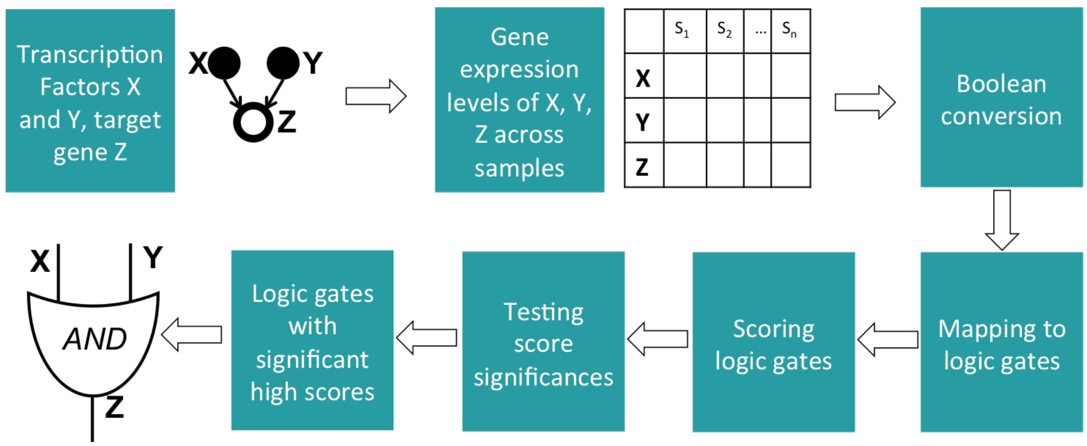


Fig. 1. Flowchart of Loregic. Given a RF-RF-target triplet, we convert their gene expression levels to Boolean values based on expression changes across samples. We then map them via scoring to all 16 possible logic gates with 2-input-1-output. We use permutation tests to test score significances, and finally match the triplet to the logic gates with significant high scores; e.g., the AND logic gate is the match in figure.

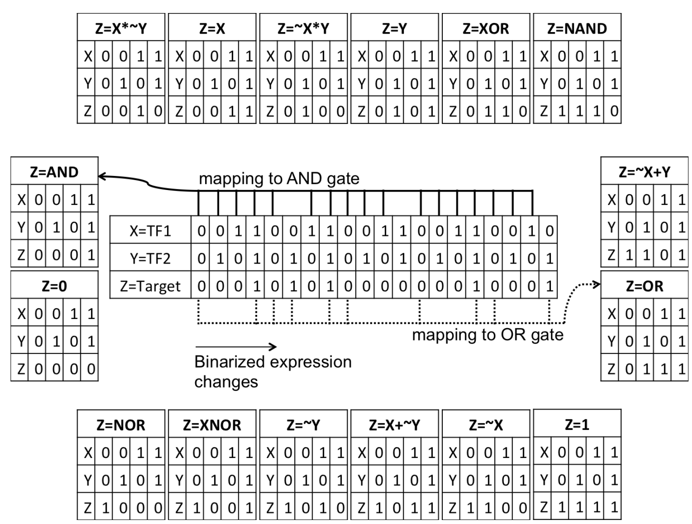


Fig. 2. Truth tables of all 16 possible logic gates with 2-input-1-output. Each logic gate is uniquely determined by combination of four different (X, Y, Z) binary vectors (columns in truth table). Suppose that we have a RF-RI target triplet shown by the middle table, (X=RF1, Y=RF2, Z=their target gene) has m=20 binary vectors aft conversion (horizontal). 17 out of 20 vectors highlighted by solids lines can match the AND gate, Z=X*Y; i.e., bot X and Y must present to activate Z to express. 10 out of 20 vectors highlighted by dash lines can match the O gate, Z=X+Y; i.e., either X or Y presents to activate Z to express. Here, '~' denotes NOT (negative regulation), '' denotes AND and '+' denotes OR logic operations.

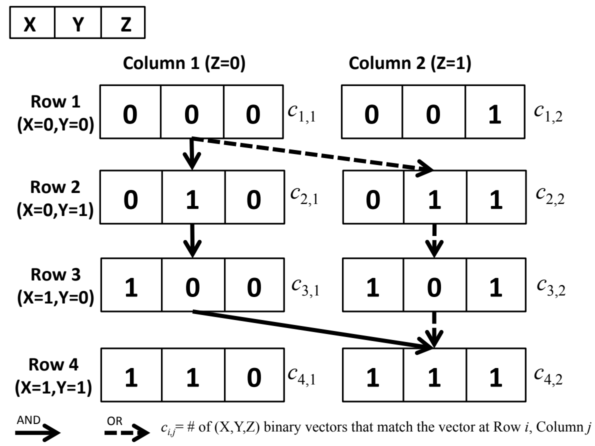


Fig. 3. Weight calculation for a triplet, (X, Y, Z). We build a matrix with 4 rows and 2 columns. The matrix elements cover all 8 different (X, Y, Z) binary vectors. The two elements at the same row share the same X and Y values (1st row: X=0, Y=0; 2nd row: X=0, Y=1; 3rd row: X=1, Y=0; 4th row: X=1, Y=1), and the four elements at the same column share the same Z value (1st column: Z=0; 2nd column: Z=1). We assign a binary *c*-value to each element to indicate if number of the element appearing in (X, Y, Z) vectors is greater than the other element at the same row with different output Z value. For the element at ith row, 1st column, if its appearances out of m (X, Y, Z) binary vectors are more than the element at 2nd column (same X-Y inputs, different Z output), we let $c_{i,1}$=1 and $c_{i,2}$=0; i.e., with the same X and Y inputs, output Z is more likely to be zero. If less, we let $c_{i,1}$=0 and $c_{i,2}$=1; i.e., with the same X and Y in-puts, output Z is more likely to be one. If equal or it happens that both elements at the same row miss, then we assume that both outputs are possible so that we let $c_{i,1}=c_{i,2}$=1; i.e., with the same X and Y in-puts, output Z is equally likely to be one or zero. The truth table of any one of 16 logic gates corresponds to a unique pathway from 1st row to 4th row that have 4 elements from different rows. We assign a weight, w to each logic gate, as the product of *c*-values of four elements on its corresponding pathway. The weight, also a binary number, indicates if four outputs (Z values) of its logic gate are no less than other logic gates in the triplet; e.g., w(AND)= $c_{1,1}$* $c_{2,1}$* $c_{3,1}$* $c_{4,2}$, and w(OR)= $c_{1,1}$* $c_{2,2}$* $c_{3,2}$* $c_{4,2}$ (see Table 1 for other logic gates).

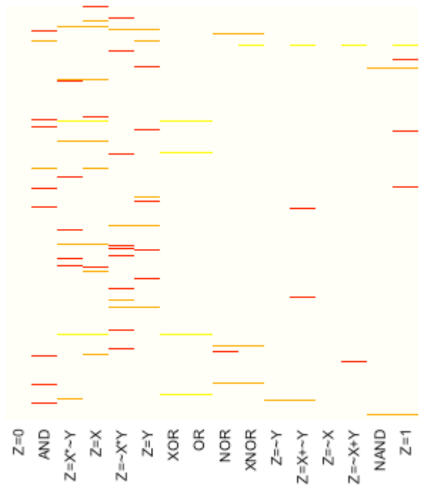


Fig. 4. Scores of 16 logic gates for ~39k TF-TF-target triplets with significant scores. In heatmap, rows represent triplets, and columns represent logic gates. The dark colors correspond to high scores. We found that cooperative logic gates, especially the gates of AND (i.e., Z=X*Y), Z=~X*Y and Z=X*~Y, have significantly higher scores than others

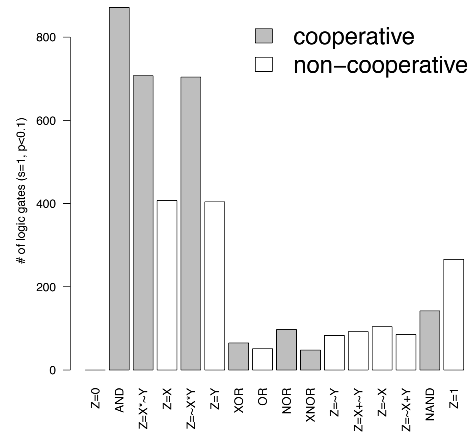
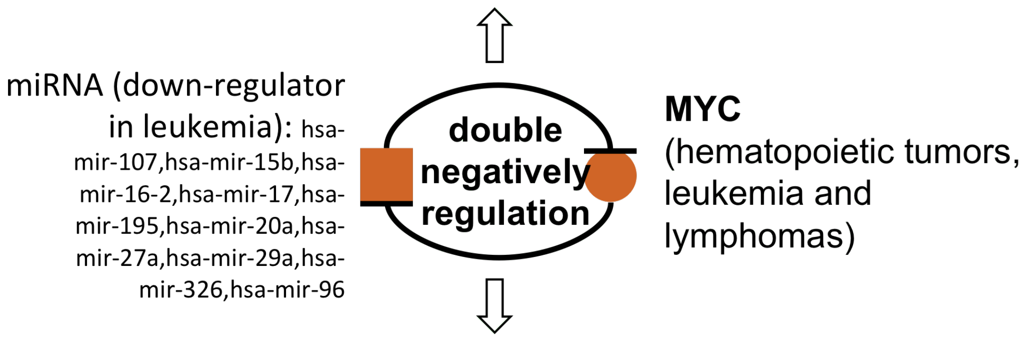


Fig. 5. Numbers of logic gates with significant highest scores (*s*=1). We define the logic gates as the cooperative logic gates (grey bars) if their inputs have AND logic relationship, and other logic gates as non-cooperative ones (white bars). Out of those highest-score logic gates, we found more cooperative logic gates than non-cooperative ones. Also, we have more AND gates than others, in which that both TFs have to be present to activate their target gene to express.

Z=~X*Y		
X=miRNA	Y=MYC	Z=Targets
0	0	0
0	1	1
1	0	0
1	1	0

Z=~X		
X=miRNA	Y=MYC	Z=Targets
0	0	1
0	1	1
1	0	0
1	1	0

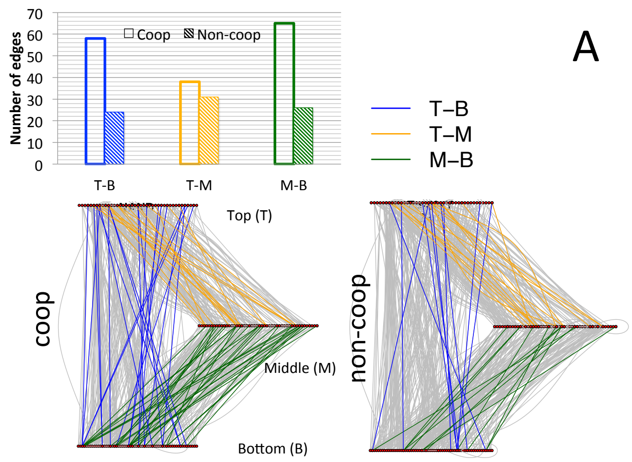


Z=Y		
X=miRNA	Y=MYC	Z=Targets
0	0	0
0	1	1
1	0	0
1	1	1

Z=~X+Y		
X=miRNA	Y=MYC	Z=Targets
0	0	1
0	1	1
1	0	0
1	1	1

Z=~X*Y and Z=Y are two most logic gates with significant highest scores for miRNA-MYC pairs

Fig. 6. Matched logic gates support that miRNAs and MYC form a double down-regulating loop. There are 1143 out of 1805 miRNA-TF-target triplets assigned the significant highest scores (s=1) for one logic gate. Out of 1143 triplets, 446 ones match Z=MYC, and 201 ones match Z=~miRNA+MYC, two most logic gates, both of which imply that MYC successfully down-regulates miRNAs so that target gene expressions can be turned on without down-regulations from miRNA. We also found that there were 56 triplets matching Z=~miRNA*MYC, and 16 triplets matching Z=~miRNA, which suggests that those targets cannot be expressed when miRNAs down-regulate MYC.



A

B

Louise is running permutation tests for all human TF-TF-target and miRNA-TF-target triplets (~100k x 1000 times)

Fig. 8. Hierarchical regulatory networks (A. yeast, B. human K562). We assigned TFs (red) to three hierarchical levels: top, middle and bottom using simulated annealing method (Gerstein, et al., 2012). The hierarchical networks highlight edges associated with TFs from cooperative (coop, left) and non-cooperative (non-coop, right) logic gates with significant highest scores (s=1). The edges from top to middle, from middle to bottom, and from top to bottom are highlighted by orange, green and blue, respectively. For yeast (Fig. 8A), we found that 58 cooperative and 24 non-cooperative edges were between top and bottom levels, 38 cooperative and 31 non-cooperative edges were between top and middle levels, and 65 cooperative and 26 non-cooperative edges were between middle and bottom levels. For human (Figs. 8B), we found that XXX cooperative and XXX non-cooperative edges were between top and bottom levels, XXX cooperative and XXX non-cooperative edges were between top and middle levels, and XXX cooperative and XXX non-cooperative edges were between middle and bottom levels.

Table 1. Weights of 16 logic gates

Gate	Weight	Gate	Weight	Gate	Weight	Gate	Weight
Z=0	$c_{1,1}* c_{2,1}* c_{3,1}* c_{4,1}$	NOR	$c_{1,2}* c_{2,1}* c_{3,1}* c_{4,1}$	Z=~X*	$c_{1,1}* c_{2,2}* c_{3,1}* c_{4,1}$	Z=~X	$c_{1,2}* c_{2,2}* c_{3,1}* c_{4,1}$
AND	$c_{1,1}* c_{2,1}* c_{3,1}* c_{4,2}$	XNOR	$c_{1,2}* c_{2,1}* c_{3,1}* c_{4,2}$	Z=Y	$c_{1,1}* c_{2,2}* c_{3,1}* c_{4,2}$	Z=~X+	$c_{1,2}* c_{2,2}* c_{3,1}* c_{4,2}$
Z=X*~Y	$c_{1,1}* c_{2,1}* c_{3,2}* c_{4,1}$	Z=~Y	$c_{1,2}* c_{2,1}* c_{3,2}* c_{4,1}$	XOR	$c_{1,1}* c_{2,2}* c_{3,2}* c_{4,1}$	NAND	$c_{1,2}* c_{2,2}* c_{3,2}* c_{4,1}$
Z=X	$c_{1,1}* c_{2,1}* c_{3,2}* c_{4,2}$	Z=X+~	$c_{1,2}* c_{2,1}* c_{3,2}* c_{4,2}$	OR	$c_{1,1}* c_{2,2}* c_{3,2}* c_{4,2}$	Z=1	$c_{1,2}* c_{2,2}* c_{3,2}* c_{4,2}$