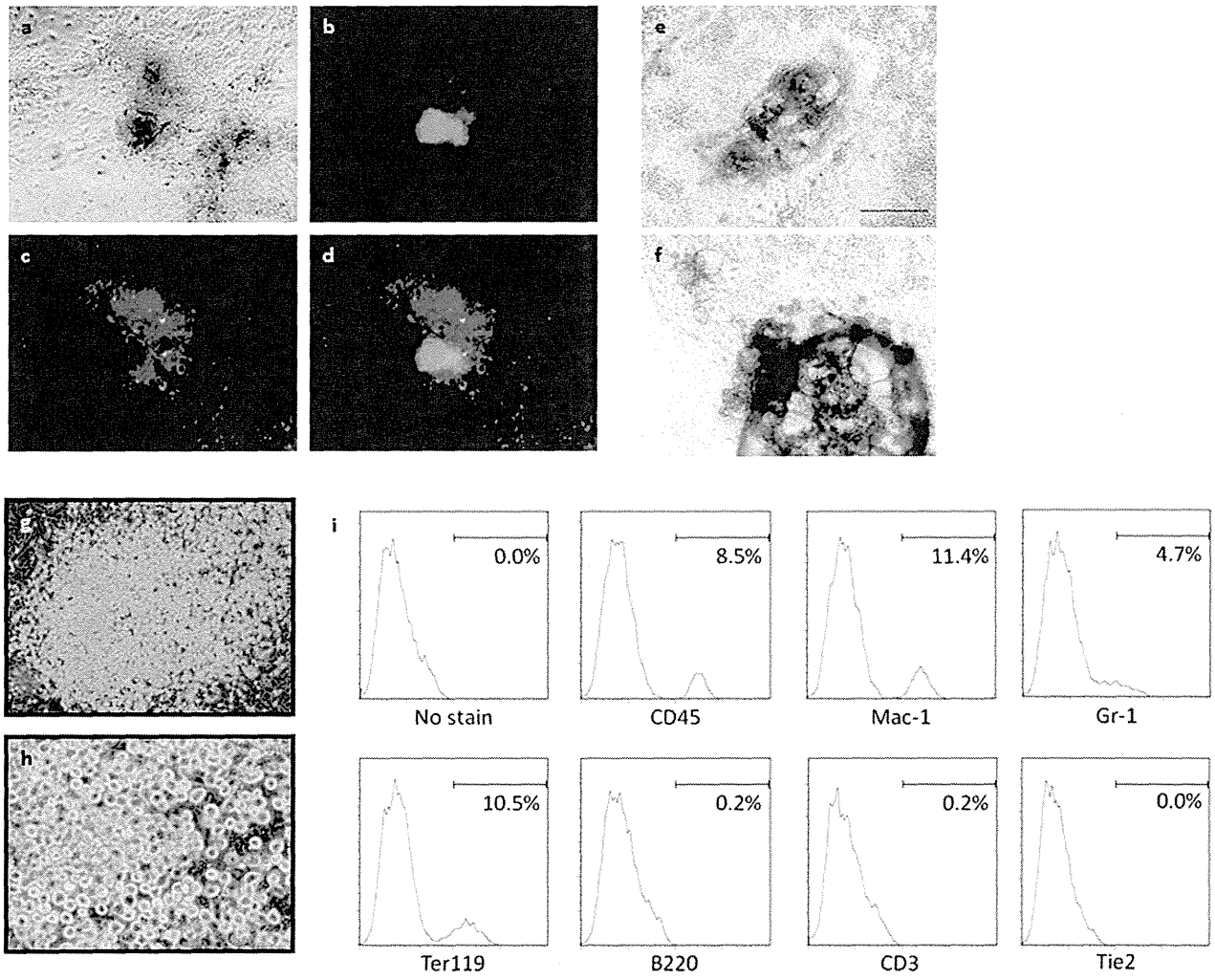

FIGURE 1. **Reprogramming factors also induce non-pluripotent cells.** *a–d*, *Nanog-GFP*$^{+}$, *DsRed*$^{-}$ iPS cell colony (*green*), and *Nanog-GFP*$^{-}$, *DsRed*$^{+}$ non-pluripotent pseudo cells (*red*); and phase-contrast (*a*), *Nanog* promoter-driven *GFP* expression (*b*), retroviral *DsRed* expression (*c*), and merged image (*d*). *e* and *f*, tail-tip fibroblasts-derived cardiomyocyte-like cells following four RTF infection. These cells can be seen pulsing in supplemental Movies S1 and S2. *g* and *h*, morphology of MEF-derived rounded blood-like cells following four TF infection. *h*, is a high magnification of *g*. *i*, flow cytometric analysis of blood-like cells. Expression levels were analyzed using the antibodies indicated.

specific embryonic antigen-1 (SSEA-1); finally, the retroviruses used for RTF introduction are silenced, whereas endogenous gene expression of pluripotency-associated molecules, such as *Oct3/4* and *Nanog*, are activated. At this time, reactivation of an X chromosome is also seen.

On the other hand, the more detailed mechanisms underlying the induction of pluripotency are largely unknown. There are some clues, such as the involvement of cell-cell contact during the generation of iPS cells, observed during time-lapse analysis, and it is also suggested that a certain probabilistic action has been influenced during iPS cell generation (6, 7). In addition, although it is clear that the demethylation of DNA and changes in histone modifications occur in the regulatory regions of pluripotency-associated genes, such as *Oct3/4* and *Nanog*, it is not known when these events take place (8). Furthermore, it was reported recently that the four RTFs mediated the induction of other cell types, in addition to iPS cells, including epiblast stem cells and cardiomyocytes (9, 10). Therefore, understanding the mechanism initiated in response to the introduction of the four RTFs is important, not only for the efficient induction of iPS cells but also for controlling other cell fates.

In this study, we focused on the ratio of the four RTFs. To analyze the different ratios for each factor, tagged vectors were generated and used to sort the transfected RTFs on the basis of their expression levels by FACS analysis. Using this sorting method, the efficiency of iPS cell generation was compared with the expression level of each of the four RTFs, and the optimal ratio of the four factors was identified as follows: *Oct3/4*-high, *Sox2*-low, *Klf4*-high, and c-*Myc*-high. Under these conditions, iPS cell generation efficiency was 88 times greater than the worst effective ratio (*Oct3/4*-low, *Sox2*-high, *Klf4*-low, and c-*Myc*-low). Finally, the molecular signature for sorting the high efficiency reprogramming conditions from low efficiency conditions was identified by comparing the gene expression profiles of mouse embryonic fibroblasts (MEFs) at 2 days after the RTFs infection.

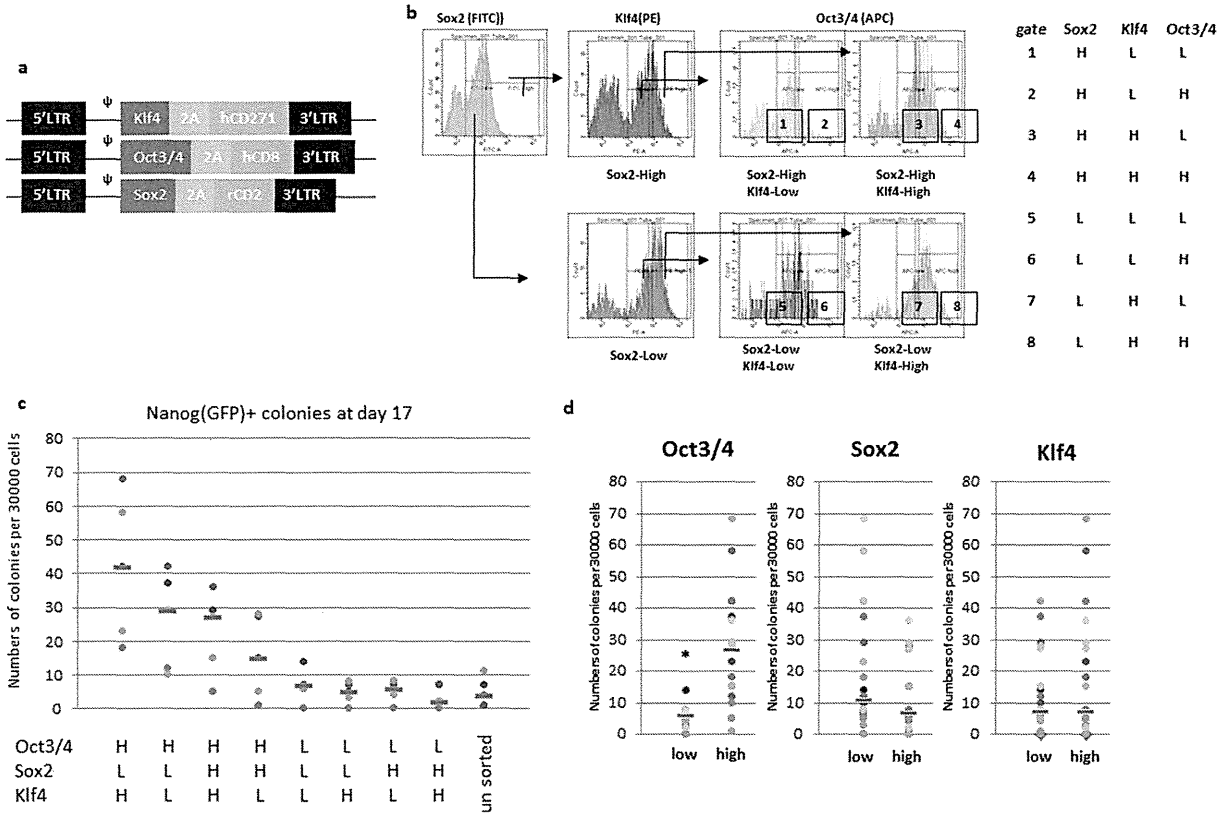

FIGURE 2. **Somatic cell reprogramming using different ratios of *Oct3/4*, *Sox2*, and *Klf4*.** *a*, retrovirus vectors with cell surface antigens. *b*, flow cytometric analysis of the introduced factors together with the sorting gates used. *c* and *d*, number of *Nanog-GFP*$^+$ colonies after sorting on day 17 of culture. MEFs were sorted using relative gene expression levels, as indicated on the *horizontal axis*. *Dots* represent the numbers of each experiment and *bar* means median. The *numbers on the graph* (*c*) were recalculated based on the expression level of each factor in *d*. *Dots* represent the numbers of each experiment and the *bar* means median. *, p = 1.14E-06. *H*, high; *L*, low.

EXPERIMENTAL PROCEDURES

Mice—The *Nanog-GFP-IRES-puro* transgenic mouse strain (RBRC02290) has been described previously (8, 11). C57BL/6 mice were purchased from Japan SLC (Shizuoka, Japan). Animal care was performed in accordance with the guidelines established by Keio University for animal and recombinant DNA experimentation. *Nanog-GFP* MEFs were generated by crossing the transgenic mice with C57BL/6 mice.

Plasmids—Retroviral plasmids for iPS cell induction have been described previously (11). The following 2A sequence was used: 5′-aaaattgtcgctcctgtcaaacaaactcttaactttgatttactcaaactggctggggatgtagaaagcaatccaggtcca-3′ (12). The surface tagging antigens were obtained from *pMXs-IRES-rat CD2*, *pMX-IRES-human CD8*, and *pMACS-human LNGFR* (Miltenyi Biotech). Human CD25 was cloned by PCR with the following primers: 5′-GCCACCATGGATTCATACCTGCTGATG-3′ and 5′-GTCGACCTAGATTGTTCTTCTACTCTT-3′. The constructs, pMXs-IRES-rat CD2 and pMX-IRES-human CD8, were donated by Dr. Masato Kubo and Dr. Takashi Saito, respectively (13, 14). For the epigenetic modifiers, *Setdb2*, *Smyd3*, and *Whsc1l1* variants *1* and *2* were cloned by PCR, inserted into the *pGEM-T-easy* plasmid (Promega) and converted to pMXs via the BamHI and XhoI sites. The PCR primers used were as follows: *Setdb2*, forward, 5′-GGATCCGCCACCATGGAAGAAAAAAATGGTGATGCA-3′; *Setdb2*, reverse, 5′-CTCGAGTTATATTAATTTTTTCCGACACTT-3′; *Smyd3*, forward, 5′-GGATCCGCCACCATGGAGGCACTGAAGGTGGAAAAG-3′; *Smyd3*, reverse, 5′-CTCGAGTTAGGAGGCTCGTATGTTGGCATC-3′; *Whsc1l1 variant 1*, forward, 5′-GGATCCGCCACCATGGATTTCTCTTTCTCTTTCATG-3′; *Whsc1l1 variant 1*, reverse, 5′-CTCGAGTCAGTCCACAGTTTCCTCTTTCGC-3′; and *Whsc1l1 variant 2*, forward, 5′-GGATCCGCCACCATGGATTTCTCTTTCTCTTTCATG-3′; *Whsc1l1 variant 2*, reverse, 5′-GTCGACTCACTCCTTTACTTCTTCTCCACT-3′.

Reprogramming of MEFs Using Tagged Vectors—*Oct3/4–2A-hCD8*, *Sox2–2A-rCD2*, and *Klf4–2A-hCD271* with, or without, *c-Myc-2A-hCD25* were introduced into MEFs by retroviruses according to the previously described method for iPS cell induction (15). Two days after infection, MEFs were collected by incubation in 0.05% trypsin EDTA for 5 min. After washing, the cells were incubated with an anti-FcγR antibody (2.4G2) (eBioscience) at 4 °C for 30 min, and then incubated with a fluorescein isothiocyanate-conjugated anti-rat CD2 monoclonal antibody (OX-34; BioLegend), a phycoerythrin-conjugated anti-human CD271 monoclonal antibody (C40–1457; BD Biosciences), and an allophycocyanin (APC)-conjugated anti-human CD8 monoclonal antibody (RPA-T8;

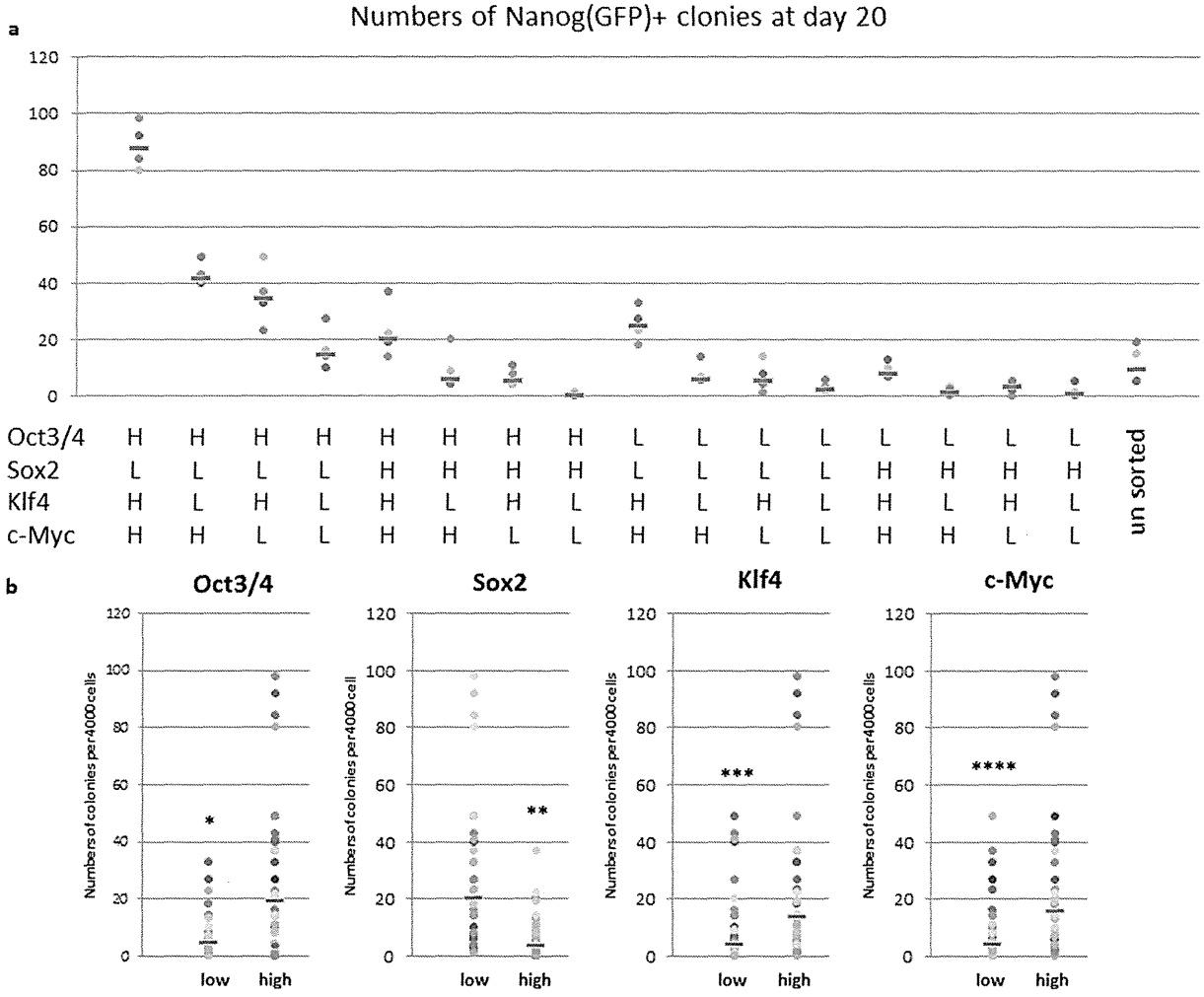

FIGURE 3. **Somatic cell reprogramming using different ratios of *Oct3/4*, *Sox2*, *Klf4*, and *c-Myc*.** *a* and *b*, number of *Nanog-GFP*$^+$ colonies after sorting on day 21 of culture. MEFs were sorted using relative gene expression levels, as indicated on the *horizontal axis*. *Dots* represent the numbers of each experiment and the *bar* means median. The *numbers on graph* (*a*) were recalculated based on the expression level of each factor in *b*. *Dots* represent the numbers of each experiment and the *bar* means median. *, p = 2.69E-04; **, p = 8.96E-06; ***, p = 3.20E-03; ****, p = 8.96.98E-04. *H*, high; *L*, low.

BioLegend) for 30 min at 4 °C. For the four factor reprogramming, a phycoerythrin-Cy7-conjugated anti-human CD25 monoclonal antibody (M-A251) was also added. After washing, samples were sorted using a FACSVantage SE cytometer (BD Biosciences). Sorted cells were cultured on STO cells at a density of 30,000 cells (without c-*Myc*) or 4,000 cells (with c-*Myc*) per well in six-well plates. The numbers of *Nanog-GFP* positive (*Nanog-GFP*$^+$) colonies were counted on days 17 or 21. Data are presented as the each dot. The median numbers are also presented as a *bar*. Statistical significance for difference of the medians was determined by exact Wilcoxon test using the R exactRankTests package.

Analysis of Chemokines for Reprogramming—MEFs carrying the four introduced reprogramming factors were reseeded on STO feeders 4 days after infection at a density of 4,500 cells/well in six-well plates. At that time, 100 ng/ml of each chemokine was added every 2 days to the culture until day 17. The medium was changed every second day. On day 7 after infection puromycin was added to the culture. Colony numbers were counted at day 23.

Statistical Analysis of Reprogramming Efficiency According to the Ratio of Reprogramming Factors—The Mann-Whitney *U* test was performed to compare differences in distribution for the number of positive colonies under the different reprogramming conditions.

Microarray Data Analysis—Expression profiles of MEFs at 2 days after the RTF infection were analyzed using the whole mouse genome 44K3D-Gene Mouse Oligo chip 24K (Agilent Technologies, Santa Clara, CA). Fluorescence intensities were detected using the Scan-Array Life Scanner (PerkinElmer Life Science) and photomultiplier tube levels were adjusted to achieve 0.1–0.5% pixel saturation. Each TIFF image was analyzed with GenePix Pro software version 6.0 (Molecular Devices, Sunnyvale, CA). The data were filtered to remove low-confidence measurements and normalized globally per array such that the median signal intensity was set at 50.

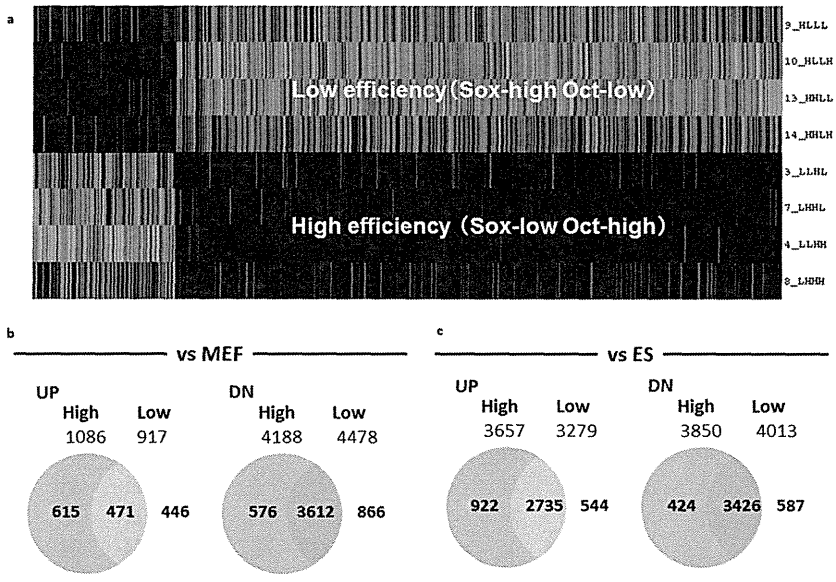

FIGURE 4. **Microarray analysis of the high and low efficiency conditions for reprogramming.** *a*, array heat map of signature genes from low and high efficiency conditions. *b*, number of signature genes that were up- or down-regulated in the high and low efficiency conditions compared with parental MEFs. *c*, number of signature genes that were up- or down-regulated compared with ES cells. The names of the genes in MEFs and ES cells are listed in supplemental Tables S1 and S2, respectively.

All 43,379 probes were collapsed into 21,609 genes with Entrez gene identifier (ID) by taking the maximum intensity among probe sets corresponding to the same gene ID. The standard Student's *t* test was performed for each comparison and the false discovery rate was estimated using the Benjamini-Hochberg procedure to obtain differentially expressed genes as a signature. In this study, a false discovery rate <5% was used as a threshold. To characterize the molecular backgrounds of the signature genes, enrichment analysis for canonical pathways and Gene Ontology biological processes (c2-cp and c5-bp gene sets in MSigDB version 3.0 (16)) was performed using the GO Term Finder (17).

RESULTS

The Four RTFs Do Not Always Induce Pluripotency in Somatic Cells—Somatic cell reprogramming is brought about by the four RTFs, *Oct3/4*, *Sox2*, *Klf4*, and c-*Myc*. Initially, these transcription factors were introduced into somatic cells by retroviral vectors; however, because these viral vectors are usually, but not always completely, inactivated toward the end of the reprogramming process, silencing of the retrovirus promoter was recognized as one of the reprogramming criteria (8). For the current study, the four RTFs were introduced into MEFs carrying green fluorescent protein (GFP) under the control of the *Nanog* promoter. To monitor silencing, a *DsRed* vector was also introduced. After induction of the four RTFs, *Nanog-GFP*$^+$ and *DsRed* negative (*DsRed*$^-$) iPS candidate cells were observed (Fig. 1*a*), as well as *Nanog-GFP*$^-$ and *DsRed*$^+$ pseudo-pluripotent iPS cells (Fig. 1*b*). These data indicated that the RTFs did not achieve pluripotency in all somatic cells.

Moreover, occasionally non-iPS cells with specific features were also seen after induction of the four RTFs; for example, Fig. 1 shows spontaneously beating cardiomyocyte-like cells generated from adult tail-tip fibroblasts (Fig. 1, *e* and *f*, and supplemental Movies S1 and S2). In addition, morphologically rounded, blood-like cells were also observed (Fig. 1, *g* and *h*). When these blood-like cells were collected by pipetting and stained for cell surface markers, they were found to be positive for the pan-hematopoietic marker, CD45 (Fig. 1*i*). Analysis of lineage markers revealed that these blood-like cells contained macrophages (Mac-1), granulocytes (Gr-1), and erythroid cells (Ter119) (Fig. 1*i*). However, B (B220) and T (CD3) lymphoid cells were not detected (Fig. 1*i*). The so-called "transdifferentiation" of these two lineages by the factors used in somatic cell reprogramming has also been reported by other groups (10, 18). These data indicated that the four RTFs do not only induce the pluripotent state but are also capable of producing terminally differentiated cells.

Optimal Ratio of the Four RTFs for Somatic Cell Reprogramming—Because the reprogramming factors can also induce other cell types as well as pluripotent cells, it should be possible to fine-tune the RTFs to produce only fully pluripotent cells. Therefore, we speculated that there would be an optimal ratio of the four RTFs for efficient pluripotent cell generation. To investigate the importance of the relative expression levels of each of the RTFs in somatic cell reprogramming, *Sox2*, *Klf4*, and *Oct3/4* were tagged with different rat and human cell surface antigens using a *2A* sequence (Fig. 2*a*). After infection of MEFs with each of these constructs, flow cytometry with specific antibodies was used to sort the cells according to the expression levels of the exogenous genes (Fig. 2*b*). Using this strategy, the MEFs were grouped based on the ratios of the three factors, and *Nanog-GFP*$^+$ colonies were counted on day 17 after infection. The effects of the expression of each of the three factors are shown in Fig. 2*c*, and the results indicated that the greatest numbers of *Nanog-GFP*$^+$ colonies were obtained with high levels of *Oct3/4*. The most effective ratio of the three

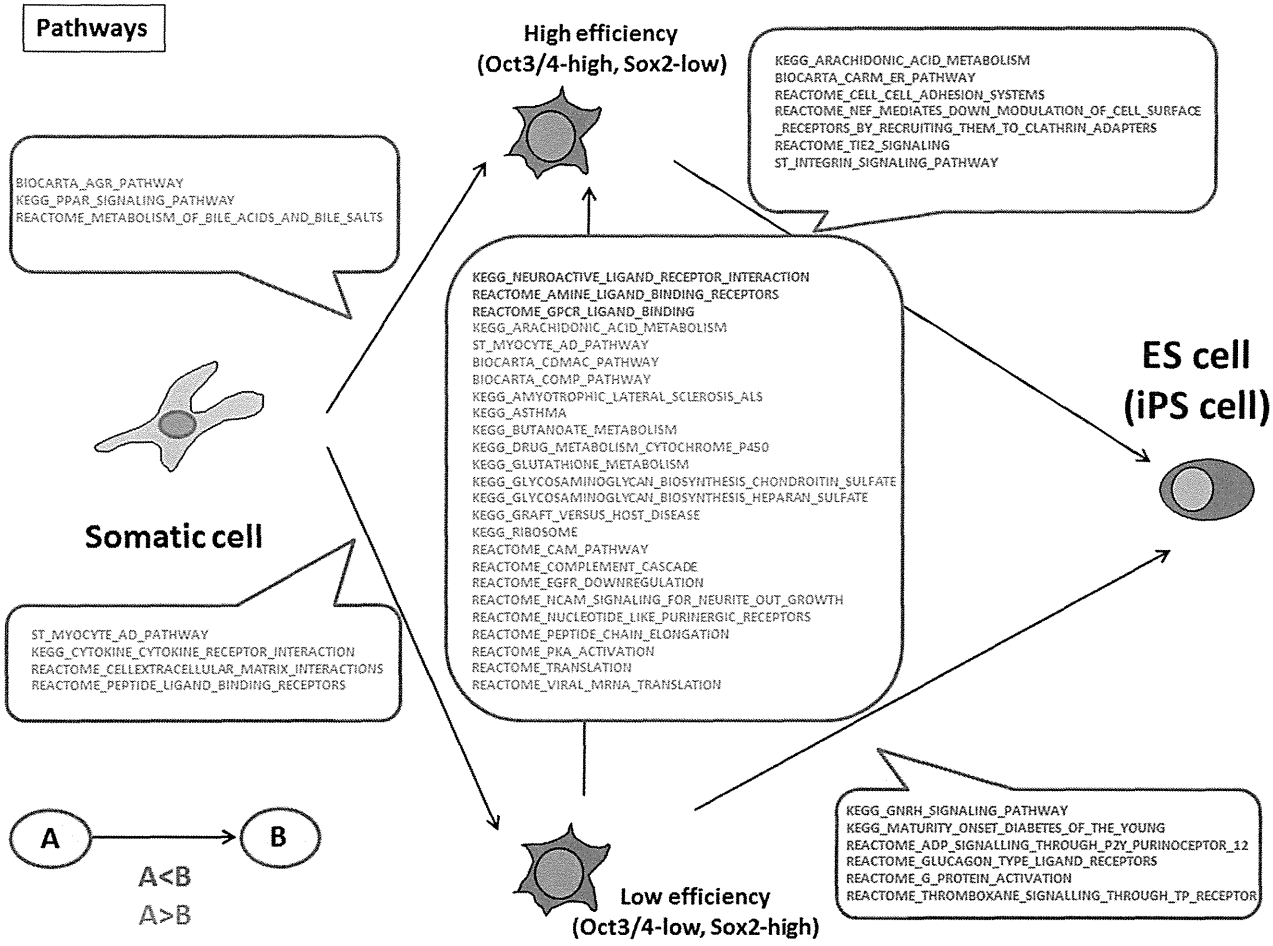

FIGURE 5. **Pathway analysis of microarray data.** Microarray data of MEFs at 2 days after the RTF infection under high (*Oct3/4*-high, *Sox2*-low) and low (*Oct3/4*-low, *Sox2*-high) efficiency conditions were compared with MEFs and ES cells, and the up- and down-regulated pathways between each cell type are shown. Up-regulated pathways are shown in *red* and down-regulated pathways are shown in *blue*.

factors (*Oct3/4*-high, *Sox2*-low, and *Klf4*-high) was seven times more efficient than for the worst effective ratio (*Oct3/4*-low, *Sox2*-high, and *Klf4*-low).

In addition to these three RTFs, the effect of c-*Myc* was also analyzed. A human *CD25*-tagged c-*Myc* vector was generated and used to monitor the relative expression of all four RTFs (supplemental Fig. S1). The expression levels of each of the factors were confirmed by RT-PCR (supplemental Figs. S2 and S3). The results are shown in Fig. 3. The addition of c-*Myc* did not affect the ratios of the other three factors. High expression of *Oct3/4*, *Klf4*, and c-*Myc* favored the induction of pluripotency, whereas low expression of *Sox2* was better for reprogramming. Similar to induction with three RTFs, the most effective ratio of the four factors (*Oct3/4*-high, *Sox2*-low, *Klf4*-high, and c-*Myc*-high) was 50 times more efficient than for the worst effective ratio (*Oct3/4*-low, *Sox2*-high, *Klf4*-low, and c-*Myc*-high). Regardless of the efficiency, generated iPS cells showed similar gene expression patterns to ES cells and have a potential to differentiate to all three germ layers (supplemental Figs. S4 and S5).

Microarray Analysis of High (Oct3/4-high and Sox2-low) and Low (Oct3/4-low and Sox2-high) Efficiency Reprogramming Conditions—We searched for the most effective combination of the four RTFs using the relationship between *Nanog-GFP*$^+$ colony numbers and the reprogramming factor ratio. Among the four factors, the *Oct3/4* and *Sox2* expression ratios correlated significantly with positive colony numbers. In cells with high levels of *Oct3/4* and low levels of *Sox2*, ~16.2 times greater numbers of positive colonies were found when all four factors were introduced (supplemental Fig. S6*a*). A similar result was also found when only three factors were used (supplemental Fig. S6*b*) even if the statistically dominant factor was only *Oct3/4*. To determine the molecular basis underlying these ratios and indeed, somatic cell reprogramming, microarray analysis was performed using the high (*Oct3/4*-high and *Sox2*-low) and the low (*Oct3/4*-low and *Sox2*-high) reprogramming conditions.

The signature genes were identified using bioinformatics calculations (Fig. 4*a*). First, the signature genes in MEFs at 2 days after the RTF infections were compared with those of the parental MEFs and with pluripotent embryonic stem (ES) cells. When compared with MEFs, ~1,000 genes were up-regulated and 4,000 genes were down-regulated under both high and low efficiency conditions. Whereas about half the up-regulated

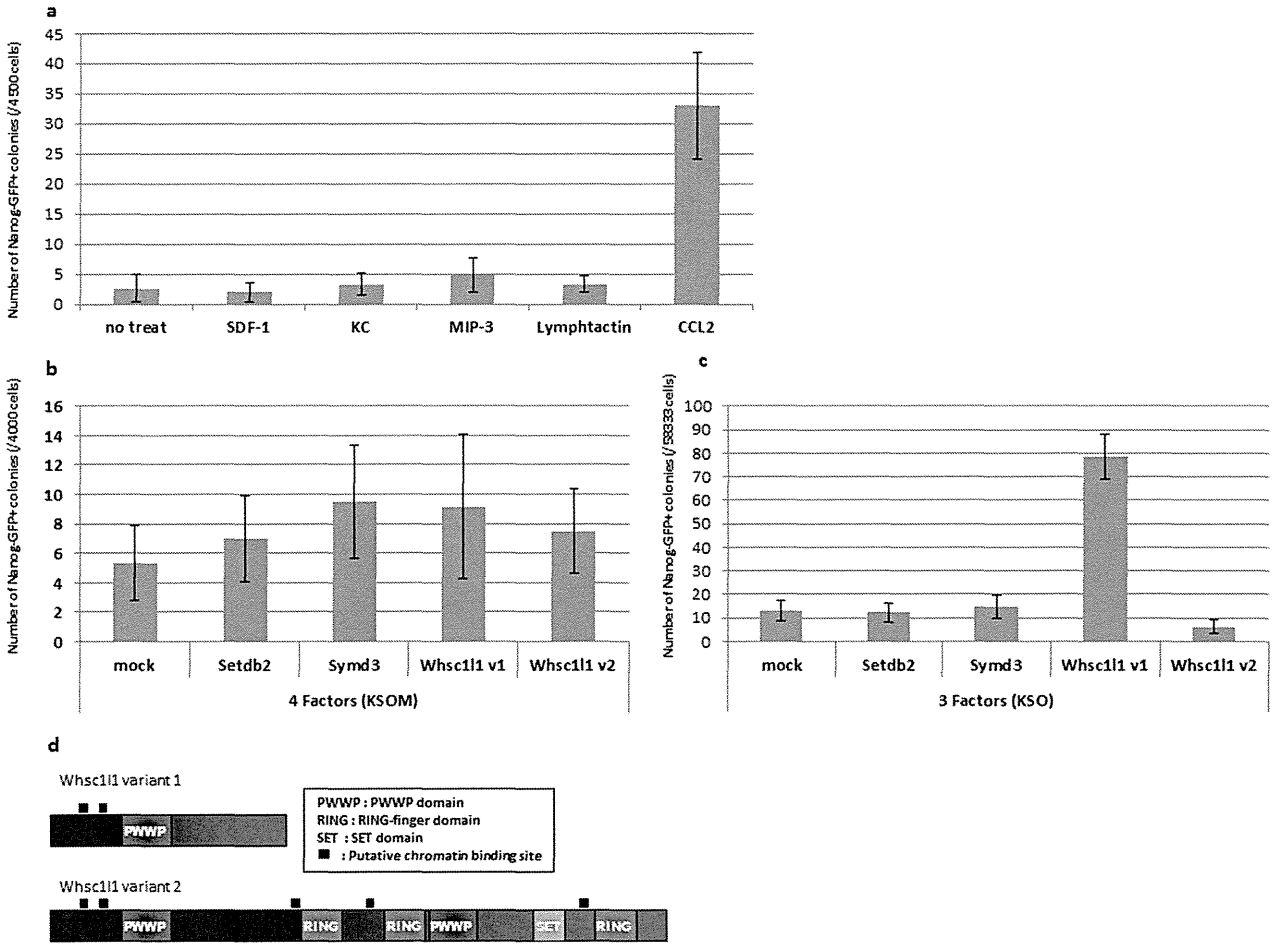

FIGURE 6. **The effect of chemokines and epigenetic modifiers on somatic cell reprogramming.** *a*, MEFs were infected with the four RTFs and the chemokines indicated were added from days 4 to 17 of the culture. The numbers of *Nanog-GFP*$^+$ colonies on day 23 of culture are indicated. *b* and *c*, MEFs were infected with the epigenetic factors indicated, together with four (*b*) or three (*c*) of the RTFs. The numbers of *Nanog-GFP*$^+$ colonies at 17 days after infection are shown.

genes were common to both the high and low reprogramming conditions, more than 80% of the down-regulated genes were common to both (Fig. 4*b* and supplemental Table S1). On the other hand, when compared with ES cells, more than 70% of the up-regulated genes and 80% of the down-regulated genes were common to both cell types under both sets of conditions (Fig. 4*c* and supplemental Table S2). These data indicated that the expression of many signature genes in MEFs at 2 days after the RTFs infection was altered under both high (*Oct3/4*-high and *Sox2*-low) and low (*Oct3/4*-low and *Sox2*-high) reprogramming conditions when compared with MEF and ES cells.

Molecular Signature for Sorting the High (Oct3/4-high and Sox2-low) Low (Oct3/4-low and Sox2-high) Efficiency Reprogramming Conditions—To determine the difference between the high (*Oct3/4*-high and *Sox2*-low) and low (*Oct3/4*-low and *Sox2*-high) efficiency conditions, the microarray data for these two conditions were compared. GO analysis showed that under the high efficiency condition, positive regulation of MAP kinase activity was down-regulated (supplemental Fig. S7) in iPS cells, which is significant because it is known that inhibition of the MAP kinase pathway is important for pluripotency (19). Furthermore, pathway analysis of the microarray data revealed that certain pathways were up-regulated preferentially under the high condition, compared with the low condition (Fig. 5). Under the high efficiency condition, we focused on enrichment of the GPCR pathways, and in particular, the chemokine members of the GPCR superfamily. To analyze the involvement of chemokines during somatic cell reprogramming, the effect of several chemokines on the generation of iPS cells was examined. Of these, the addition of CCL2 achieved a 12.3 times greater reprogramming efficiency than in untreated cells (Fig. 6*a*). These results suggested that the microarray data contained clues for the optimization of pluripotency induction.

To understand the mechanism further, transcription factors and epigenetic modifiers were analyzed as these factors direct cell fate and alter the regulation of multiple genes. Although under the low efficiency condition only nine TFs were up-regulated, 60 TFs were up-regulated under the high efficiency condition (supplemental Fig. S8*a* and Table 1). Furthermore, when the epigenetic modifiers were investigated, only one gene was up-regulated under the low efficiency condition and four under the high condition (supplemental Fig. S8*b* and Table 2). These

TABLE 1
Transcription factors up-regulated under high and low efficiency conditions

Symbol	Description
High efficiency condition	
POU5F1	POU class 5 homeobox 1
HOXC4	Homeobox C4
IRX4	Iroquois homeobox 4
NEUROG1	Neurogenin 1
BARHL1	BarH-like homeobox 1
FOXN1	Forkhead box N1
KLF17	Kruppel-like factor 17
NR5A1	Nuclear receptor subfamily 5, group A, member 1
ZNF43	Zinc finger protein 43
POU4F1	POU class 4 homeobox 1
RFX4	Regulatory factor X, 4 (influences HLA class II expression)
ESRRG	Estrogen-related receptor gamma
FOXH1	Forkhead box H1
SOX15	SRY (sex determining region Y)-box 15
LHX1	LIM homeobox 1
TOPORS	Topoisomerase I binding, arginine/serine-rich
HNF4A	Hepatocyte nuclear factor 4, α
NKX61	NK6 homeobox 1
PROP1	PROP paired-like homeobox 1
CAMTA1	Calmodulin binding transcription activator 1
ARID5B	AT-rich interactive domain 5B (MRF1-like)
SOX17	SRY (sex determining region Y)-box 17
FOXQ1	forkhead box Q1
MAF	v-*maf* musculoaponeurotic fibrosarcoma oncogene homolog (avian)
TCF2	HNF1 homeobox B
FEV	FEV (ETS oncogene family)
HES2	Hairy and enhancer of split 2 (*Drosophila*)
PITX3	Paired-like homeodomain 3
HOXA3	Homeobox A3
HNF4G	Hepatocyte nuclear factor 4, γ
TCF7L2	Transcription factor 7-like2 (T-cell specific, HMG-box)
TP73	Tumor protein p73
NR3C2	Nuclear receptor subfamily 3, group C, member 2
HSF1	Heat shock transcription factor 1
GLI1	GLI family zinc finger 1
SOX1	SRY (sex determining region Y)-box 1
ZNF124	Zinc finger protein 124
CDK2	Cyclin-dependent kinase 2
FOXE3	Forkhead box E3
RBPJ	Recombination signal-binding protein for immunoglobulin κJ region
CREBBP	CREB-binding protein
HOXB9	Homeobox B9
FOXL2	Forkhead box L2
FOXF2	Forkhead box F2
NCX	T-cell leukemia homeobox 2
TFDP2	Transcription factor Dp-2 (E2F dimerization partner 2)
ATBF1	Zinc finger homeobox 3
NR1I3	Nuclear receptor subfamily 1, group I, member 3
SOX12	SRY (sex determining region Y)-box 12
LMO3	LIM domain only3 (rhombotin-like 2)
ABL1	c-*abl* oncogene 1, receptor tyrosine kinase
GTF2IRD1	GTF2I repeat domain containing 1
IRF1	Interferon regulatory factor 1
NFIA	Nuclear factor I/A
SS18L1	Synovial sarcoma translocation gene on chromosome 18-like 1
NFATC2	Nuclear factor of activated T-cells, cytoplasmic, calcineurin-dependent 2
STAT5B	Signal transducer and activator of transcription 5B
FOXO4	Forkhead box O4
HOXB6	Homeobox B6
RUNX2	Runt-related transcription factor 2
Low efficiency condition	
ID3	Inhibitor of DNA binding 3, dominant negative helix-loop-helix protein
XPA	*Xeroderma pigmentosum*, complementation group A
LEF1	Lymphoid enhancer-binding factor 1
KLF2	Kruppel-like factor 2 (lung)
HEY1	Hairy/enhancer of split related with YRPW motif 1
PRDM1	PR domain containing 1, with ZNF domain
ELOF1	Elongation factor 1 homolog (*Saccharomyces cerevisiae*)
SREBF1	Sterol regulatory element binding transcription factor 1
TBX2	T-box 2

TABLE 2
Epigenetic modifiers upregulated under high and low efficiency conditions

Symbol	Description
High efficiency condition	
SETDB2	SET domain, bifurcated 2
WHSC1L1	Wolf-Hirschhorn syndrome candidate 1-like 1
CREBBP	CREB-binding protein
SMYD3	SET and MYND domain containing 3
Low efficiency condition	
PRDM1	PR domain containing 1, with ZNF domain

data indicated that more transcription factors and epigenetic modifiers appear to be up-regulated under the high condition.

To assess the function of these epigenetic modifiers for somatic cell reprogramming, retrovirus vectors were prepared for *Setdb2, Smyd3*, and *Whsc1l1 variants 1* and *2*, epigenetic modifiers that were up-regulated under the high condition. These factors were introduced into MEFs together with three or four of the RTFs, and *Nanog-GFP*$^+$ colonies were counted on day 17 after infection (Fig. 6, *b* and *c*). When introduced with the three RTFs, *Whsc1l1 variant 1* produced many more colonies than the control; however, *variant 2* had no significant effect (Fig. 6*c*).

DISCUSSION

From investigations into the mechanisms governing somatic cell reprogramming that underlies iPS cell technology, several groups have reported that specific combinations of individual transcription factors can induce the generation of particular cell types (20, 21). In contrast to the induction of pluripotent stem cells, the technology for the generation of lineage-restricted cells is known as transdifferentiation or direct reprogramming. A particular combination of specific transcription factors, which are critical for the development and/or maintenance of the lineage-restricted cells, is used for transdifferentiation. On the other hand, it is reported that pluripotency inducible factors also mediate transdifferentiation (9, 10, 18). Therefore, it is both interesting and feasible to analyze the fine-tuning of the RTFs required for pluripotency. In the present study, the relative ratio of the four RTFs was examined and the results demonstrated that there is, indeed, an optimal ratio (*Oct3/4*-high, *Sox2*-low, *Klf4*-high, and c-*Myc*-high) of these factors for iPS cell generation and, moreover, that the ratio, *Oct3/4*-high and *Sox2*-low, is critical.

It was reported previously that high expression of *Oct3/4* improves reprogramming efficiencies and that modified *Oct3/4* with greater transcriptional activity further enhances the reprogramming efficiency (22, 23). Furthermore, control of *Oct3/4* expression is essential for maintaining ES cells in the undifferentiated state, and both the overexpression and down-regulation of *Oct3/4* can induce ES cell differentiation, suggesting that tightly controlled regulation of *Oct3/4* expression levels controls the maintenance of pluripotency (24). In the current study, we have shown that, in the presence of other factors, high *Oct3/4* expression is critical for somatic cell reprogramming, whereas low levels of *Oct3/4* result in a lower induction efficiency (Figs. 2 and 3).

In contrast to *Oct3/4*, low *Sox2* expression is more efficient for the acquisition of pluripotency, and it is reported that low

Sox2 expression increased the reprogramming efficiency by repressing ectoderm and mesoderm marker genes (25). In the array data presented here, the ectoderm maker, *CryM*, showed a statistically significant decrease in expression under low *Sox2* conditions (supplemental Fig. S9*a* and Table S3). Although another ectoderm marker, *Sox13*, also decreased in the presence of low *Sox2*, the expression of *Sox21* was not linked to the level of *Sox2* (supplemental Table S3). On the other hand, expression of the mesoderm marker, *Myh2*, did not change. However, when *Klf4* expression was altered (high or low), *Myh2* expression was lower in cells under low *Sox2* conditions than under high *Sox2* conditions (supplemental Fig. S9*b* and Table S3). These data indicated that although low *Sox2* expression may repress ectoderm and mesoderm markers, the other RTFs are also involved in the repression of ectoderm and mesoderm marker genes. Furthermore, it has been proposed that a two-step reprogramming mechanism is necessary for the induction of pluripotency, and that *Sox2* functions in the latter stages of reprogramming (26). Our data and a previous report suggest that *Sox2* expression levels are low during the early phase of reprogramming (25). Thus, it is important to analyze the effects of *Sox2* during the different phases of somatic cell reprogramming.

To understand the molecular basis for these events, we performed microarray analyses of the high (*Oct3/4*-high and *Sox2*-low) and low (*Oct3/4*-low and *Sox2*-high) reprogramming conditions. We observed that 50% of the up-regulated and 80% of the down-regulated genes were common to both conditions when iPS cells were compared with MEF and ES cells (Fig. 4, *b* and *c*). Because all four RTFs were introduced for this analysis, it is conceivable that many genes were commonly up- and down-regulated compared with MEF and ES cells. However, when we focused on gene expression levels between the two conditions, the GO terms showed down-regulation of cellular recognition under the low efficiency condition (supplemental Fig. S7), whereas GPCR signaling emerged as a significant pathway under the high condition (Fig. 5). As reported previously, for transdifferentiation using the four RTFs, culture conditions are important for defining cell fate (9, 10), and it is interesting that, in the current study, the high efficiency condition up-regulated the signaling pathway from cell surface molecules, whereas the low efficiency condition down-regulated cellular recognition as demonstrated by the GO terms. One could predict that the four RTFs alter the original program in the somatic cells and up-regulate cell surface molecules to produce favorable signals, including those involved in cell adhesion, required to direct different cell fates. It has been reported that cells adhered together during iPS cell generation, through the up-regulation of the cell adhesion molecule, E-cadherin (7, 27, 28). Furthermore, in the present study, we have confirmed the importance of the GPCR pathway by the addition of the chemokine, CCL2, which binds to the GPCR, CCR2. Interestingly, addition of CCL2 was effective for the high (*Oct3/4*-high and *Sox2*-low) but not low (*Oct3/4*-low and *Sox2*-high) reprogramming conditions (supplemental Fig. S10). CCL2 was recently reported to maintain pluripotency in ES cells by inducing *Klf4* via the activation of STAT3 (29). In the current study, we demonstrated that CCL2 also has a function in the induction of pluripotency. In the case of iPS cell induction, *Klf4* is introduced exogenously; therefore, it is important to know whether other pathways are activated during the induction of pluripotency.

When we focused on the role of transcription factors and epigenetic modifiers of the signature genes, the results showed that the high efficiency condition had more activated genes than the low condition. Thus, because epigenetic modifiers affect the expression of multiple genes, it is important to analyze the listed factors. SETDB2 and SMYD3 contain a SET domain, which has putative methyltransferase activity (30, 31), whereas WHSC1L1 is linked to Wolf-Hirschhorn syndrome (32). None of these genes have been well analyzed with respect to their roles in the induction of pluripotency. However, we found that *Whsc1l1 variant 1*, but not *variant 2*, enhances the reprogramming efficiency in the presence of *Oct3/4*, *Sox2*, and *Klf4* (Fig. 6*c*). WHSC1L1 variant 1 is shorter and about the half the length of variant 2, and interestingly, variant 1 lacks the SET domain, which has putative histone methyltransferase activity (Fig. 6*d*). In future, to improve our understanding of somatic cell reprogramming, it will be important to analyze the reprogramming activity and the supporting roles played by the other genes identified as pluripotency signature genes in this study.

Acknowledgments—We thank N. Tago for cell sorting and A. Kumakubo for technical assistance.

REFERENCES

1. Takahashi, K., and Yamanaka, S. (2006) Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. *Cell* **126,** 663–676
2. Takahashi, K., Tanabe, K., Ohnuki, M., Narita, M., Ichisaka, T., Tomoda, K., and Yamanaka, S. (2007) Induction of pluripotent stem cells from adult human fibroblasts by defined factors. *Cell* **131,** 861–872
3. Hanna, J., Wernig, M., Markoulaki, S., Sun, C. W., Meissner, A., Cassady, J. P., Beard, C., Brambrink, T., Wu, L. C., Townes, T. M., and Jaenisch, R. (2007) Treatment of sickle cell anemia mouse model with iPS cells generated from autologous skin. *Science* **318,** 1920–1923
4. Wernig, M., Zhao, J. P., Pruszak, J., Hedlund, E., Fu, D., Soldner, F., Broccoli, V., Constantine-Paton, M., Isacson, O., and Jaenisch, R. (2008) Neurons derived from reprogrammed fibroblasts functionally integrate into the fetal brain and improve symptoms of rats with Parkinson disease. *Proc. Natl. Acad. Sci. U.S.A.* **105,** 5856–5861
5. Miura, K., Okada, Y., Aoi, T., Okada, A., Takahashi, K., Okita, K., Nakagawa, M., Koyanagi, M., Tanabe, K., Ohnuki, M., Ogawa, D., Ikeda, E., Okano, H., and Yamanaka, S. (2009) Variation in the safety of induced pluripotent stem cell lines. *Nat. Biotechnol.* **27,** 743–745
6. Stadtfeld, M., Maherali, N., Breault, D. T., and Hochedlinger, K. (2008) Defining molecular cornerstones during fibroblast to iPS cell reprogramming in mouse. *Cell Stem Cell* **2,** 230–240
7. Araki, R., Jincho, Y., Hoki, Y., Nakamura, M., Tamura, C., Ando, S., Kasama, Y., and Abe, M. (2010) Conversion of ancestral fibroblasts to induced pluripotent stem cells. *Stem Cells* **28,** 213–220
8. Okita, K., Ichisaka, T., and Yamanaka, S. (2007) Generation of germline-competent induced pluripotent stem cells. *Nature* **448,** 313–317
9. Han, D. W., Greber, B., Wu, G., Tapia, N., Araúzo-Bravo, M. J., Ko, K., Bernemann, C., Stehling, M., and Schöler, H. R. (2011) Direct reprogramming of fibroblasts into epiblast stem cells. *Nat. Cell Biol.* **13,** 66–71
10. Efe, J. A., Hilcove, S., Kim, J., Zhou, H., Ouyang, K., Wang, G., Chen, J., and Ding, S. (2011) Conversion of mouse fibroblasts into cardiomyocytes using a direct reprogramming strategy. *Nat. Cell Biol.* **13,** 215–222
11. Nagamatsu, G., Kosaka, T., Kawasumi, M., Kinoshita, T., Takubo, K.,

Akiyama, H., Sudo, T., Kobayashi, T., Oya, M., and Suda, T. (2011) A germ cell-specific gene, *Prmt5*, works in somatic cell reprogramming. *J. Biol. Chem.* **286,** 10641–10648
12. Hasegawa, K., Cowan, A. B., Nakatsuji, N., and Suemori, H. (2007) Efficient multicistronic expression of a transgene in human embryonic stem cells. *Stem Cells* **25,** 1707–1712
13. Komine, O., Hayashi, K., Natsume, W., Watanabe, T., Seki, Y., Seki, N., Yagi, R., Sukzuki, W., Tamauchi, H., Hozumi, K., Habu, S., Kubo, M., and Satake, M. (2003) The Runx1 transcription factor inhibits the differentiation of naive $CD4^+$ T cells into the Th2 lineage by repressing GATA3 expression. *J. Exp. Med.* **198,** 51–61
14. Yamasaki, S., Ishikawa, E., Sakuma, M., Ogata, K., Sakata-Sogawa, K., Hiroshima, M., Wiest, D. L., Tokunaga, M., and Saito, T. (2006) Mechanistic basis of pre-T cell receptor-mediated autonomous signaling critical for thymocyte development. *Nat. Immunol.* **7,** 67–75
15. Takahashi, K., Okita, K., Nakagawa, M., and Yamanaka, S. (2007) Induction of pluripotent stem cells from fibroblast cultures. *Nat. Protoc.* **2,** 3081–3089
16. Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005) Gene set enrichment analysis. A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U.S.A.* **102,** 15545–15550
17. Boyle, E. I., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J. M., and Sherlock, G. (2004) GO::TermFinder. Open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics* **20,** 3710–3715
18. Szabo, E., Rampalli, S., Risueño, R. M., Schnerch, A., Mitchell, R., Fiebig-Comyn, A., Levadoux-Martin, M., and Bhatia, M. (2010) Direct conversion of human fibroblasts to multilineage blood progenitors. *Nature* **468,** 521–526
19. Ying, Q. L., Wray, J., Nichols, J., Batlle-Morera, L., Doble, B., Woodgett, J., Cohen, P., and Smith, A. (2008) The ground state of embryonic stem cell self-renewal. *Nature* **453,** 519–523
20. Zhou, Q., Brown, J., Kanarek, A., Rajagopal, J., and Melton, D. A. (2008) *In vivo* reprogramming of adult pancreatic exocrine cells to beta cells. *Nature* **455,** 627–632
21. Sekiya, S., and Suzuki, A. (2011) Direct conversion of mouse fibroblasts to hepatocyte-like cells by defined factors. *Nature* **475,** 390–393
22. Wang, Y., Chen, J., Hu, J. L., Wei, X. X., Qin, D., Gao, J., Zhang, L., Jiang, J., Li, J. S., Liu, J., Lai, K. Y., Kuang, X., Zhang, J., Pei, D., and Xu, G. L. (2011) Reprogramming of mouse and human somatic cells by high-performance engineered factors. *EMBO Rep.* **12,** 373–378
23. Hirai, H., Tani, T., Katoku-Kikyo, N., Kellner, S., Karian, P., Firpo, M., and Kikyo, N. (2011) Radical acceleration of nuclear reprogramming by chromatin remodeling with the transactivation domain of MyoD. *Stem Cells* **29,** 1349–1361
24. Niwa, H., Miyazaki, J., and Smith, A. G. (2000) Quantitative expression of Oct-3/4 defines differentiation, dedifferentiation, or self-renewal of ES cells. *Nat. Genet.* **24,** 372–376
25. Yamaguchi, S., Hirano, K., Nagata, S., and Tada, T. (2011) Sox2 expression effects on direct reprogramming efficiency as determined by alternative somatic cell fate. *Stem Cell Res.* **6,** 177–186
26. Lin, Z., Perez, P., Lei, D., Xu, J., Gao, X., and Bao, J. (2011) Two-phase analysis of molecular pathways underlying induced pluripotent stem cell induction. *Stem Cells* **29,** 1963–1974
27. Chen, T., Yuan, D., Wei, B., Jiang, J., Kang, J., Ling, K., Gu, Y., Li, J., Xiao, L., and Pei, G. (2010) E-cadherin-mediated cell-cell contact is critical for induced pluripotent stem cell generation. *Stem Cells* **28,** 1315–1325
28. Redmer, T., Diecke, S., Grigoryan, T., Quiroga-Negreira, A., Birchmeier, W., and Besser, D. (2011) E-cadherin is crucial for embryonic stem cell pluripotency and can replace OCT4 during somatic cell reprogramming. *EMBO Rep.* **12,** 720–726
29. Hasegawa, Y., Takahashi, N., Forrest, A. R., Shin, J. W., Kinoshita, Y., Suzuki, H., and Hayashizaki, Y. (2011) CC chemokine ligand 2 and leukemia inhibitory factor cooperatively promote pluripotency in mouse induced pluripotent cells. *Stem Cells* **29,** 1196–1205
30. Xu, P. F., Zhu, K. Y., Jin, Y., Chen, Y., Sun, X. J., Deng, M., Chen, S. J., Chen, Z., and Liu, T. X. (2010) Setdb2 restricts dorsal organizer territory and regulates left-right asymmetry through suppressing fgf8 activity. *Proc. Natl. Acad. Sci. U.S.A.* **107,** 2521–2526
31. Hamamoto, R., Furukawa, Y., Morita, M., Iimura, Y., Silva, F. P., Li, M., Yagyu, R., and Nakamura, Y. (2004) SMYD3 encodes a histone methyltransferase involved in the proliferation of cancer cells. *Nat. Cell Biol.* **6,** 731–740
32. Stec, I., van Ommen, G. J., and den Dunnen, J. T. (2001) WHSC1L1, on human chromosome 8p11.2, closely resembles WHSC1 and maps to a duplicated region shared with 4p16.3. *Genomics* **76,** 5–8

The Scientific World Journal
Volume 2012, Article ID 957620, 8 pages
doi:10.1100/2012/957620

The ScientificWorldJOURNAL

Research Article

Network Completion Using Dynamic Programming and Least-Squares Fitting

Natsu Nakajima,[1] Takeyuki Tamura,[1] Yoshihiro Yamanishi,[2] Katsuhisa Horimoto,[3] and Tatsuya Akutsu[1]

[1] *Bioinformatics Center, Institute for Chemical Research, Kyoto University Gokasho, Uji, Kyoto 611-0011, Japan*
[2] *Division of System Cohort, Multi-scale Research Center for Medical Science, Medical Institute of Bioregulation, Kyushu University, 3-1-1 Maidashi, Higashi-ku, Fukuoka, Fukuoka 812-8582, Japan*
[3] *Computational Biology Research Center, National Institute of Advanced Industrial Science and Technology, 2-4-7 Aomi, Koto-ku, Tokyo 135-0064, Japan*

Correspondence should be addressed to Tatsuya Akutsu, takutsu@kuicr.kyoto-u.ac.jp

Received 30 August 2012; Accepted 26 September 2012

Academic Editors: W. Tian and X.-M. Zhao

We consider the problem of network completion, which is to make the minimum amount of modifications to a given network so that the resulting network is most consistent with the observed data. We employ here a certain type of differential equations as gene regulation rules in a genetic network, gene expression time series data as observed data, and deletions and additions of edges as basic modification operations. In addition, we assume that the numbers of deleted and added edges are specified. For this problem, we present a novel method using dynamic programming and least-squares fitting and show that it outputs a network with the minimum sum squared error in polynomial time if the maximum indegree of the network is bounded by a constant. We also perform computational experiments using both artificially generated and real gene expression time series data.

1. Introduction

Analysis of biological networks is one of the central research topics in computational systems biology. In particular, extensive studies have been done on inference of genetic networks using gene expression time series data, and a number of computational methods have been proposed, which include methods based on Boolean networks [1, 2], Bayesian networks [3, 4], time-delayed Bayesian networks [5], graphical Gaussian models [6–8], differential equations [9, 10], mutual information [11, 12], and linear classification [13]. However, there is not yet an established or standard method for inference of genetic networks, and thus it still remains a challenging problem.

One of the possible reasons for the difficulty of inference is that the amount of available high-quality gene expression time series data is still not enough, and thus it is intrinsically difficult to infer the correct or nearly correct network from such a small amount of data. Therefore, it is reasonable to try to develop another approach. For that purpose, we proposed an approach called network completion [14] by following Occam's razor, which is a well-known principle in scientific discovery. Network completion is, given an initial network and an observed dataset, to modify the network by the minimum amount of modifications so that the resulting network is (most) consistent with the observed data. Since we were interested in inference of signaling networks in our previous study [14], we assumed that activity levels or quantities of one or a few kinds of proteins can only be observed. Furthermore, since measurement errors were considered to be large and we were interested in theoretical analysis of computational complexity and sample complexity, we adopted the Boolean network [15] as a model of signaling networks. We proved that network completion is computationally intractable (NP-hard) even for tree-structured networks. In order to cope with this computational difficulty, we developed an integer linear programming-based method for completion of signaling pathways [16]. However, this method could not handle addition of edges because of its high computational cost.

In this paper, we propose a novel method, DPLSQ, for completing genetic networks using gene expression time series data. Different from our previous studies [14, 16], we employ a model based on differential equations and assume that expression values of all nodes can be observed. DPLSQ is a combination of least-squares fitting and dynamic programming, where least-squares fitting is used for estimating parameters in differential equations and dynamic programming is used for minimizing the sum of least-squares errors by integrating partial fitting results on individual genes under the constraint that the numbers of added and deleted edges must be equal to the specified ones. One of the important features of DPLSQ is that it can output an optimal solution (i.e., minimum squared sum) in polynomial time if the maximum indegree (i.e., the maximum number of input genes to a gene) is bounded by a constant. Although DPLSQ does not automatically find the minimum modification, it can be found by examining varying numbers of added/deleted edges, where the total number of such combinations is polynomially bounded. If a null network (i.e., a network having no edges) is given as an initial network, DPLSQ can work as an inference method for genetic networks.

In order to examine the effectiveness of DPLSQ, we perform computational experiments using artificially generated data. We also make computational comparison of DPLSQ as an inference method with other existing tools using artificial data. Furthermore, we perform computational experiments on DPLSQ using real cell cycle expression data of *Saccharomyces cerevisiae*.

2. Method

The purpose of network completion is to modify a given network with the minimum number of modifications so that the resulting network is most consistent with the observed data. In this paper, we consider additions and deletions of edges as modification operations (see Figure 1). If we begin with a network with an empty set of edges, it corresponds to network inference. Therefore, network completion includes network inference although it may not necessarily work better than the existing methods if applied to network inference.

In the following, $G(V, E)$ denotes a given network where V and E are the sets of nodes and directed edges respectively, where each node corresponds to a gene and each edge represents some direct regulation between two genes. Self loops are not allowed in E although it is possible to modify the method so that self-loops are allowed. In this paper, n denotes the number of genes (i.e., the number of nodes) and we let $V = \{v_1, \ldots, v_n\}$. For each node v_i, $e^-(v_i)$ and $\deg^-(v_i)$, respectively, denote the set of incoming edges to v_i and the number of incoming edges to v_i as defined follows:

$$e^-(v_i) = \left\{v_j \mid \left(v_j, v_i\right) \in E\right\},$$
$$\deg^-(v_i) = \left|e^-(v_i)\right|. \tag{1}$$

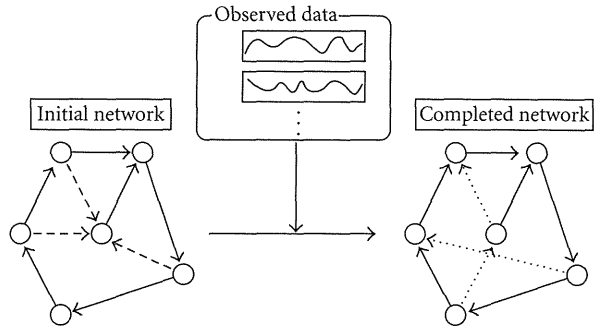

FIGURE 1: Network completion by addition and deletion of edges. Dashed edges and dotted edges denote deleted edges and added edges, respectively.

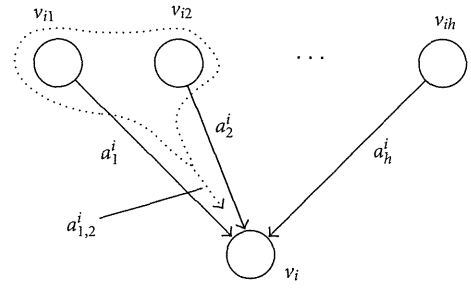

FIGURE 2: Dynamics model for a node.

DPLSQ consists of two parts: (i) parameter estimation and (ii) network structure inference. We employ least-squares fitting for the former part and dynamic programming for the latter part. Furthermore, there are three variants on the latter parts: (a) completion by addition of edges, (b) completion by deletion of edges, and (c) completion by addition and deletion of edges. Although the last case includes the first and second cases, we explain all of these for the sake of simplicity of explanation.

2.1. Model of Differential Equation and Estimation of Parameters. We assume that dynamics of each node v_i is determined by a differential equation:

$$\frac{dx_i}{dt} = a^i_0 + \sum_{j=1}^{h} a^i_j x_{i_j} + \sum_{j<k} a^i_{j,k} x_{i_j} x_{i_k} + b^i \omega, \tag{2}$$

where $v_{i_1}, \ldots, v_{i_h}$ are incoming nodes to v_i, x_i corresponds to the expression value of the ith gene, and ω denotes a random noise. The second and third terms of the right-hand side of the equation represent linear and nonlinear effects to node v_i, respectively (see Figure 2), where positive a^i_j or $a^i_{j,k}$ corresponds to an activation effect and negative a^i_j or $a^i_{j,k}$ corresponds to an inhibition effect.

In practice, we replace derivative by difference and ignore the noise term as follows:

$$x_i(t+1) = x_i(t) + \Delta t\left(a^i_0 + \sum_{j=1}^{h} a^i_j x_{i_j}(t) + \sum_{j<k} a^i_{j,k} x_{i_j}(t) x_{i_k}(t)\right), \tag{3}$$

where Δt denotes the time step.

We assume that time series data $y_i(t)$s, which correspond to $x_i(t)$s, are given for $t = 0, 1, \ldots, m$. Then, we can use the standard least-squares fitting method to estimate the parameters a_j^is and $a_{j,k}^i$s.

In applying the least-squares fitting method, we minimize the following objective function:

$$S_{i_1,i_2,\ldots,i_h}^i = \sum_{t=1}^{m} \left| y_i(t+1) - \left[y_i(t) + \Delta t \left(a_0^i + \sum_{j=1}^{h} a_j^i y_{i_j}(t) + \sum_{j<k} a_{j,k}^i y_{i_j}(t) y_{i_k}(t) \right) \right] \right|^2. \tag{4}$$

2.2. Completion by Addition of Edges. In this subsection, we consider the problem of adding k edges in total so that the sum of least-squares errors is minimized.

Let $\sigma_{k_j,j}^+$ denote the minimum least-squares error when adding k_j edges to the jth node, which is formally defined by

$$\sigma_{k_j,j}^+ = \min_{j_1,j_2,\ldots,j_{k_j}} S_{j_1,j_2,\ldots,j_{k_j}}^j, \tag{5}$$

where each v_{j_l} must be selected from $V - v_j - e^-(v_j)$. In order to avoid combinatorial explosion, we constrain the maximum k to be a small constant K and let $\sigma_{k_j,j}^+ = +\infty$ for $k_j > K$ or $k_j + \deg^-(v_j) \geq n$. Then, the problem is stated as

$$\min_{k_1+k_2+\cdots+k_n=k} \sum_{j=1}^{n} \sigma_{k_j,j}^+. \tag{6}$$

Here, we define $D^+[k,i]$ by

$$D^+[k,i] = \min_{k_1+k_2+\cdots+k_i=k} \sum_{j=1}^{i} \sigma_{k_j,j}^+. \tag{7}$$

Then, $D^+[k,n]$ is the objective value (i.e., the minimum of the sum of least-squares errors when adding k edges).

The entries of $D^+[k,j]$ can be computed by the following dynamic programming algorithm:

$$\begin{aligned} D^+[k,1] &= \sigma_{k,1}^+, \\ D^+[k,j+1] &= \min_{k'+k''=k} \left\{ D^+[k',j] + \sigma_{k'',j+1}^+ \right\}. \end{aligned} \tag{8}$$

It is to be noted that $D^+[k,n]$ is determined uniquely regardless of the ordering of nodes in the network. The correctness of this dynamic programming algorithm can be seen by

$$\begin{aligned} \min_{k_1+k_2+\cdots+k_n=k} \sum_{j=1}^{n} \sigma_{k_j,j}^+ &= \min_{k'+k''=k} \left\{ \min_{k_1+k_2+\cdots+k_{n-1}=k'} \sum_{j=1}^{n-1} \sigma_{k_j,j}^+ + \sigma_{k'',n}^+ \right\} \\ &= \min_{k'+k''=k} D^+[k', n-1] + \sigma_{k'',n}^+. \end{aligned} \tag{9}$$

2.3. Completion by Deletion of Edges. In the above, we considered network completion by addition of edges. Here, we consider the problem of deleting h edges in total so that the sum of least-squares errors is minimized.

Let $\sigma_{h_j,j}^-$ denote the minimum least-squares error when deleting h_j edges from the set $e^-(v)$ of incoming edges to v_j. As in Section 2.2, we constrain the maximum h_j to be a small constant H and let $\sigma_{h_j,j}^- = +\infty$ if $h_j > H$ or $\deg^-(v_j) - h_j < 0$. Then, the problem is stated as

$$\min_{h_1+h_2+\cdots+h_n=h} \sum_{j=1}^{n} \sigma_{k_j,j}^-. \tag{10}$$

Here, we define $D^-[k,i]$ by

$$D^-[k,i] = \min_{k_1+k_2+\cdots+k_i=k} \sum_{j=1}^{i} \sigma_{k_j,j}^-. \tag{11}$$

Then, we can solve network completion by deletion of edges using the following dynamic programming algorithm:

$$\begin{aligned} D^-[k,1] &= \sigma_{k,1}^-, \\ D^-[k,j+1] &= \min_{k'+k''=k} \left\{ D^-[k',j] + \sigma_{k'',j+1}^- \right\}. \end{aligned} \tag{12}$$

2.4. Completion by Addition and Deletion of Edges. We can combine the above two methods into network completion by addition and deletion of edges.

Let $\sigma_{h_j,k_j,j}$ denote the minimum least-squares error when deleting h_j edges from $e^-(v_j)$ and adding k_j edges to $e^-(v_j)$ where deleted and added edges must be disjoint. We constrain the maximum h_j and k_j to be small constants H and K. We let $\sigma_{h_j,k_j,j} = +\infty$ if $h_j > H$, $k_j > K$, $k_j - h_j + \deg^-(v_j) \geq n$, or $k_j - h_j + \deg^-(v_j) < 0$ holds. Then, the problem is stated as

$$\min_{\substack{h_1+h_2+\cdots+h_n=h \\ k_1+k_2+\cdots+k_n=k}} \sum_{j=1}^{n} \sigma_{h_j,k_j,j}. \tag{13}$$

Here, we define $D[h,k,i]$ by

$$D[h,k,i] \min_{\substack{h_1+h_2+\cdots+h_i=h \\ k_1+k_2+\cdots+k_i=k}} \sum_{j=1}^{i} \sigma_{h_j,k_j,j}. \tag{14}$$

Then, we can solve network completion by addition and deletion of edges using the following dynamic programming algorithm:

$$\begin{aligned} D[h,k,1] &= \sigma_{h,k,1}, \\ D[h,k,j+1] &= \min_{\substack{h'+h''=h \\ k'+k''=k}} \left\{ D[h',k',j] + \sigma_{h'',k'',j+1} \right\}. \end{aligned} \tag{15}$$

2.5. Time Complexity Analysis. In this subsection, we analyze the time complexity of DPLSQ. Since completion by addition of edges and completion by deletion of edges are special cases

of completion by addition and deletion of edges, we focus on completion by addition and deletion of edges.

First, we analyze the time complexity required per least-squares fitting. It is known that least-squares fitting for linear systems can be done in $O(mp^2 + p^3)$ time where m is the number of data points and p is the number of parameters. Since our model has $O(n^2)$ parameters, the time complexity is $O(mn^4+n^6)$. However, if we can assume that the maximum indegree in a given network is bounded by a constant, the number of parameters is bounded by a constant, where we have already assumed that H and K are constants. In this case, the time complexity for least-squares fitting can be estimated as $O(m)$.

Next, we analyze the time complexity required for computing $\sigma_{h_j,k_j,j}$. In this computation, we need to examine combinations of deletions of h_j edges and additions of k_j edges. Since h_j and k_j are, respectively, bounded by constants H and K, the number of combinations is $O(n^{H+K})$. Therefore, the computation time required per $\sigma_{h_j,k_j,j}$ is $O(n^{H+K}(mn^4 + n^6))$ including the time for least-squares fitting. Since we need to compute $\sigma_{h_j,k_j,j}$ for $H \times K \times n$ combinations, the total time required for computation of $\sigma_{h_j,k_j,j}$s is $O(n^{H+K+1}(mn^4 + n^6))$.

Finally, we analyze the time complexity required for computing $D[h,k,i]$s. We note that the size of table $D[h,k,i]$ is $O(n^3)$, where we are assuming that h and k are $O(n)$. In order to compute the minimum value for each entry in the dynamic programming procedure, we need to examine $(H + 1)(K + 1)$ combinations, which is $O(1)$. Therefore, the computation time required for computing $D[h,k,i]$s is $O(n^3)$. Since this value is clearly smaller than the one for $\sigma_{h_j,k_j,j}$s, the total time complexity is

$$O\left(n^{H+K+1} \cdot (mn^4 + n^6)\right). \tag{16}$$

Although this value is too high, it can be significantly reduced if we can assume that the maximum degree of an input network is bounded by a constant. In this case, the least-squares fitting can be done in $O(m)$ time per execution. Furthermore, the number of combinations of deleting at most h_j edges is bounded by a constant. Therefore, the time complexity required for computing $\sigma_{h_j,k_j,j}$s is reduced to $O(mn^{K+1})$. Since the time complexity for computing $D[h,k,i]$s remains $O(n^3)$, the total time complexity is

$$O\left(mn^{K+1} + n^3\right). \tag{17}$$

This number is allowable in practice if $K \leq 2$ and n is not too large (e.g., $n \leq 100$).

3. Results

We performed computational experiments using both artificial data and real data. All experiments on DPLSQ were performed on a PC with Intel Core i7-2630QM CPU (2.00 GHz) with 8 GB RAM running under the Cygwin on Windows 7. We employed the liblsq library (http://www2.nict.go.jp/aeri/sts/stmg/K5/VSSP/install_lsq.html) for a least-squares fitting method.

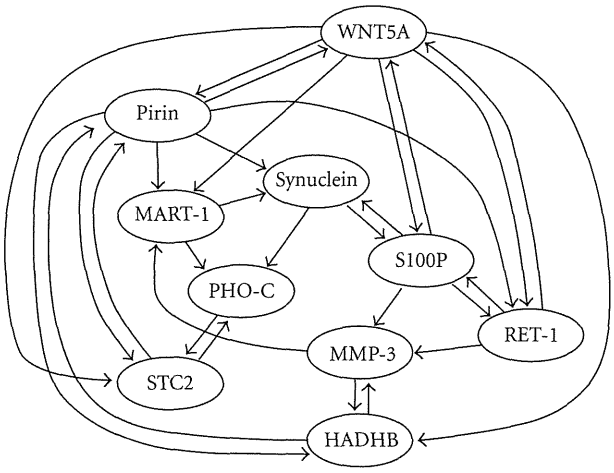

FIGURE 3: Structure of WNT5A network [17].

3.1. Completion Using Artificial Data. In order to evaluate the potential effectiveness of DPLSQ, we began with network completion using artificial data. To our knowledge, there is no available tool that performs the same task. Although some of the existing inference methods employ incremental modifications of networks, the number of added/deleted edges cannot be specified. Therefore, we did not compare DPLSQ for network completion with other methods (but we compared it with the existing tools for network inference).

We employed the structure of the real biological network named WNT5A (see Figure 3) [17]. For each node v_i with h input nodes, we considered the following model:

$$x_i(t+1) = x_i(t) + \Delta t\left(a_0^i + \sum_{j=1}^{h} a_j^i x_{i_j} + \sum_{j<k} a_{j,k}^i x_{i_j}(t)x_{i_k}(t) + b_i\omega\right), \tag{18}$$

where a_j^is and $a_{j,k}^i$s are constants selected uniformly at random from $[-1,1]$ and $[-0.5,0.5]$, respectively. The reason why the domain of $a_{j,k}^i$s is smaller than that for a_j^is is that non-linear terms are not considered as strong as linear terms. It should also be noted that $b_i\omega$ is a stochastic term, where b_i is a constant (we used $b_i = 0.2$ in all computational experiments) and ω is a random noise taken uniformly at random from $[-1,1]$.

For artificial generation of observed data $y_i(t)$, we used

$$y_i(t) = x_i(t) + o^i\epsilon, \tag{19}$$

where o^i is a constant denoting the level of observation errors and ϵ is a random noise taken uniformly at random from $[1,-1]$. Since the use of time series data beginning from only one set of initial values easily resulted in overfitting, we generated time series data beginning from 20 sets of initial values taken uniformly at random from $[1,-1]$, where the number of time points for each set was set to 10 and $\Delta t = 0.2$ was used as the period between the consecutive two time points. Therefore, 20 sets of time series data, each of which consisted of 10 time points, were used per trial (200 time points were used in total per trial). It is to be noted that in

our preliminary experiments, the use of too small Δt resulted in too small changes of expression values whereas the use of large Δt resulted in divergence of time series data. Therefore, after some trials, $\Delta t = 0.2$ was selected and used throughout the paper.

Under the above model, we examined several o^is as shown in Table 1. In order to examine network completion, WNT5A was modified by randomly adding h edges and deleting k edges and the resulting network was given as an initial network.

We evaluated the performance of the method in terms of the accuracy of the modified edges and the success rate. The accuracy is defined here by

$$\frac{h + k + \left|E_{\text{orig}}\right| - \left|E_{\text{orig}} \cap E_{\text{cmpl}}\right|}{h + k}, \tag{20}$$

where E_{orig} and E_{cmpl} are the sets of edges in the original network and the completed network, respectively. This value takes 1 if all deleted and added edges are correct and 0 if none of the deleted and added edges is correct. For each (h, k), we took the average accuracy over a combination of 10 parameters (a^i_js and $a^i_{j,k}$s) and 10 random modifications (i.e., addition of h edges and deletion of k edges to construct an initial network). The success rate is the frequency of the trials (among 10×10 trials) in which the original network was correctly obtained by network completion. The result is shown in Table 1. It is seen from this table that DPLSQ works well if the observation error level is small. It is also seen that the accuracies are high in the case of $h = 0$. However, no clear trend can be observed on a relationship between h, k values and the accuracies. It is reasonable because we evaluated the result in terms of the accuracy per deleted/added edge. On the other hand, it is seen that the success rate decreases considerably as h and k increase or the observation error level increases. This dependence on h and k is reasonable because the probability of having at least one wrong edge increases as the number of edges to be deleted and added increases. As for the computation time, the CPU time for each trial was within a few seconds, where we used the default values of $H = K = 3$. Although these default values were larger than h, k here, it did not cause any effects on the accuracy or the success rate. How to choose H and K is not a trivial problem. As discussed in Section 2.5, we cannot choose large H or K because of the time complexity issue. Therefore, it might be better in practice to examine several combinations of small values H and K and select the best result although how to determine the best result is left as another issue.

3.2. Inference Using Artificial Data. We also examined DPLSQ for network inference, using artificially generated time series data. In this case, we used the same network and dynamics model as previously mentioned but we let $E = \emptyset$ in the initial network. Since the method was applied to inference, we let $H = 0$, $K = 3$, and $k = 30$. It is to be noted that $\deg^-(v_i) = 3$ holds for all nodes v_i in the WNT5A network. Furthermore, in order to examine how CPU time changes as the size of the network grows, we made networks with 30 genes and 50 genes (with $k = 90$ and $k = 150$) by making 3 and 5 copies of the original networks, respectively.

Since the number of added edges was always equal to the number of edges in the original network, we evaluated the results by the average accuracy, which was defined as the ratio of the number of correctly inferred edges to the number of edges in the correct network (i.e., the number of added edges). We examined observation error levels of 0.1, 0.3, 0.5, and 0.7, for each of which we took the average over 10 trials using randomly generated different parameter values (i.e., a^i_js and $a^i_{j,k}$s), where time series data were generated as in Section 3.1. The result is shown in Table 2, where the accuracy and the average CPU time (user time + sys time) per trial are shown for each case. It is seen from the table that the accuracy is high even for large networks if the error level is not high. It is also seen that although the CPU time grows rapidly as the size of a network increases, it is still allowable for networks with 50 genes.

We also compared DPLSQ with two well-known existing tools for inference of genetic networks, ARACNE [11, 12] and GeneNet [7, 8]. The former is based on mutual information and the latter is based on graphical Gaussian models and partial correlations. Computational experiments on ARACNE were performed under the same environment as that for DPLSQ, whereas those on GeneNet were performed on a PC with Intel Core i7-2600 CPU (3.40 GHz) with 16 GB RAM running under the Cygwin on Windows 7 because of the availability of the R platform on which GeneNet works. We employed datasets that were generated in the same way as for DPLSQ and default parameter settings for both tools.

Since both tools output undirected edges along with their significance values (or their probabilities), we selected the top M edges in the output where M was the number of edges in the original network and regarded $\{v_i, v_j\}$ as a correct edge if either (v_i, v_j) or (v_j, v_i) was included in the edge set of the original network. As in Table 2, we evaluated the results by the average accuracy, that is, the ratio of the number of correctly inferred edges to the number of edges in the original network.

The result is shown in Table 3. Interestingly, both tools have similar performances. It is also interesting that the performance does not change much in each method even if the level of observation error changes. Readers may think that the accuracies shown in Table 3 are close to those by random prediction. However, these accuracies were much higher than those obtained by assigning random probabilities to edges, and thus we can mention that these tools outputted meaningful results.

It is seen from Tables 2 and 3 that the accuracies by DPLSQ are much higher than those by ARACNE and GeneNet even though both directions of edges are taken into account for ARACNE and GeneNet. However, it should be noted that time series data were generated according to the differential equation model on which DPLSQ relies. Therefore, we can only mention that DPLSQ works well if time series data are generated according to appropriate differential equation models. It is to be noted that we can use

Table 1: Result on completion of WNT5A network, where the average accuracy is shown for each case.

No. deleted edges	No. added edges		Observation error level 0.1	0.3	0.5	0.7
$h = 0$	$k = 1$	Accuracy	0.990	0.910	0.730	0.410
		Success rate	0.99	0.91	0.73	0.41
$h = 0$	$k = 2$	Accuracy	1.000	0.955	0.670	0.395
		Success rate	1.00	0.91	0.42	0.17
$h = 1$	$k = 0$	Accuracy	0.990	0.850	0.470	0.240
		Success rate	0.99	0.85	0.47	0.24
$h = 1$	$k = 1$	Accuracy	0.995	0.845	0.405	0.210
		Success rate	0.99	0.71	0.11	0.02
$h = 1$	$k = 2$	Accuracy	0.983	0.843	0.470	0.190
		Success rate	0.95	0.58	0.11	0.00
$h = 2$	$k = 0$	Accuracy	1.000	0.795	0.440	0.215
		Success rate	1.00	0.67	0.18	0.01
$h = 2$	$k = 1$	Accuracy	0.996	0.833	0.453	0.223
		Success rate	0.99	0.53	0.05	0.01
$h = 2$	$k = 2$	Accuracy	1.000	0.862	0.517	0.285
		Success rate	1.00	0.56	0.03	0.01

Table 2: Result on inference of WNT5A network by DPLSQ.

		Observation error level 0.1	0.3	0.5	0.7
$n = 10$	Accuracy	1.000	0.966	0.803	0.620
	CPU time (sec.)	0.685	0.682	0.682	0.685
$n = 30$	Accuracy	0.995	0.914	0.663	0.443
	CPU time (sec.)	66.2	66.2	66.1	65.9
$n = 50$	Accuracy	0.999	0.913	0.613	0.392
	CPU time (sec.)	534.0	534.2	533.6	533.5

Table 3: Result on inference of WNT5A network using ARACNE and GeneNet, where the accuracy is shown for each case.

	Method	Observation error level 0.1	0.3	0.5	0.7
$n = 10$	ARACNE	0.523	0.523	0.523	0.526
	GeneNet	0.526	0.526	0.533	0.533
$n = 30$	ARACNE	0.332	0.328	0.326	0.326
	GeneNet	0.368	0.380	0.383	0.384
$n = 50$	ARACNE	0.308	0.312	0.310	0.391
	GeneNet	0.313	0.316	0.314	0.316

other differential equation models as long as parameters can be estimated by least-squares fitting.

As for computation time, both methods were much faster than DPLSQ. Even for the case of $N = 50$, each of ARACNE and GeneNet worked in less than a few seconds per trial. Therefore, DPLSQ does not have merits on practical computation time.

3.3. Inference Using Real Data. We also examined DPLSQ for inference of genetic networks using real gene expression data. Since there is no gold standard on genetic networks and thus we cannot know the correct answers, we did not compare it with the existing methods.

We employed a part of the cell cycle network of *Saccharomyces cerevisiae* extracted from the KEGG database [18], which is shown in Figure 4. Although the detailed mechanism of the cell cycle network is still unclear, we used this network as the correct answer, which may not be true. Although each of (MCM1, YOX1, YHP1), (SWI4, SWI6), (CLN3, CDC28), (MBP1, SWI6) constitutes a protein complex, we treated them separately and ignored the interactions inside a complex because making a protein complex does not necessarily mean a regulation relationship between the corresponding genes.

As for time series data of gene expression, we employed four sets of times series data (alpha, cdc15, cdc28, elu) in [19] that were obtained by four different experiments. Since there were several missing values in the time series data, these values were filled by linear interpolation and data on some endpoint time points were discarded because of too many missing values. As a result, alpha, cdc15, cdc28, and elu datasets consist of gene expression data of 18, 24, 11, and 14 time points, respectively. In order to examine a relationship between the number of time points, and accuracy, we examined four combinations of datasets as shown in Table 4. We evaluated the performance of DPLSQ by means of the accuracy (i.e., the ratio of the number of correctly inferred edges to the number of added edges), where $K = 3$ and $k = 25$ were used. The result is shown in Table 4.

Since the total number of edges in both the original network and the inferred networks is 25 and there exist

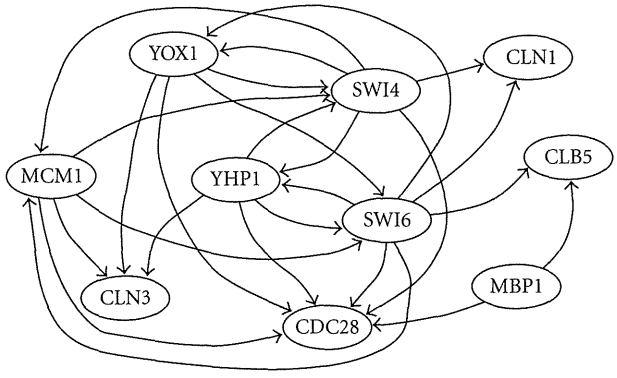

FIGURE 4: Structure of part of yeast cell cycle network.

$9 \times 10 = 90$ possible edges (excluding self loops), the expected number of corrected edges is roughly estimated as

$$\frac{25}{90} \times 25 = 6.944\ldots, \tag{21}$$

if 25 edges are randomly selected and added. Therefore, the result shown in Table 4 suggests that DPLSQ can do much better than random inference when appropriate datasets are provided (e.g., cdc15 only or cdc15+cdc28+alpha+elu). It is a bit strange that the accuracies for the first and last datasets are better than those for the second and third datasets because it is usually expected that adding more evidences results in more accurate networks. The reason may be that time series of cdc28 and alpha may contain larger measurement errors than those of cdc15 and elu, or some regulation rules that are hidden in Figure 4 may be activated under the conditions of cdc28 and/or alpha.

4. Conclusion

In this paper, we have proposed a network completion method, DPLSQ, using dynamic programming and least-squares fitting based on our previously proposed methodology of network completion [14]. As mentioned in Section 1, network completion is to make the minimum amount of modifications to a given network so that the resulting network is (most) consistent with the observed data. In our previous model [14], we employed the Boolean network as a model of networks and assumed that only expression or other values of one or a few nodes are observed. However, in this paper, we assumed that expression values of all nodes are observed, which correspond to gene microarray data, and regulation rules are given in the form of differential equations. The most important theoretical difference between this model and our previous model is that network completion can be done in polynomial time if the maximum indegree is bounded by a constant in this model whereas it is NP-hard in our previous model even if the maximum indegree is bounded by a constant. This difference arises not from the introduction of a least-squares fitting method but from the assumption that expression values of all nodes are observed.

It should also be noted that the optimality of the solution is not guaranteed in most of the existing methods for inference of genetic networks, whereas it is guaranteed in DPLSQ if it is applied to inference of a genetic network with a bounded maximum indegree. Of course, the objective function (i.e., minimizing the sum of squared errors) is different from existing ones, and thus this property does not necessarily mean that DPLSQ is superior to existing methods in real applications. Indeed, the result using real gene expression data in Section 3.3 does not seem to be very good. However, DPLSQ has much room for extensions. For example, least-squares fitting can be replaced by another fitting/regression method (with some regularization term) and the objective function can be replaced by another function as long as it can be computed by sum or product of some error terms. Studies on such extensions might lead to development of better network completion and/or inference methods.

TABLE 4: Result on inference of a yeast cell cycle network.

Experimental conditions	Accuracy
cdc15	11/25
cdc15 + cdc28	8/25
cdc15 + cdc28 + alpha	8/25
cdc15 + cdc28 + alpha + elu	11/25

Acknowledgments

T. Akutsu was partially supported by JSPS, Japan (Grants-in-Aid 22240009 and 22650045). T. Tamura was partially supported by JSPS, Japan (Grant-in-Aid for Young Scientists (B) 23700017). K. Horimoto was partially supported by the Chinese Academy of Sciences Visiting Professorship for Senior International Scientists Grant no. 2012T1S0012.

References

[1] S. Liang, S. Fuhrman, and R. Somogyi, "REVEAL, a general reverse engineering algorithm for inference of genetic network architectures," in *Proceedings of the Pacific Symposium on Biocomputing*, vol. 3, pp. 18–29, 1998.

[2] T. Akutsu, S. Miyano, and S. Kuhara, "Inferring qualitative relations in genetic networks and metabolic pathways," *Bioinformatics*, vol. 16, no. 8, pp. 727–734, 2000.

[3] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 601–620, 2000.

[4] S. Imoto, S. Kim, T. Goto et al., "Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network," *Journal of Bioinformatics and Computational Biology*, vol. 1, no. 2, pp. 231–252, 2003.

[5] T. F. Liu, W. K. Sung, and A. Mittal, "Learning gene network using time-delayed Bayesian network," *International Journal on Artificial Intelligence Tools*, vol. 15, no. 3, pp. 353–370, 2006.

[6] H. Toh and K. Horimoto, "Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling," *Bioinformatics*, vol. 18, no. 2, pp. 287–297, 2002.

[7] R. Opgen-Rhein and K. Strimmer, "Inferring gene dependency networks from genomic longitudinal data: a functional data approach," *RevStat*, vol. 4, no. 1, pp. 53–65, 2006.

[8] R. Opgen-Rhein and K. Strimmer, "From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data," *BMC Systems Biology*, vol. 1, article 37, 2007.

[9] P. D'Haeseleer, S. Liang, and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, 2000.

[10] Y. Wang, T. Joshi, X. S. Zhang, D. Xu, and L. Chen, "Inferring gene regulatory networks from multiple microarray datasets," *Bioinformatics*, vol. 22, no. 19, pp. 2413–2420, 2006.

[11] A. A. Margolin, K. Wang, W. K. Lim, M. Kustagi, I. Nemenman, and A. Califano, "Reverse engineering cellular networks," *Nature Protocols*, vol. 1, no. 2, pp. 662–671, 2006.

[12] A. A. Margolin, I. Nemenman, K. Basso et al., "ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context," *BMC Bioinformatics*, vol. 8, supplement 1, no. 1, article S7, 2006.

[13] S. Kimura, S. Nakayama, and M. Hatakeyama, "Genetic network inference as a series of discrimination tasks," *Bioinformatics*, vol. 25, no. 7, pp. 918–925, 2009.

[14] T. Akutsu, T. Tamura, and K. Horimoto, "Completing networks using observed data," *Lecture Notes in Artificial Intelligence*, vol. 5809, pp. 126–140, 2009.

[15] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York, NY, USA, 1993.

[16] T. Tamura, Y. Yamanishi, M. Tanabe et al., "Integer programming-based method for completing signaling pathways and its application to analysis of colorectal cancer," *Genome Informatics*, vol. 24, pp. 193–203, 2010.

[17] S. Kim, H. Li, E. R. Dougherty et al., "Can Markov chain models mimic biological regulation?" *Journal of Biological Systems*, vol. 10, no. 4, pp. 337–357, 2002.

[18] M. Kanehisa, S. Goto, M. Furumichi, M. Tanabe, and M. Hirakawa, "KEGG for representation and analysis of molecular networks involving diseases and drugs," *Nucleic Acids Research*, vol. 38, no. 1, Article ID gkp896, pp. D355–D360, 2009.

[19] P. T. Spellman, G. Sherlock, M. Q. Zhang et al., "Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization," *Molecular Biology of the Cell*, vol. 9, no. 12, pp. 3273–3297, 1998.