Supplementary Material for
"RESQUE: Network reduction using semi-Markov random walk scores for efficient querying of biological networks"

Sayed Mohammad Ebrahim Sahraeian and Byung-Jun Yoon

Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA

S1 Analytical study of the performance of the iterative reduction scheme

Here, we want to show that the proposed reduction scheme improves the expected accuracy of the querying result.

Let A represent the space of all possible matchings between the query and the target networks, where a matching a uniquely maps the query nodes $\mathcal{V}_\mathcal{Q}$ to a subgraph region in the target network. Besides, let a^* be the (unknown) matching from A that most nearly represents the *true* biological querying result of $\mathcal{G}_\mathcal{Q}$ in $\mathcal{G}$. Here, we seek an scheme that yields a matching $a \in A$ that maximizes the *accuracy* of the querying result, which we define with respect to a^* as:

$$\text{accuracy}(a, a^*) = \frac{1}{N_\mathcal{Q}} \sum_{q_i \sim v_j \in a} \mathbf{1}\{q_i \sim v_j \in a^*\} \tag{1}$$

where $q_i \sim v_j$ indicates the alignment between $q_i \in \mathcal{V}_\mathcal{Q}$ and $v_j \in \mathcal{V}$, and $\mathbf{1}\{\cdot\}$ is the indicator function which equals 1 whenever its argument is true and 0 otherwise.

However, in a practical querying problem, a^* is unknown, so instead we aim at maximizing the *expected accuracy* of the querying result, which can be computed as:

$$\begin{aligned} \mathbf{E}_{a^*}[\text{accuracy}(a, a^*)|\mathcal{G}_\mathcal{Q}, \mathcal{G}] &= \frac{1}{N_\mathcal{Q}} \sum_{q_i \sim v_j \in a} \mathbf{E}_{a^*}[\mathbf{1}\{q_i \sim v_j \in a^*\}|\mathcal{G}_\mathcal{Q}, \mathcal{G}] \\ &= \frac{1}{N_\mathcal{Q}} \sum_{q_i \sim v_j \in a} P(q_i \sim v_j \in a^*|\mathcal{G}_\mathcal{Q}, \mathcal{G}) \end{aligned} \tag{2}$$

where, $P(q_i \sim v_j \in a^*|\mathcal{G}_\mathcal{Q}, \mathcal{G})$ is the *posterior probability* of matching q_i to v_j.

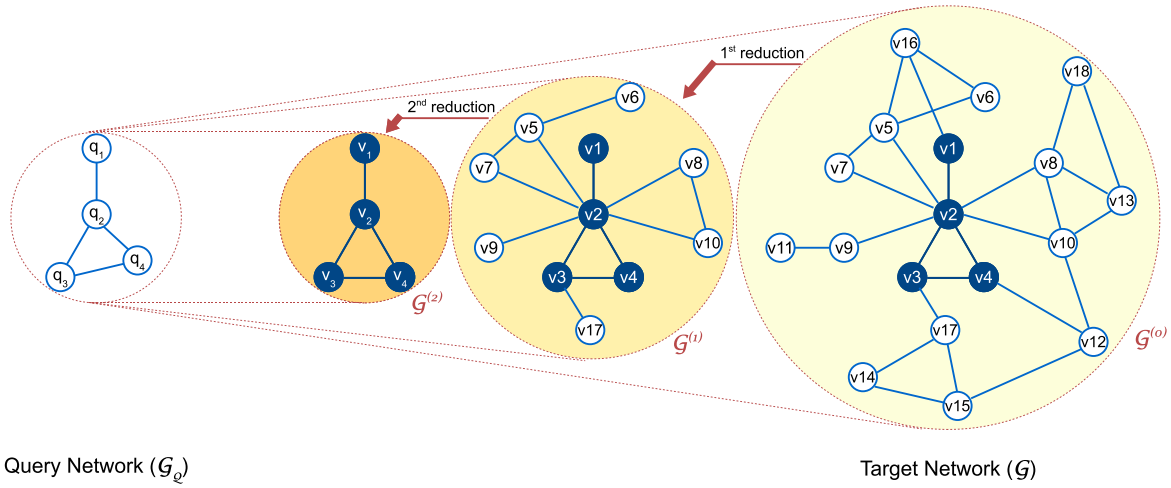

Figure S1: Detection of the matching subnetwork through iterative network reduction.

If we define a bipartite graph with the node set $\mathcal{V}_\mathcal{Q} \bigcup \mathcal{V}$ and a set of weighted edges joining each pair of nodes $q_i \in \mathcal{V}_\mathcal{Q}$ and $v_j \in \mathcal{V}$ with the weight given as $\omega_{i,j} = P(q_i \sim v_j \in a^* | \mathcal{G}_\mathcal{Q}, \mathcal{G})$, the above maximum expectation accuracy problem can be formulated as the maximum–weight bipartite matching, for which the solution can be obtained using the well–known *Hungarian* algorithm in polynomial time [1].

The global correspondence score $\mathbf{S}[i,j]$ can serve as a good estimate of the posterior probability $P(q_i \sim v_j \in a^*)$. Thus, employing the MWM algorithm on the above bipartite graph with weights defined as $\omega_{i,j} = \mathbf{S}[i,j]$ can maximize the expected accuracy.

For any query node q_i, we denote the true matching node of q_i in the target network as v_{i^*} (i.e., $q_i \sim v_{i^*} \in a^*$). We define our objective function as $f(\mathbf{S}) = \sum_{i=1}^{N_\mathcal{Q}} \mathbf{S}[i, i^*]$, constrained on the constant L_1 norm of $\mathbf{S}$ (i.e., $\|\mathbf{S}\|_1 = 1$). Therefore, the objective function $f(\mathbf{S})$ is proportional to the expected accuracy defined in (2)). Thus, using the MWM on the bipartite graph with edge weights $\omega_{i,j} = \mathbf{S}^*[i,j]$, where $\mathbf{S}^* = \underset{\mathbf{S}}{\operatorname{argmax}}\, f(\mathbf{S})$, will lead us to the maximum expected accuracy solution.

Here, we will show that the proposed network reduction scheme enhances the objective function $f(\mathbf{S})$ in each iteration, which means that it improves the estimation of the correspondences between the query and target networks, thereby yielding more accurate results. In this discussion, we assume the unit interaction reliabilities w for the target network edges. Extension to more general networks is straightforward.

In the k^{th} iteration, we pick a target node $v_d \in \mathcal{V}^{(k-1)}$ which has minimal correspondence

to the query network, as follows:

$$\begin{aligned} d &= \operatorname*{argmin}_j P(v_j \in a^*) = \operatorname*{argmin}_j P(\exists q_i \in \mathcal{V}_\mathcal{Q}, q_i \sim v_j \in a^*) \\ &= \operatorname*{argmin}_j \left(1 - \prod_{q_i \in \mathcal{V}_\mathcal{Q}} [1 - P(q_i \sim v_j \in a^*)]\right) \\ &\simeq \operatorname*{argmin}_j \left(1 - \prod_{q_i \in \mathcal{V}_\mathcal{Q}} [1 - \mathbf{S}^{(k)}[i,j]]\right), \end{aligned} \tag{3}$$

where we assumed the correspondence scores computed in the k^{th} iteration, as an estimator of the posterior matching probabilities. We then discard v_d from $\mathcal{V}^{(k-1)}$. As discussed before, we also discard the set of all newly isolated nodes $\mathcal{V}_\mathcal{O} = \{v_{d'_1}, v_{d'_2} \cdots\}$, if they are not a best match to some query node. The nodes in $\mathcal{V}_\mathcal{O}$ are identified as the nodes in $\mathcal{V}^{(k-1)}$ whose only neighbor is v_d. With the above scheme we minimize the chance of discarding a core target node. Following the deletion of these nodes, we recompute the new correspondence scores $\mathbf{S}^{(k+1)}$ to be used in the next iteration.

We can measure the difference in the objective function in two consecutive iteration steps as:

$$\begin{aligned} D^{(k+1)} &= f(\mathbf{S}^{(k+1)}) - f(\mathbf{S}^{(k)}) = \sum_{i=1}^{N_\mathcal{Q}} \mathbf{S}^{(k+1)}[i,i^*] - \mathbf{S}^{(k)}[i,i^*] \\ &= \sum_{i=1}^{N_\mathcal{Q}} \frac{\mathbf{P}^{(k+1)}[i,i^*]\mathbf{H}[i,i^*]}{\text{trace}(\mathbf{P}^{(k+1)}\mathbf{H}^T)} - \frac{\mathbf{P}^{(k)}[i,i^*]\mathbf{H}[i,i^*]}{\text{trace}(\mathbf{P}^{(k)}\mathbf{H}^T)} \\ &\simeq \frac{1}{\text{trace}(\mathbf{P}^{(k+1)}\mathbf{H}^T)} \sum_{i=1}^{N_\mathcal{Q}} \pi_\mathcal{Q}(q_i) h(q_i, v_{i^*}) [\pi^{(k+1)}(v_{i^*}) - \pi^{(k)}(v_{i^*})], \end{aligned} \tag{4}$$

where $\pi^{(k+1)}(v_{i^*})$ is the steady state distribution of node $v_{i^*} \in \mathcal{V}$ based on the reduced target network in k^{th} iteration. Here, we assumed $\text{trace}(\mathbf{P}^{(k)}\mathbf{H}^T)$ to be constant over successive iterations. If $D^{(k+1)} \geq 0$, we can say that the proposed iterative approach improves the objective function in each iteration.

To have $D^{(k+1)} \geq 0$, we need the steady state distribution of the core target nodes $\pi^{(k)}(v_{i^*})$ to be increasing over iterations. In general network structure, omission of one node v_d, will generally increase the long-run proportion of time spent on most of other nodes. Only, for a few nodes which have v_d as the source of their connection to the major part of the network, this may cause a degradation in their steady state distribution. Considering the fact that v_d has a minimal correspondence to the query, it has distance relation to the core nodes of the target network. Thus, here, we argue on $\tilde{f}(\mathcal{G}^{(k)}) = \sum_{i=1}^{N_\mathcal{Q}} \pi^{(k+1)}(v_{i^*})$, sum of the steady state distribution over the core nodes. This can be viewed as the proportion of time that a random walker on the target network spends on the core region. We aim to show that this measure

increases through the proposed iterative approach. Thus, we define $\tilde{D}^{(k+1)} = \tilde{f}(\mathcal{G}^{(k+1)}) - \tilde{f}(\mathcal{G}^{(k)})$, and we show $\tilde{D}^{(k+1)} \geq 0$, which can result in $D^{(k+1)} \geq 0$ in general networks.

We know that the transition probability $a(j,l)$ that measures the normalized contribution from the neighboring node v_l to the node v_j, can be computed based on the interaction reliability $w(v_j, v_l)$ as:

$$a(j,l) = \frac{w(v_j, v_l)}{\sum_{v_{l'} \in \mathcal{N}(v_j)} w(v_j, v_{l'})}. \tag{5}$$

where $\mathcal{N}(v_j)$ is the set of neighbors of v_i. Thus, for the unit interaction reliabilities scores w, the steady state distribution for a node $v_j \in \mathcal{V}$ is defined as

$$\pi(v_j) = \sum_{v_l \in \mathcal{N}(v_j)} \frac{1}{n_l} \pi(v_l), \tag{6}$$

where, n_l is the number of neighbors of v_l. Thus, we can write:

$$\begin{aligned} \pi^{(k+1)}(v_{i^*}) - \pi^{(k)}(v_{i^*}) &= \sum_{\substack{v_l \in \mathcal{N}(v_{i^*}) \\ , v_l \notin \mathcal{N}(v_d)}} \frac{1}{n_l} [\pi^{(k+1)}(v_l) - \pi^{(k)}(v_l)] \\ &+ \sum_{\substack{v_l \in \mathcal{N}(v_{i^*}) \\ , v_l \in \mathcal{N}(v_d)}} [\frac{\pi^{(k+1)}(v_l)}{n_l - 1} - \frac{\pi^{(k)}(v_l)}{n_l}] - \frac{\pi^{(k)}(v_d)}{n_d} \mathbf{1}\{v_d \in \mathcal{N}(v_{i^*})\}, \end{aligned} \tag{7}$$

an, we can write $\tilde{D}^{(k+1)}$ as:

$$\begin{aligned} \tilde{D}^{(k+1)} &= \sum_{i=1}^{N_{\mathcal{Q}}} [\pi^{(k+1)}(v_{i^*}) - \pi^{(k)}(v_{i^*})] \\ &= \sum_{l=1}^{|\mathcal{V}^{(k)}|} \left[\frac{1}{n_l} [\pi^{(k+1)}(v_l) - \pi^{(k)}(v_l)] \mathbf{1}\{v_l \notin \mathcal{N}(v_d)\} \sum_{i=1}^{N_{\mathcal{Q}}} \mathbf{1}\{v_l \in \mathcal{N}(v_{i^*})\} \right] \\ &+ \sum_{l=1}^{|\mathcal{V}^{(k)}|} \left[[\frac{\pi^{(k+1)}(v_l)}{n_l - 1} - \frac{\pi^{(k)}(v_l)}{n_l}] \mathbf{1}\{v_l \in \mathcal{N}(v_d)\} \sum_{i=1}^{N_{\mathcal{Q}}} \mathbf{1}\{v_l \in \mathcal{N}(v_{i^*})\} \right] \\ &- \frac{\pi^{(k)}(v_d)}{n_d} \sum_{i=1}^{N_{\mathcal{Q}}} \mathbf{1}\{v_d \in \mathcal{N}(v_{i^*})\} \end{aligned} \tag{8}$$

which can be simplified as:

$$\tilde{D}^{(k+1)} = \underbrace{\sum_{l=1}^{|\mathcal{V}^{(k)}|} [\pi^{(k+1)}(v_l) - \pi^{(k)}(v_l)] \cdot \frac{\lambda_l}{n_l}}_{X} + \underbrace{\sum_{l \in \mathcal{N}(v_d)} [\frac{\pi^{(k+1)}(v_l)}{n_l} \cdot \frac{\lambda_l}{n_l - 1}]}_{Y} - \underbrace{\pi^{(k)}(v_d) \frac{\lambda_d}{n_d}}_{Z} \tag{9}$$

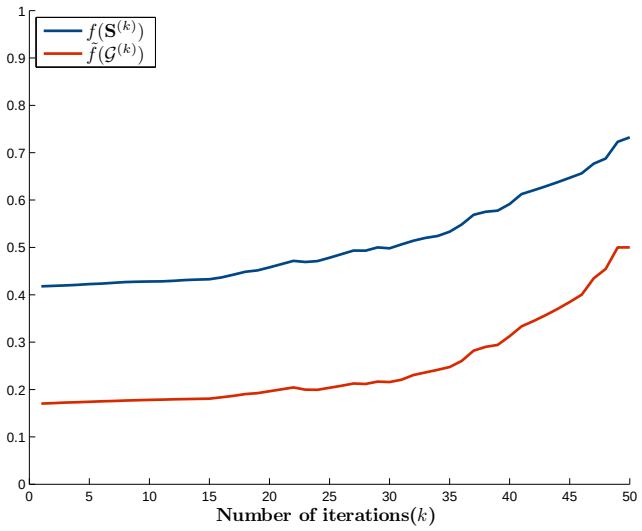

Figure S2: Testing the validity of the conjuncture that iterative network reduction scheme will increase the expected accuracy for an example case of querying TFIIH complex of *S. cerevisiae* in the PPI network of *H. sapiens*.

where, $\lambda_l = \sum_{i=1}^{N_\mathcal{Q}} \mathbf{1}\{v_l \in \mathcal{N}(v_{i^*})\}$ is the number of neighbors of v_l in the core region of $\mathcal{G}^{(k-1)}$, and λ_l is the numbers of neighbors of the deleted node v_d in that region. In practice, we have $X \geq Z$ in (9), which is mainly true since v_d is chosen to have small correspondence to the query network which typically leads to small $\pi^{(k)}(v_d)$. Also, we know that $\sum_{l=1}^{|\mathcal{V}^{(k)}|} \pi^{(k+1)}(v_l) = 1$, while $\sum_{l=1}^{|\mathcal{V}^{(k)}|} \pi^{(k)}(v_l) = 1 - \pi^{(k)}(v_d) - \sum_{v_{d'} \in \mathcal{V}_\mathcal{O}} \pi^{(k)}(v_{d'})$, where $\mathcal{V}_\mathcal{O}$ is the set of isolated nodes. This assures $\sum_{l=1}^{|\mathcal{V}^{(k)}|} [\pi^{(k+1)}(v_l) - \pi^{(k)}(v_l)] \geq \pi^{(k)}(v_d)$. Besides, since in general case we have $\mathbf{E}[\frac{\lambda_l}{n_l}] \geq \frac{\lambda_d}{n_d}$, that suggests the deleted node has less proportion of its neighbors in the core region, $X \geq Z$ remains a valid assumption in most of the iterations . Thus, considering the non-negativity of the second term (Y) in (9), $X \geq Z$ will lead to $\tilde{D}^{(k+1)} \geq 0$, as desired.

To verify the validity of the assumptions made in this section, we test the proposed scheme on querying the TFIIH complex of *S. cerevisiae* in the PPI network of *H. sapiens*. Fig. S2 demonstrates the increasing trend of $f(\mathbf{S}^{(k)})$ and $\tilde{f}(\mathcal{G}^{(k)})$ which is in consistency with the considered conjectures. We also verified that in this query example and other test cases, in more than 95% of the iterations the assumption $X \geq Z$ holds true.

S2 Comparison with the preliminary results presented in [2]

The initial results of our network querying method, which were very preliminary at the time, has been presented at ICASSP 2011 (International Conference on Acoustics, Speech, and Signal Processing) [2]. The main goal of this 4-page conference paper was simply to present our novel idea for network querying (based on network reduction approach) rather than proposing a full-fledged network querying algorithm. RESQUE, presented in this paper, fully develops and implements the initial idea presented in [2], with optimized performance, many additional technical advances and additions, comprehensive and detailed performance analyses that include thorough validation and extensive comparison against existing algorithms.

Important differences between RESQUE and the preliminary algorithm in [2] include the following: (i) RESQUE is mathematically grounded based on a probabilistic modeling of the network querying problem, while in the ICASSP paper we didn't have any analytical validation; (ii) RESQUE utilizes a prefiltering step (Section 2.2) that considerably speeds up the algorithm and reduces the chance of false positive errors; (iii) in the current paper the proposed iterative reduction approach is supported by analytical validation (Section 2.2 and the supplementary data); (iv) in the ICASSP paper, we just included the results based on the maximum-weighted matching (similar to RESQUE-M), while in this work, we also propose RESQUE-C that reports the largest connected component in the reduced target network (which is shown to clearly outperform all the other algorithms); (v) in the current paper, we conduct extensive performance analyses by querying real PPI networks using nearly 1,200 complexes and pathways as well as synthetic examples (generate using a new biologically realistic scheme), and also include several querying examples, to demonstrate the effectiveness of the proposed scheme; (vi) in this work we compare RESQUE-C and RESQUE-M against the current state-of-the-art (Torque and IsoRankN); (vii) the current paper includes both analytical and experimental results on the algorithmic complexity.

Figs. S3 and S4 compare the performance and computational time of the algorithm presented in ICASSP [2] against RESQUE-C and RESQUE-M, proposed in this paper. As we can see in Fig. S3, RESQUE-C and RESQUE-M clearly outperform the one presented in [2], in terms of meaningful hits, functional coherence, specificity, as well as computational efficiency, due to the further improvements made in RESQUE.

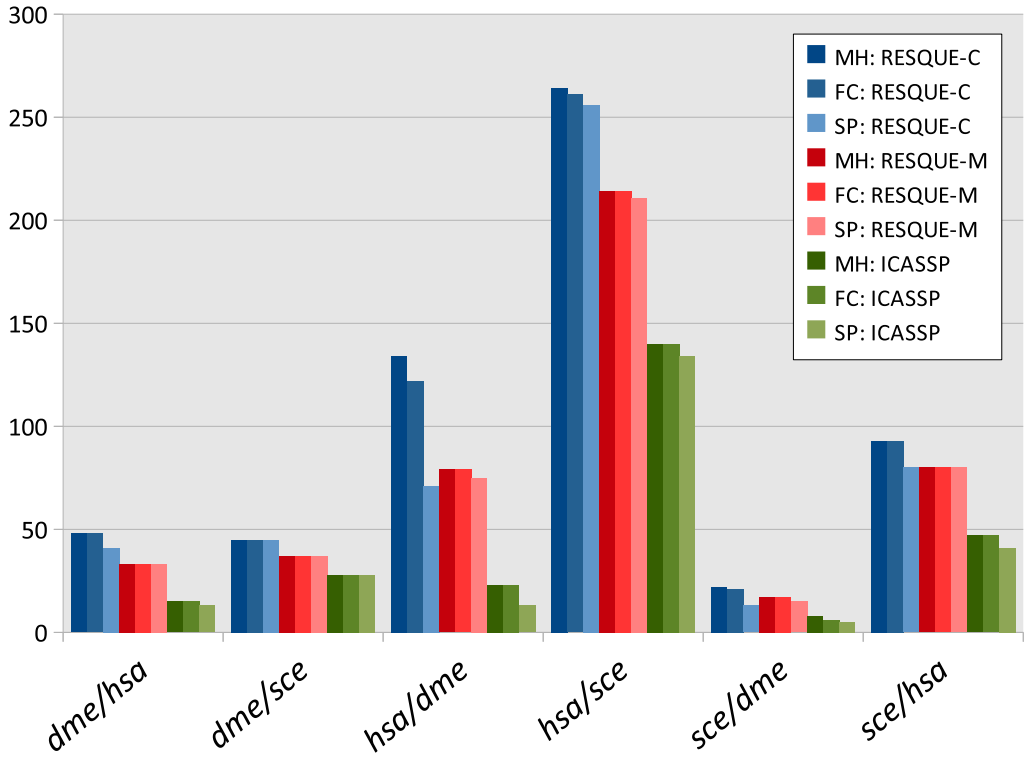

Figure S3: Performance of RESQUE-C, RESQUE-M, and the the preliminary results presented in ICASSP [2] for six pairs of query/target. For every query/target pair, we report the following metrics for each algorithm: the number of meaningful hits (MH), the number of functionally coherent hits (FC), and the number of specific hits with significant overlap to a known complex (SP). (dme: *D. melanogaster*, hsa: *H. sapiens*, sce: *S. cerevisiae*).

S3 Computational Complexity of the RESQUE algorithm

For a query complex of size $N_{\mathcal{Q}}$, and the target network with N nodes and m edges, different steps of RESQUE has the following computational complexity:

- **Semi-Markov random walk** Due to the decoupling property of the proposed random walk on the two graphs, the steady state distribution over the product graph can be computed as the kronecker product of the steady state distributions on each graph. Computing the steady state distributions is $O(m_{\mathcal{Q}})$ for the querying and $O(m)$ for the target networks. The kronecker product has the complexity of $O(z)$, where z is the number of non-zero homologues across the two networks (that is the number of the non-zero elements of the homology matrix $\mathbf{H}$, where $\mathbf{H}[i,j] = h(q_i, v_j)$). The homology matrix $\mathbf{H}$ is $N \times N_{\mathcal{Q}}$, but in practice $\mathbf{H}$ is highly sparse and $z \ll N \times N_{\mathcal{Q}}$, while in the worst

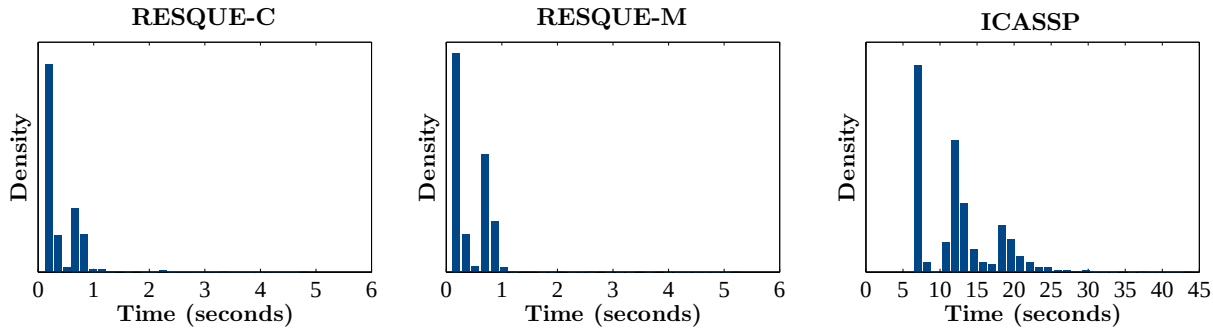

Figure S4: Comparison of the running times of RESQUE-C, RESQUE-M, and the the preliminary results presented in ICASSP [2]. Figure shows the distribution of the computational time for each algorithm over all queried complexes.

case scenario where all the pair nodes across the two networks have non-zero similarity scores, the complexity is $O(z) = O(N \times N_{\mathcal{Q}})$. So the total complexity of computing the semi-Markov corresponding scores in the matrix $\mathbf{S}$ is $O(m_{\mathcal{Q}} + m + z)$.

- **Iterative network reduction** Each iteration of the network reduction approach include the computation of semi-Markov corresponding scores and the identification of the minimal correspondence nodes. In practice we may go through $O(N)$ number of iterations since in each iteration we remove one target node. It must be noted that in the computation of semi-Markov scores (discussed in the previous item) the computation of the steady state distribution on the query network is needed to be done only once while the computation of the steady state distribution on the target network should be updated in each iteration. Besides, the identification of minimal correspondence node v_d has the complexity of $O(z + N)$ for each iteration. Thus total we need $O(m_{\mathcal{Q}} + (m + z + N)N)$ for at most N iteration of network reduction.

- The node matching step for RESQUE-C consists of application of Hungarian algorithm on two networks of size ($N_{\mathcal{Q}}$ and $2N_{\mathcal{Q}}$) whose complexity is cubic $O(N_{\mathcal{Q}}^3)$. For RESQUE-M which we apply the greedy matching we have quadratic complexity of $O(N_{\mathcal{Q}}^2 \log(N_{\mathcal{Q}}))$.

So the total complexity is $O(m_{\mathcal{Q}} + (m + z + N)N + N_{\mathcal{Q}}^3)$ for RESQUE-C and $O(m_{\mathcal{Q}} + (m + z + N)N + N_{\mathcal{Q}}^2 \log(N_{\mathcal{Q}}))$ for RESQUE-M. In practice $O(N_{\mathcal{Q}}) \ll O(N)$, $O(m_{\mathcal{Q}}) \leq O(N_{\mathcal{Q}}^2 \log(N_{\mathcal{Q}})) < O(N_{\mathcal{Q}}^3)$, and $O(m) > O(N)$, thus the complexity can be shown as $O(mN + zN + N_{\mathcal{Q}}^3)$ for RESQUE-C and $O(mN + zN + N_{\mathcal{Q}}^2 \log(N_{\mathcal{Q}}))$ for RESQUE-M.

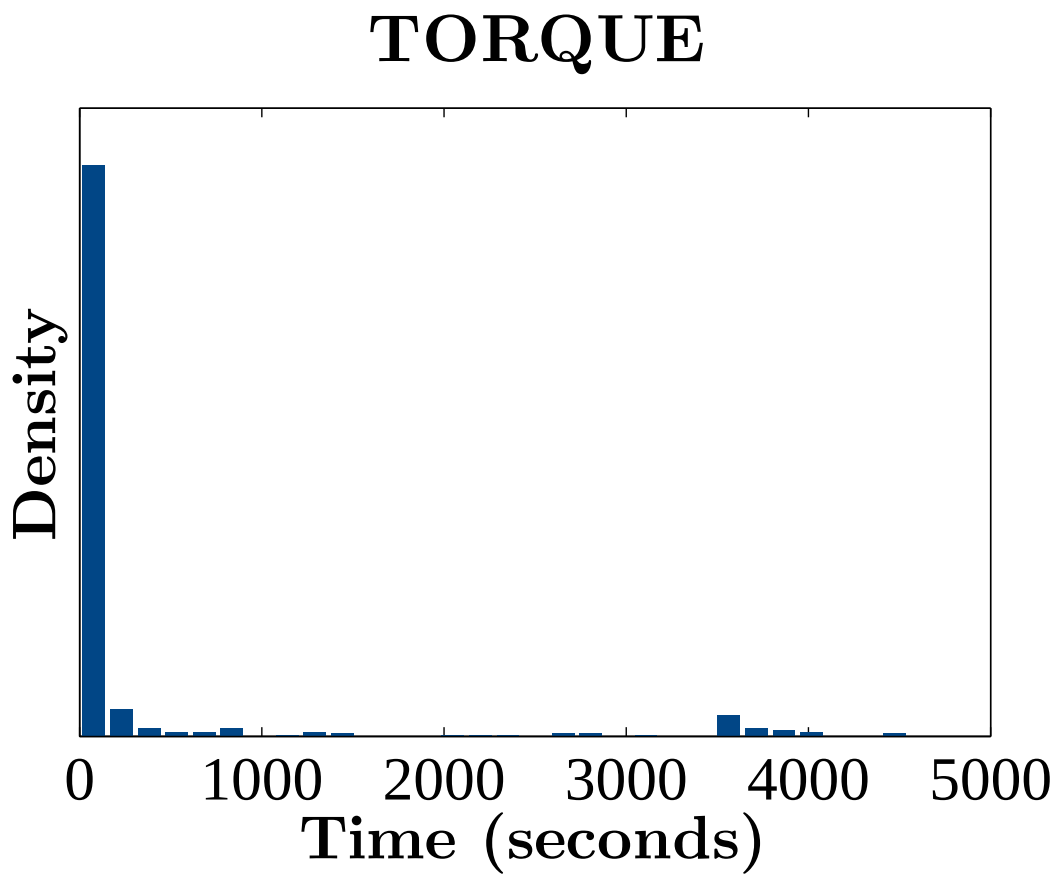

Figure S5: Running times of Torque. Figure shows the distribution of the computational time over all queried complexes.

Table S1: Novel Detections for Querying *D. melanogaster* complexes in *H. sapiens*

	Query Complex in *D. melanogaster*	**Detected Nodes in *H. sapiens***
1	DMAP1, Eaf6, CG17691, CG8199	PDHA1, PDHA2, PDHB
2	CG13349, CG13779, CG31742, CG8858, Mov34, Pomp, Pros25, Pros26, Pros29, Pros35, Pros45, Pros54, Prosbeta3, ProsMA5, REG, Rpn6, Rpt3R, StIP, TER94, CG6255, Scsalpha, Sucb, dik, Rpb4	VCP, PSMC1, PSMC3, PSMC2, PSMC4, ADRM1, PSMC5, PSMD4, PSMD7, PSMB7, PSMA1, PSMB5, PSMA4, PSMA3, PSMA7, PSMA6, PSMB1, PSMA5, PSMB6, PSMA2, PSMB3, PSMB2, POMP
3	Mhc, Mlc1, Mlc2, CG1746, CG3321, CG4692	MYO7A, MYO9B, MYO10, MYO6, CALM1

4	CG7766, PhKgamma, CG3397, eag, elk, Hk, sei, Sh, Shab, Shal, Shaw, CG12304, CG8235, CG4050, CG5038	KCNH1, KCNH5, KCNB2, KCNB1, KCNV2, KCNG4, KCNG1, KCNC1, KCNV1, KCNF1, KCNS3, KCNG2, KCNG3
5	exo70, exo84, sec10, sec15, sec3, sec5, sec6, sec8, CG16940, Csl4, Dis3, Mtr3, Rrp4, Rrp45, Rrp46, Ski6, bic, CG4415, Nacalpha, Atac1, Hcf, Pcaf, Rpb4, g	CREBBP, EP300, EXOC4, AP1B1, EXOC2, KAT2A, KAT2B, AP1G1, EXOC3, EXOC7, EXOC8, TADA2A
6	CkIIalpha, CkIIbeta, CkIIbeta2, Ste12DOR	RPS6KB1, MAPK8, MAPK3, MAPK1, MAPK14, CSNK2A2
7	CG11552, CG5604, CG7257, CG9588, Mov34, Pros45, Pros54, Rpn1, Rpn11, Rpn12, Rpn2, Rpn5, Rpn6, Rpn7, Rpn9, Rpt1, Rpt3, Rpt4	UBE3C, PSMD2, VCP, PSMC1, PSMC3, PSMC2, PSMC4, PSMC5, PSMD7, PSMD10
8	Bap60, brm, Caf1, dalao, dom, Iswi, mor, Mov34, Rpn11, Rpn12, Rpn5, Rpn6, Rpn7, Rpn9, CG11909, CG14476, CG33080, CG6453, CG7685	CHD3, SMARCA4, SMARCA2, SMARCC2, SMARCC1, SMARCA5, SMARCD3, RBBP7, RBBP4, SMARCE1
9	FKBP59, Galpha49B, inaC, inaD, ninaC, ninaE, norpA, trp, trpl, Oscp, CG11340, CG12344, CG6927, pHCl	PLCB1, PLCB2, TRPC4, TRPC5, TRPC6, TRPC7, TRPC3, TRPC1, PRKCE, PRKCG, ADRBK1, PRKCD, PRKCB, PRKCA, PRKCZ, GABRG2, GABRR1, GABRR2, GABRA1, GNA13, OPRD1, GNAQ, PRKACA

Table S2: Novel Detections for Querying *D. melanogaster* complexes in *S. cerevisiae*

	Query Complex in *D. melanogaster*	**Detected Nodes in *S. cerevisiae***
1	CG15831, CG17237, CG31638, CG34435, ck, cnn, d, didum, jar, Mhc, Mhcl, Mlc1, Mlc2, Myo28B1, Myo31DF, Myo61F, Myo95E, ninaC, zip, HisCl1, ort	MYO1, MLP1, MLP2, MYO2, BCK1, MYO4, SMC4, EDE1, SSK22, RAD50, MYO3, SMC3, SMC1, MYO5, SMC2, SPC110, IMH1, SLK19, YGR130C, ADY3, MAD1, STE11, SGM1, BRE1, TID3, COY1, PBS2, SPO21, CNM67, STE7, RUD3, MLC1, CMD1
2	CG7766, PhKgamma, CG3397, eag, elk, Hk, sei, Sh, Shab, Shal, Shaw, CG12304, CG8235, CG4050, CG5038	BCK1, KIN2, SAK1, KIC1, KSP1, CDC15, STE20, RAD53, CBK1, STE11, CDC5, SNF1, DBF2, DUN1, RCK1, MKK1, SLT2, HOG1, IPL1, FUS3, CDC28
3	Brf, Caf1, CG13176, Dref, ey, eyg, gsb, Iswi, pan, prd, Ptx1, TfIIB, toe, tral, Trf, Trf2	MOT1, CHD1, ISW1, ISW2, TAF5, BRF1, SUA7, SPT15
4	Cbp20, Cbp80, Dp, E2f, Rbf, Rbf2	STO1, PAB1, PUB1, SGN1, CBC2
5	CG1193, CG17293, kat80, Sur, CG12027, CG31477, CG7211, CG7813	RPT1, RPT4, RPT2, RPT5, RPT3, RPT6
6	Acf1, asf1, Caf1, Pi3K21B	SPT7, TAF5, GCN5
7	Mes4, Pole2, RnrL, RnrS	RNR1, RNR3, RNR2, RNR4
8	Bap60, brm, Caf1, dalao, dom, Iswi, mor, Mov34, Rpn11, Rpn12, Rpn5, Rpn6, Rpn7, Rpn9, CG11909, CG14476, CG33080, CG6453, CG7685	MOT1, SNF2, CHD1, STH1, ISW1, ISW2, RSC1, RSC2, SWI3, RSC4, SNF12, RSC8, NHP6B, NHP6A
9	FKBP59, Galpha49B, inaC, inaD, ninaC, ninaE, norpA, trp, trpl, Oscp, CG11340, CG12344, CG6927, pHCl	BCK1, MYO3, MYO5, PKC1, PKH1, YPK2, MKK1, TPK1, TPK3, TPK2, CPR6

Table S3: Novel Detections for Querying *H. sapiens* complexes in *D. melanogaster*

	Query Complex in *H. sapiens*	**Detected Nodes in *D. melanogaster***
1	CCNT1, BRD4, CDK9, MED1	Doa, CG7028, KP78b
2	ESR1, SRC, PIK3R1, BCAR1	CG11870, RhoGAP92B
3	CHEK2, NASP, HIST1H3F, CHAF1B, ASF1B, ASF1A	Doa, KP78b, lok
4	KDM1A, CHD3, MTA1, HDAC1, CHD4, HDAC2	E(bx), Acf1, Iswi
5	MLH1, BLM, TOP3A, RMI1	CG6227, Pms2, Mlh1
6	TCEB2, CUL5, ASB6, RNF7	mask, CG9121
7	CCNT1, POLR2A, NOTCH1, CDK8, CDK7, CDK9, RBPJ, EP300, MED1, SUPT6H, MAML1, INPP5K, SUPT16H	CG32703, SAK, MED15, Pak, ald, Eip63E, Cdk8, CycK, CG7236, Cdk4, cdc2c, Cdk5, CycC
8	ERCC2, ERCC3, GTF2H1, CDK7, CCNH, MNAT1, GTF2H3, GTF2H5	Doa, CG7028, KP78b, lok
9	ITGB3, ITGA2B, SRC, CD47	Ack, dock
10	BACH1, BRCA1, MLH1, MSH6, TOPBP1, BARD1	N, CG4393, mib1
11	CDK1, PARP1, POLA1, TOP1, CCNB1, RPA2, CCNA2, RPA1, POLD1, RPA3, RFC2, RFC1, POLE	SAK, hpo, CycD, CycJ, Cdk4, cdc2c, Cdk5
12	ARNT, AHR, RBX1, CUL4B	sima, trh, sim, tgo
13	ADRB1, AKAP5, PRKACG, PRKACB	Doa, KP78b, lok
14	NCAPH, NCAPD2, NCAPG, SMC4	rad50, Khc
15	CCNT1, NCL, POLR2A, CDK9, CPSF2, PPARGC1A	SAK, CycK, CG7236, Cdk4, cdc2c
16	RBPJ, CTBP1, SPEN, RBBP8	CtBP, Hrb27C
17	MED14, MED6, CCNC, CDK8, MED21, MED10, MED23	Eip63E, Cdk8, Cdk4, cdc2c, Cdk5, CycC
18	INSR, PIK3R1, IRS1, KHDRBS1	CG11870, RhoGAP92B
19	CDH5, PARD3, PARD6G, PARD6A	skf, veli
20	RAD21, SMC1A, STAG2, SMC3	CG6905, Klp68D
21	IKBKB, TROVE2, NCOA2, CREBBP, IKBKG, NCOA3	sima, trh, sim, tgo

22	CBX4, ZEB2, KDM1A, ZNF217, ZEB1, CTBP1, HDAC1, ZNF516, RREB1, HDAC2, LCOR, RCOR3, RCOR1, CDYL	CG8301, CG12391, wek, CG3032, Neu2, CG9331, CG15436, drm
23	TBL1X, SKP1, SIAH1, CACYBP	ago, slmb, wds, skpA, skpB
24	NBN, XRCC5, LIG4, MRE11A, XRCC4, RAD50	rad50, Khc, mre11, CG32137
25	AXIN1, CTNNB1, GSK3B, PPP2R5A	Doa, CG7028, KP78b, lok
26	BRCA1, UIMC1, RBBP8, BARD1	N, CG4393, mib1
27	POLA1, CCNA2, CDK2, RFC4, RFC2, RFC1, RFC5, RFC3, POLA2	SAK, hpo, CycD, Cdk4, cdc2c, Cdk5
28	TCEB2, CUL5, ASB2, RNF7	N, CG4393, mib1
29	SEPT2, SEPT7, SEPT8, SEPT11, SEPT9	Sep4, Sep2, Sep1
30	NKX3-2, NKX3-1, TLE1, TLE2, TLE3, TLE4, TLE6	gro, inv, unc-4, H2.0, lbl
31	MTA1, RBBP7, HDAC1, HDAC2	ago, wds
32	MED7, MED14, MED24, MED6, CCNC, CDK8, MED21, MED1, MED27, MED12, MED10, MED20, MED17, MED13, THRAP3, MED16, MED31	SAK, Pak, ald, Eip63E, Cdk8, CycK, CG7236, CycH, Cdk4, cdc2c, Cdk5, CycC
33	ERCC3, CCNC, POLR2A, GTF2F1, CDK8, SMARCA2, SMARCA4, GTF2B, SMARCB1, MED21, GTF2H3, CREBBP, KAT2B	SAK, Pak, Eip63E, Cdk8, CycK, CG7236, CycH, Cdk4, cdc2c, Cdk5, CycC
34	ITGB3, ITGAV, PXN, PTK2B	Lmpt, tup, Lim3
35	MED7, MED14, MED24, MED6, CCNC, CDK8, MED21, MED1, MED25, MED12, MED15, MED17, MED13, MED23	CG32703, SAK, MED15, Pak, ald, Eip63E, Cdk8, CycK, CG7236, Cdk4, cdc2c, Cdk5, CycC
36	PARP1, POLA1, TOP1, RPA2, LIG1, CCNA2, CDK2, RPA1, POLD1, RPA3, RFC2, RFC1, POLE	SAK, hpo, CycD, CycJ, Cdk4, cdc2c, Cdk5
37	APEX1, STAT3, HIF1A, CITED4, CITED1, CITED2	sima, trh, sim, tgo
38	RAB5A, EEA1, VPS11, VPS18	CG10703, cbs, Rab6
39	BRCA1, LMO4, LDB1, RBBP8	Chi, tup, Lim3, Lim1, ap, CG5708
40	RBX1, TCEB2, CUL5, WSB1	ago, wds

41	HNRPDL, DHX15, DDX5, HNRNPH1, FUS, HNRNPM, HNRNPU, EWSR1, ILF2, ILF3, TARDBP, DGCR8, DDX1, TAF15, DDX17, SRPK1, HNRNPUL1, RNASEN, RALY	Doa, RecQ4, CG7922, CG30122, pea, CG7879, fus, CG7028, CG6418, Rm62, CG3225, rump, ald, CG11266, glo, Rox8, CG3162, Rbp4, bsk, CG32169, lark, Spx, Cdk4, cdc2c, png, SF2, SC35
42	TCEB2, ASB12, CUL5, RNF7	N, CG4393, mib1
43	EP300, CREBBP, KAT2B, NCOA3	sima, trh, sim, tgo
44	COPS2, RBX1, GPS1, CUL4A, COPS6, DDB2, COPS5, COPS8, COPS4, COPS3, COPS7B, COPS7A	alien, Rpn6, CSN4, CSN6, CSN5, CSN7
45	MED7, MED14, MED24, MED6, MED26, CCNC, CDK8, MED21, MED1, MED25, MED12, MED15, MED17, MED13, MED23	CG32703, SAK, MED15, Pak, ald, Eip63E, grp, Cdk8, CycK, CG7236, Cdk4, cdc2c, Cdk5, CycC
46	TCEB2, CUL5, ASB7, RNF7	N, CG4393, mib1
47	RAD21, SMC1A, STAG1, SMC3	CG6905, Klp68D
48	HSPA8, SKP1, RYBP, KDM2B, RNF2, PCGF1	CG4221, CG9772, ppa, CG2010, skpA, skpB, skpC
49	RFC3, CHTF8, CHTF18, DSCC1	CG8142, RfC3, RfC40
50	XRCC5, MRE11A, TERF2, RAD50, TINF2, TERF2IP	rad50, Khc, mre11, CG32137
51	NBN, MRE11A, ATM, MDC1, RAD50, FANCD2	rad50, Khc, mre11
52	BIRC5, CDCA8, AURKB, INCENP	SAK, Cdk4, cdc2c
53	C19orf2, POLR2E, SKP1, SKP2, CUL1, PFDN2, STAP1, RUVBL2, RUVBL1	CG4221, CG8272, CG9772, ppa, CG2010, skpA, skpB, skpC
54	NBN, RFC4, RFC2, RFC1, BRCA1, MLH1, MSH2, MRE11A, MSH6, BLM, ATM, RAD50	alpha-Spec, didum, osp, CG6227, Klp3A, Bsg25D, asl, Pms2, Mlh1
55	ESR1, BCL3, RELA, NCOA3	N, CG4393, mib1
56	MTA2, MBD3, HDAC1, CHD4, RBBP7, HDAC2	E(bx), Acf1, Iswi

Table S4: Novel Detections for Querying *H. sapiens* complexes in *S. cerevisiae*

	Query Complex in *H. sapiens*	**Detected Nodes in *S. cerevisiae***
1	ESR1, SRC, PIK3R1, BCAR1	YSC84, LSB3
2	RAD21, SMARCA5, MTA2, MBD3, CHD3, MTA1, HDAC1, SMC1A, RBBP7, STAG2, STAG1, HDAC2, BAZ1A, MBD2, SMC3	SMC4, RAD50, SMC3, SMC1, SMC2, IRR1, SLK19, HDA1, BRE1, HDA2, PWP1, MCD1, HOS1
3	ACTL6A, TCF3, INO80D, NFRKB, INO80C, INO80E, MCRS1, INO80B, ACTR8, ACTR5, INO80, RUVBL2, RUVBL1	IRC20, SWR1, INO80, ARP4, RVB2, RVB1, ARP3, ARP6, ARP2, ACT1
4	MYST2, PHF16, MEAF6, ING4	SAS3, YNG1
5	CCNT1, POLR2A, NOTCH1, CDK8, CDK7, CDK9, RBPJ, EP300, MED1, SUPT6H, MAML1, INPP5K, SUPT16H	RPO21, GIN4, SAK1, CDC15, SGV1, SNF1, SSN3, CTK1, SLT2, SMK1, RIM11, CAK1, IPL1, FUS3, SSN8, KIN28, PHO85, CDC28
6	SIN3B, SAP30, HDAC1, RBBP7, HDAC2, SIN3A, BRMS1	SIN3, TUP1, HDA1, RPD3
7	ACTN4, SVIL, ITGB2, HSPA8, PTPRB, MPP3, LAMA3	EDE1, SPC110, SSE1, SSA1, SSA2, SSB1, CMD1
8	ADRB1, AKAP5, PRKACG, PRKACB	YPK2, SNF1, RCK1, MEK1
9	ARID1A, MTA2, MBD3, ACTL6A, HNRNPC, SMARCA4, SMARCB1, HDAC1, CHD4, SMARCC2, GATAD2B, HDAC2, DPF2, SMARCC1, SMARCD2, SMARCE1, MBD2	MOT1, SNF2, CHD1, STH1, ISW1, ISW2, RSC1, SNF5, RSC2, SWI3, RSC4, SNF12, RSC8, ARP7, SFH1, NHP6B, NHP6A
10	NCAPH, NCAPD2, NCAPG, SMC4	SMC4, SMC3, SMC1, SMC2
11	CBL, SHC1, GRB2, LCP2	SLA1, ABP1, RVS167, YSC84, HSE1, LSB3
12	MED7, MED14, MED6, MED26, CDK8, MED1, MED25, MED12, MED15, MED17, MED13, MED23, MED16	BCK1, GIN4, SAK1, PKH2, CDC15, STE20, CLA4, SGV1, SNF1, SSN3, CTK1, STE7, SLT2, SMK1, RIM11, KSS1, CAK1, IPL1, FUS3, KIN28, CDC28
13	INSR, PIK3R1, IRS1, KHDRBS1	RGA2, CLA4, CDC28
14	KDM1A, MTA2, CHD3, MTA1, HDAC1, ZMYM3, KIAA0182, CHD4, RBBP7, HDAC2, PHF21A, HMG20B, ZMYM2, RCOR1, SIN3B, SIN3A	MOT1, SIN3, CHD1, ISW1, ISW2, TUP1, HDA1, RCO1, RPD3
15	IKBKB, TROVE2, NCOA2, CREBBP, IKBKG, NCOA3	TPK1, TPK3, TPK2

16	MYST2, PHF17, MEAF6, ING4	SAS3, YNG1
17	ITGB3, ITGA2B, PTK2, CD47	CDC15, STE20, IPL1
18	MLL2, KDM6A, WDR5, NCOA6, RBBP5, PAXIP1, N4BP2, C13orf23, ASH2L, ZNF281	SET1, PFS2, SWD1, SWD2, SWD3
19	MYST2, MEAF6, PHF15, ING4	SAS3, YNG1
20	AXIN1, CTNNB1, GSK3B, PPP2R5A	GIN4, RTS1, RIM11, CDC28
21	NPM1, PARP1, HSPA5, H2AFX, CALR, DHX30	SSE1, SSA1, SSA2, SSB1
22	PARP1, XRCC1, NCAPH, NCAPD2, NCAPG, SMC4	SMC4, SMC3, SMC1, SMC2, SLK19
23	PLCG1, PIK3R1, GRB2, SOS1	SLA1, ABP1, RVS167, YSC84, HSE1, LSB3
24	TROVE2, NCOA2, NCOA1, CREBBP	SPT7, GCN5
25	MTA1, RBBP7, HDAC1, HDAC2	PWP2, UTP13, SOF1
26	MED7, MED14, MED24, MED6, CCNC, CDK8, MED21, MED1, MED27, MED12, MED10, MED20, MED17, MED13, THRAP3, MED16, MED31	BCK1, GIN4, SAK1, PKH2, CDC15, STE20, CLA4, KNS1, SGV1, SNF1, SSN3, CTK1, STE7, SLT2, CCL1, SMK1, RIM11, KSS1, CAK1, IPL1, FUS3, SSN8, KIN28, PHO85, CDC28
27	ITGB3, ITGAV, PXN, PTK2B	CDC15, STE20, IPL1
28	ITGB1, SLC7A8, SLC3A2, SLC7A5	LYP1, MUP1
29	HNRNPR, HSPA5, DDX5, NCL, HNRNPH1, FUS, HNRNPU, DHX9, ILF3, DGCR8, DDX17	PRP22, MAK5, HCA4, DRS1, DBP7, NOP4, SSC1, DBP1, DED1, PAB1, DHH1, HAS1, PUB1, DBP8, NOP13, SGN1
30	MED14, CCNT1, BRD4, MED24, CDK9, MED1, MED12, MED17	SNF1, SSN3, CTK1, FUS3, SSN8, KIN28, PHO85, CDC28
31	SMARCA4, SMAD3, SMAD4, SMAD2, ARID1B, SMARCC2, CREBBP, SMARCC1, TRIM33, NCOA3	MOT1, SNF2, CHD1, STH1, SWI1, ISW1, ISW2, RSC1, RSC2, SWI3, RSC4, RSC8
32	NDC80, SPC24, NUF2, SPC25	SMC1, SMC2, SPC110, SLK19, KAR3, TID3, NUF2
33	POLB, XRCC1, LIG3, PNKP	DNL4, POL4
34	HNRPDL, DHX15, DDX5, HNRNPH1, FUS, HNRNPM, HNRNPU, EWSR1, ILF2, ILF3, TARDBP, DGCR8, DDX1, TAF15, DDX17, SRPK1, HNRNPUL1, RNASEN, RALY	PRP22, SAK1, PKH2, PRP16, PRP43, DRS1, DBP7, MSS116, SNF1, PAB1, DBP2, HRP1, CTK1, SUB2, HRB1, GBP2, NSR1, NPL3, NOP13, CDC28, RNA15, SGN1

35	THOC5, BAT1, THOC7, THOC4, THOC6, THOC2, THOC1	PWP2, MAK5, HCA4, ROK1, HAS1, DBP8
36	GNA11, TRPC1, CAV1, ITPR3	GPA1, GPA2
37	FANCA, RPA2, RPA1, RPA3, MLH1, BLM, FANCC, TOP3A, RMI1, FANCE, FANCF, FANCL	SGS1, PMS1, MLH1, MLH3, MLH2, TOP3
38	HIRA, HIST1H3F, CHAF1A, CHAF1B, ASF1B, ASF1A	HIR2, HIR1, ASF1
39	ITGB4, ITGA6, SHC1, GRB2	SLA1, ABP1, RVS167, YSC84, HSE1, LSB3
40	CASC3, EIF4A3, MAGOH, UPF3B, WIBG, RBM8A	MAK5, HCA4, DBP1, DED1, PAB1, DHH1, HAS1, PUB1, RNA15, SGN1, IST3
41	NPM1, HSPA5, H2AFX, CSDA, CALR, HIST1H4A, PTCD3, HIST1H2BM	SSE1, SSA1, SSA2, SSB1
42	NBN, RFC4, RFC2, RFC1, BRCA1, MLH1, MSH2, MRE11A, MSH6, BLM, ATM, RAD50	RAD50, SMC3, SMC1, SMC2, MSH5, MSH4, SLK19, BRE1, MRE11
43	MTA2, MBD3, HDAC1, CHD4, RBBP7, GATAD2B, HDAC2, MBD2	MOT1, CHD1, ISW1, ISW2

Table S5: Novel Detections for Querying *S. cerevisiae* complexes in *D. melanogaster*

	Query Complex in *S. cerevisiae*	**Detected Nodes in *D. melanogaster***
1	APS2, APM4, APL3, APL1	AP-1gamma, AP-1sigma
2	APM3, APL6, APS3, APL5	AP-1gamma, AP-1sigma
3	GAL83, SNF1, SIP2, SIP1, SNF4	Doa, KP78b, lok
4	TFB1, KIN28, TFB3, CCL1, SSL1, TFB4, TFB2, RAD3, SSL2	Eip63E, Cdk8, CycH, Cdk4, cdc2c, Cdk5, CycC
5	SMC3, REC8, MCD1, IRR1, SMC1	GM130, hk, nudE
6	RPN12, ESC2, RPN6, RPN11, RPN8, RPN7, RPN13, RPN9, RPN5, RPN3	alien, Rpn6, CSN4, CSN6, CSN5
7	BUD27, GIM3, GIM4, PFD1, GIM5, PAC10, YKE2	CG6719, CG15676, CG7048, l(3)01239, CG7770
8	HAP4, HAP3, HAP2, HAP5	NC2alpha, NC2beta, CG10447, Mes4
9	CDC34, SAF1, UFO1, DIA2, HRT3, SKP1, CDC53, HRT1, DAS1, MET30, CDC4, GRR1	ago, CG9461, CG4221, CG8272, ppa, slmb, morgue, wds, skpA, skpB, skpC

Table S6: Novel Detections for Querying *S. cerevisiae* complexes in *H. sapiens*

	Query Complex in *S. cerevisiae*	**Detected Nodes in *H. sapiens***
1	RHO1, FKS1, FKS3, GSC2	CDC42, RHOJ, RAC1, RAC2
2	HPC2, HIR3, HIR2, HIR1	TBL1X, CORO2A, TBL1XR1
3	RPC82, RET1, RPC34, RPB5, RPC53, RPC19, DSE2, RPC40, RPO31, RPC37, RPC31, RPB8, RPC11, RPC17, RPO26, RPC25, RPB10	POLR2A, POLR2B, TCEA1, POLR2C, POLR2E, POLR2J, POLR2H, POLR2F, POLR2I, POLR2L
4	RRP7, UTP22, IFH1, CKB2, FHL1	FOXO3, FOXO1, FOXG1, FOXH1
5	BRR2, SMD1, LIN1, SMX3, PRP28, SME1, SMD3, PRP18, PRP8, SMB1, AAR2, SNU114, SMD2, DIB1	DDX20, SNRPB, SNRPN, SNRPD3, SNRPD1, SNRPD2, LSM8, SNRPF, LSM6
6	RPA14, RPA43, RPB5, RPA12, RPC19, DSE2, RPA34, RPC40, RPB8, RPA49, RPO26, RPA190, RPA135, RPB10	POLR2A, POLR2B, TCEA1, POLR2C, POLR2E, POLR2J, POLR2H, POLR2F, POLR2L
7	PCL7, PCL10, CDC28, PCL6, PCL9, PCL1, CLN2, CKS1, PCL2, CLG1, PHO80, PCL8, PHO85, PCL5	RET, RPS6KA2, RPS6KB1, PAK2, PCTK1, MAPK8, MAP2K1, MAPK3, MAPK1, MAPK14, CDK7, CDK2, CDK1, CDK5, CKS2
8	UBX2, NPL4, CDC48, SSM4, CUE1	VCP, PSMC1, PSMC3, PSMC2, PSMC4, PSMC5
9	RPN10, RPN1, RPN12, ESC2, RPT4, RPT3, RPN6, RPT6, RPN11, RPN8, RPN7, RPN13, RPN9, RPN2, RPN4, RPT2, UBP6, RPT1, NAS6, RPT5, RPN5, RPN3	NFKB1, PSMD2, VCP, BARD1, GPS1, COPS2, BCL3, PSMC1, PSMC3, PSMC2, COPS3, PSMD11, PSMC4, COPS4, PSMC5, COPS5, COPS6, PSMD7, NFKBIA, PSMD10, CDKN2C, MTPN
10	JNM1, ARP10, NIP100, ARP1	ACTR3, ACTR2, ACTC1
11	CKB2, CKB1, CKA2, CKA1	CSNK2A1, CDK7, CDK6, CDK2, CDK1
12	RSC58, NPL6, RSC6, LDB7, ARP9, RSC2, ARP7, RTT102, RSC3, RSC9, STH1, RSC1, RSC4, SFH1, RSC8	SRCAP, CREBBP, EP300, BRWD1, SMARCA4, SMARCA2, SMARCC2, SMARCC1, KAT2A, KAT2B, ACTL6A, ACTL6B, SMARCB1
13	CDC34, SAF1, UFO1, DIA2, HRT3, SKP1, CDC53, HRT1, DAS1, MET30, CDC4, GRR1	CUL4B, CUL5, CUL1, CUL3, CUL2, FBXW7, TRAF7, CUL4A, BTRC, FBXO9, FBXW11, FBXW2, FBXL2, CDC34, SKP1, UBE2D2, UBE2D1, RNF7, RBX1
14	ABF1, RAD16, RAD7, ELC1	CHD3, SMARCA5

References

[1] Kuhn HW (1955) The Hungarian method for the assignment problem. Naval Research Logistic Quarterly 2: 83–97.

[2] Sahraeian S, Yoon BJ (2011) Fast network querying algorithm for searching large-scale biological networks. In: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. pp. 6008 -6011.