Role of network topology based methods in discovering novel disease-gene associations

Emre Güney

Structural Bioinformatics Laboratory
GRIB, Pompeu Fabra University (UPF)
Barcelona, Catalunya, Spain

April 10, 2013

Outline

No Two Humans Are Identical

image from twinsrealm.com

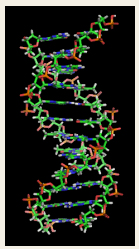
No Two Humans Are Identical

image from twinsrealm.com

- variations in genetic sequence
- environmental factors
- epigenetic differences

From Genotype to Phenotype

Genotype



Inheritable information coded within an organism

From Genotype to Phenotype

Genotype

Inheritable information coded within an organism

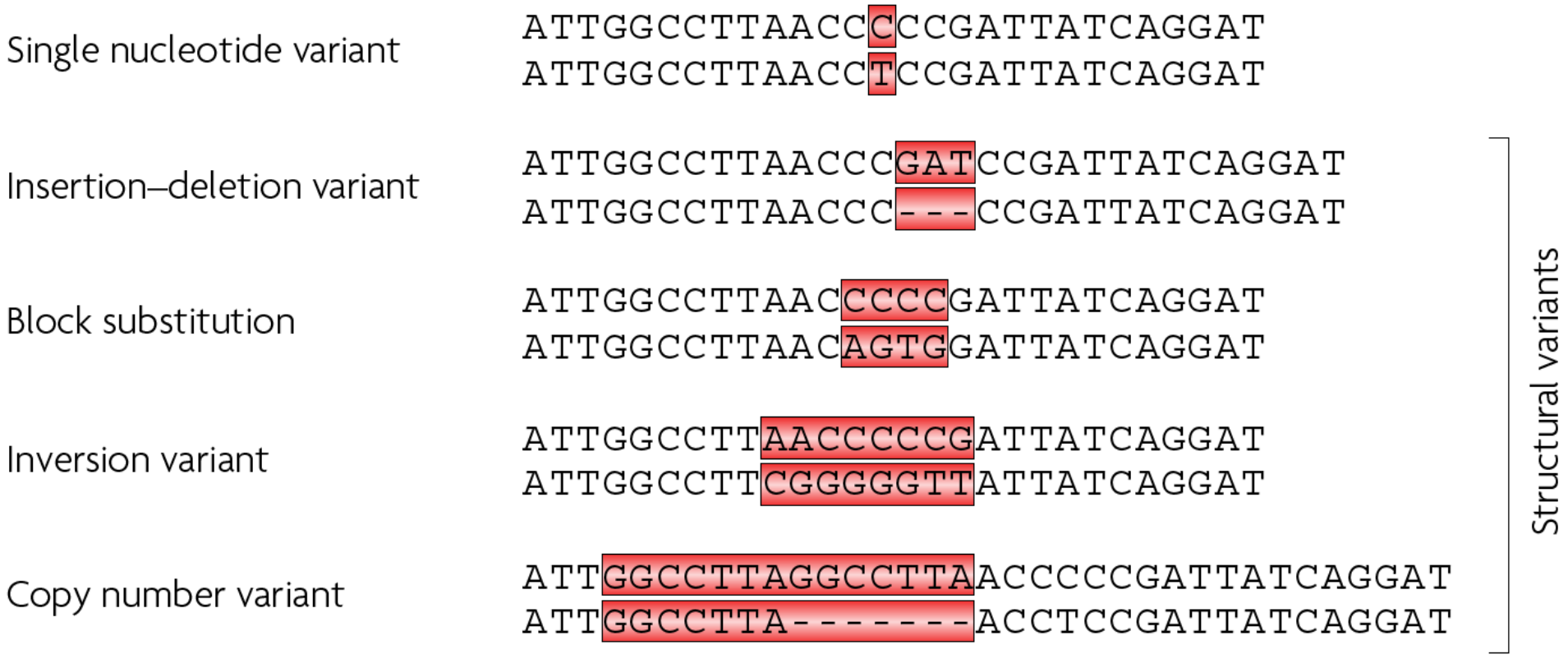
$\Downarrow$

Phenotype

The physical manifestation of an organism

images from wikipedia.com

Variations in Human Genome



Frazer et al., 2009

Human Genetic Variants and Diseases

- cancer
- autism
- diabetes
- asthma
- obesity
- Crohn's disease
- Alzheimer's Disease
- schizophrenia
- Parkinson disease
- haemophilia
- Hunter syndrome
- muscular dystrophy

Shastry, 2002, Hirschhorn et al., 2002, Feuk et al., 2006, Eichler et al., 2007, Jr and Scherer, 2008, Stankiewicz and Lupski, 2010

Disease-genes

- share structural and sequential features *(Furney et al., 2008)*

Disease-genes

- share structural and sequential features *(Furney et al., 2008)*
- coexpressed in the tissues related to the pathology *(Goh et al., 2007)*

Disease-genes

- share structural and sequential features *(Furney et al., 2008)*
- coexpressed in the tissues related to the pathology *(Goh et al., 2007)*
- highly inter-connected with each other and non-randomly positioned in the interaction network *(Wachi et al., 2005, Jonsson and Bates, 2006, Goh et al., 2007, Kar et al., 2009)*
  - non-essential genetic disease genes are on the periphery
  - essential genes and disease-genes with somatic mutations are central

Disease-genes

- share structural and sequential features *(Furney et al., 2008)*
- coexpressed in the tissues related to the pathology *(Goh et al., 2007)*
- highly inter-connected with each other and non-randomly positioned in the interaction network *(Wachi et al., 2005, Jonsson and Bates, 2006, Goh et al., 2007, Kar et al., 2009)*
  - non-essential genetic disease genes are on the periphery
  - essential genes and disease-genes with somatic mutations are central
- show dysregulation patterns in the regulatory network *(Ergun et al., 2007, Guo et al., 2007, Mani et al., 2008)*

Disease-genes

- 10% of human genes are estimated to contribute to disease phenotypes *(Strausberg et al., 2003, Amberger et al., 2009)*

Disease-genes

- 10% of human genes are estimated to contribute to disease phenotypes *(Strausberg et al., 2003, Amberger et al., 2009)*
- many interrelated pathways are implicated in complex diseases *(Wood et al., 2007, Barabasi et al., 2011)*

Disease-genes

- 10% of human genes are estimated to contribute to disease phenotypes *(Strausberg et al., 2003, Amberger et al., 2009)*
- many interrelated pathways are implicated in complex diseases *(Wood et al., 2007, Barabasi et al., 2011)*
- incorporating relationships between genes at various levels is indispensable to characterize such pathways *(Rhodes and Chinnaiyan, 2005, Joyce and Palsson, 2006)*

From Parts to Whole: Systems Biology

Biomolecule (Gene, RNA, protein, ...)

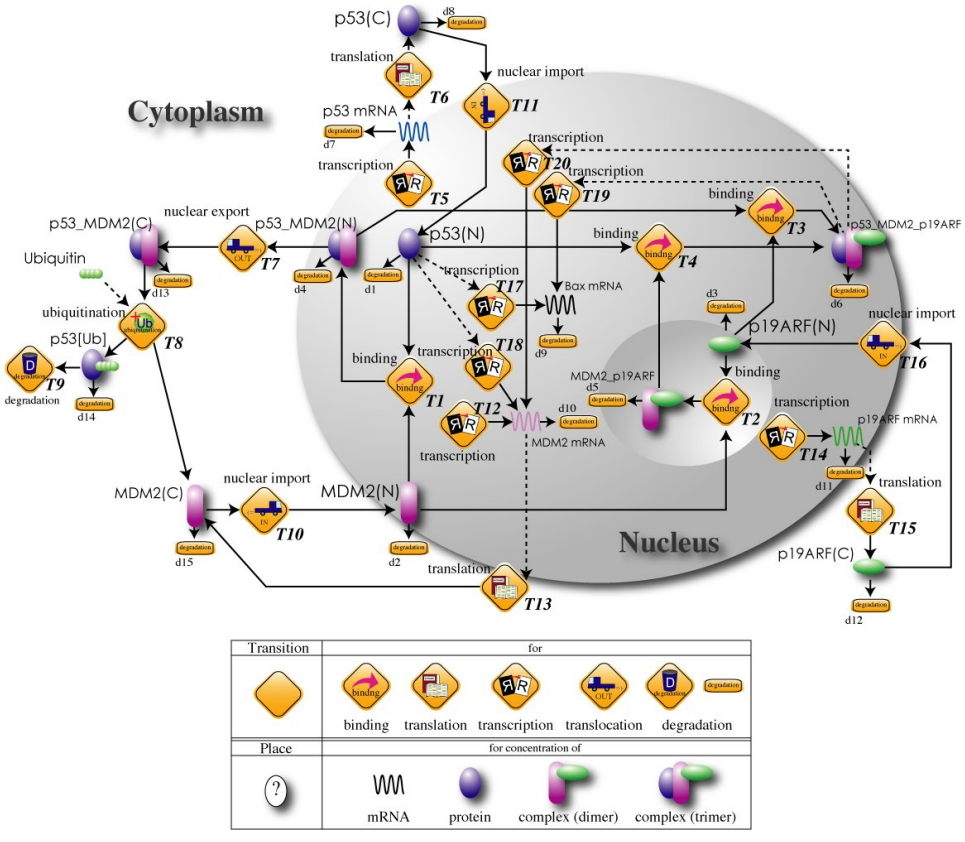
$\Rightarrow$

System consisting of biomolecules

Systems biology studies **interactions between the components** of the biological system within the framework of the biological system as a whole

images from cs.haifa.ac.il/~oren and greenpeace.org

From Parts to Whole: Systems Biology



Transition	for
	binding, translation, transcription, translocation, degradation
Place	for concentration of
?	mRNA, protein, complex (dimer), complex (trimer)

- metabolic
- regulatory
- genetic
- protein-protein

Network biology is the study of these **coupled biological networks** giving rise to the behaviour of the cell *(Barabasi and Oltvai, 2004)*

Protein-protein Interaction Networks

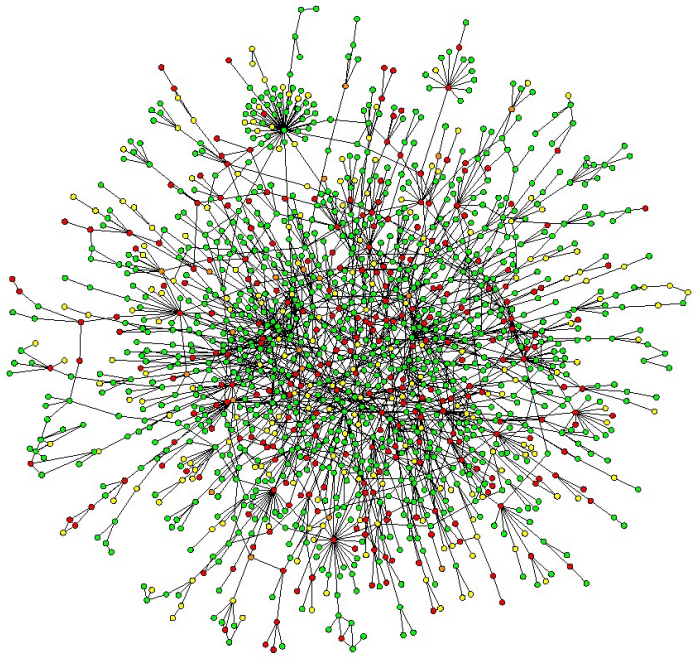


image from bordalierinstitute.com

PPI networks are scale-free *(Albert et al., 2000, Barabasi and Oltvai, 2004)*

- most nodes with small degrees
- a small number of "hubs"
- ultra "small world" property *(Barabasi and Albert, 1999)*

Robustness of Biological Systems

Tolerance to internal mechanistic failures and external changes *(Kitano, 2004)*

- feedback control *(Morohashi et al., 2002)*
- modularity and functional decoupling *(Hartwell et al., 1999)*
- redundancy *(Agrawal, 2001)*

Robustness of Biological Systems

Tolerance to internal mechanistic failures and external changes *(Kitano, 2004)*

- feedback control *(Morohashi et al., 2002)*
- modularity and functional decoupling *(Hartwell et al., 1999)*
- redundancy *(Agrawal, 2001)*

Trade-off between robustness and fragility *(Carlson and Doyle, 2002, Kitano, 2004)*

- cancer
- diabetes
- HIV

Network-based Disease-gene Prioritization

Prioritize genes using "similarity" to known disease-genes (seeds)

guilt-by-association

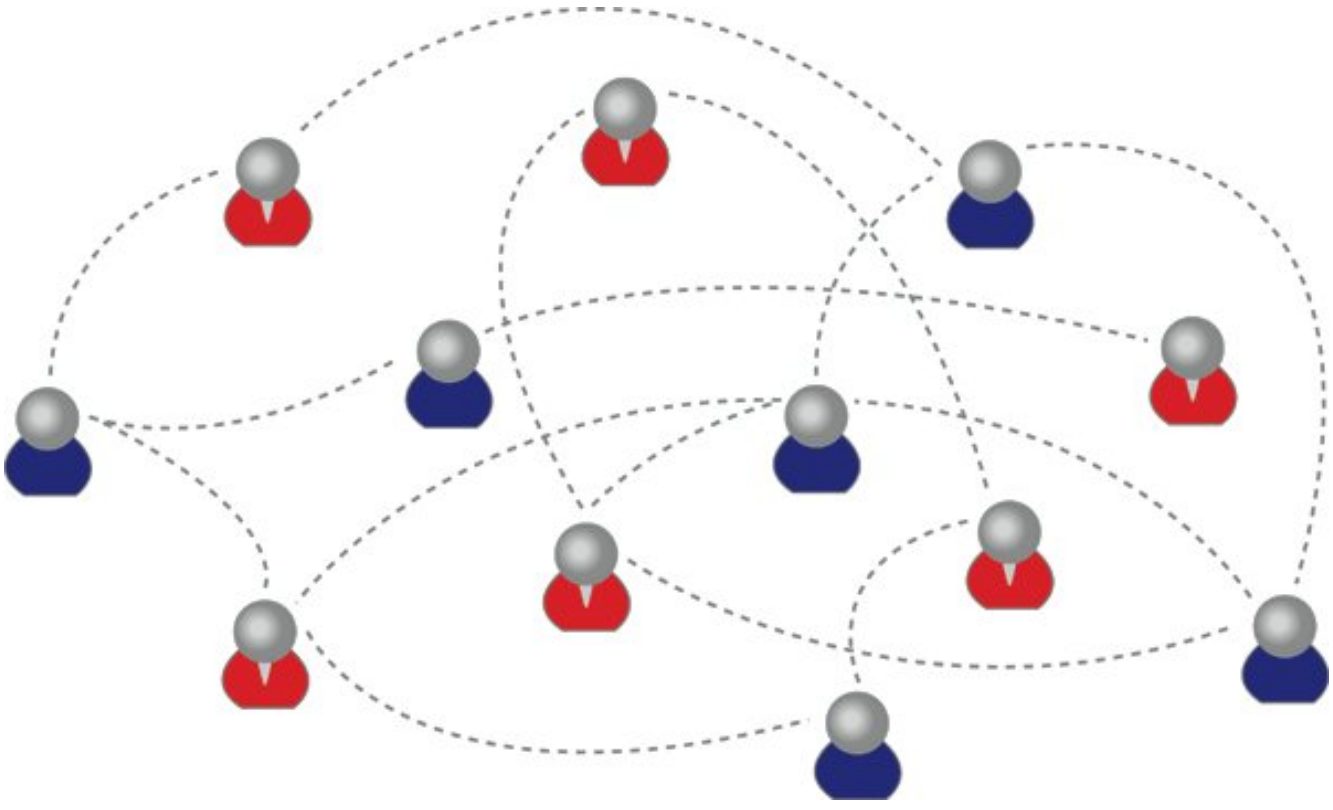
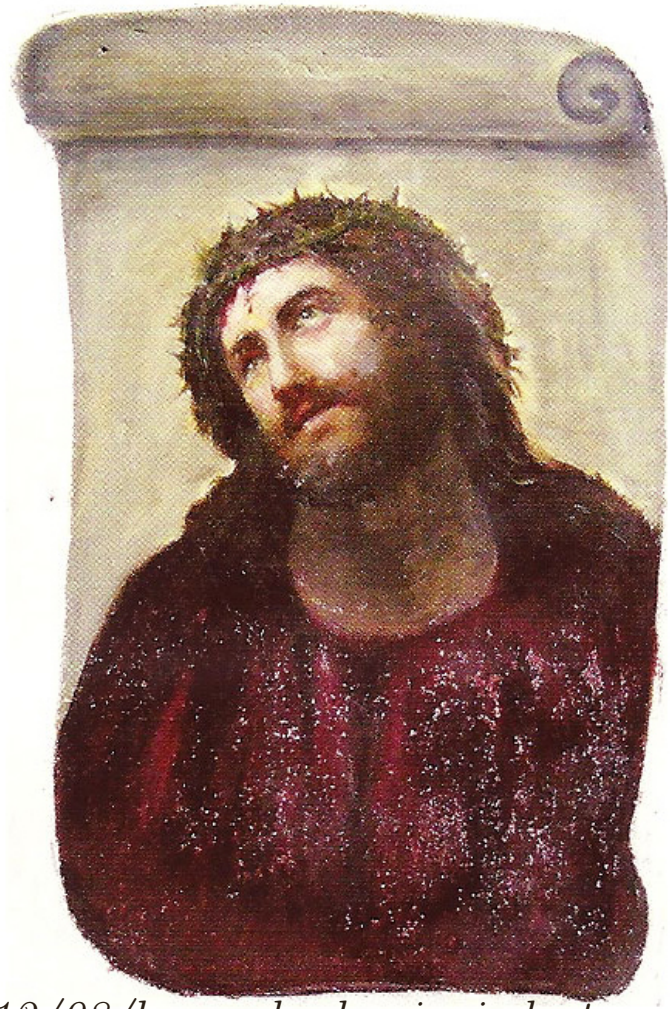


image retrieved from relenet.com

Network-based Disease-gene Prioritization

guilt-by-association

$\Rightarrow$



images from blog.newhumanist.org.uk/2012/08/leonardo-da-vincis-last-supper.html

Network-based Disease-gene Prioritization

guilt-by-association

$\Rightarrow$

images from blog.newhumanist.org.uk/2012/08/leonardo-da-vincis-last-supper.html

Network-based Disease-gene Prioritization

- direct interaction partners *(Xu and Li, 2006, Oti et al., 2006, Lage et al., 2007, Pujana et al., 2007, Aragues et al., 2008, Wu et al., 2008, Ostlund, et al. 2010)*

Network-based Disease-gene Prioritization

- direct interaction partners *(Xu and Li, 2006, Oti et al., 2006, Lage et al., 2007, Pujana et al., 2007, Aragues et al., 2008, Wu et al., 2008, Ostlund, et al. 2010)*
- clustering based *(Milenkovic et al., 2010, Navlakha and Kingsford, 2010)*

Network-based Disease-gene Prioritization

- direct interaction partners *(Xu and Li, 2006, Oti et al., 2006, Lage et al., 2007, Pujana et al., 2007, Aragues et al., 2008, Wu et al., 2008, Ostlund, et al. 2010)*
- clustering based *(Milenkovic et al., 2010, Navlakha and Kingsford, 2010)*
- global topology based
  - calculating shortest distance *(Franke et al., 2006, Wu et al., 2008, Kohler et al., 2008, Dezso et al., 2009)*
  - weighting by a diffusion kernel *(Ma et al., 2007, Nitsch et al., 2010, Qiu et al., 2010)*
  - simulating random walks *(Kohler et al., 2008, Chen et al., 2009, Vanunu et al., 2011)*

Network-based Disease-gene Prioritization

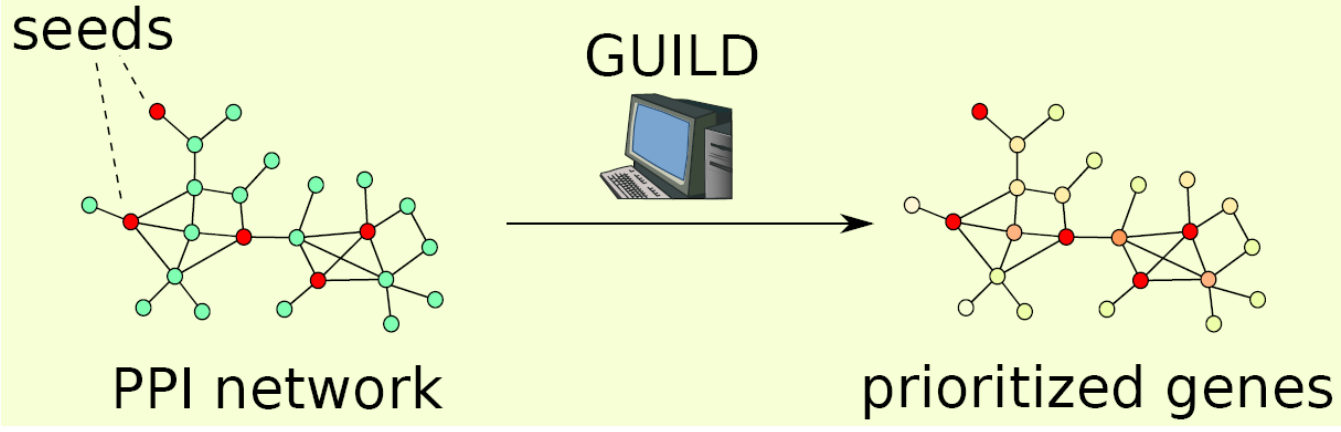
- direct interaction partners *(Xu and Li, 2006, Oti et al., 2006, Lage et al., 2007, Pujana et al., 2007, Aragues et al., 2008, Wu et al., 2008, Ostlund, et al. 2010)*
- clustering based *(Milenkovic et al., 2010, Navlakha and Kingsford, 2010)*
- global topology based
  - calculating shortest distance *(Franke et al., 2006, Wu et al., 2008, Kohler et al., 2008, Dez*
  - weighting by a diff *2010, Qiu et al., 2010)*
  - simulating random *, Vanunu et al., 2011)*

- candidate gene information *(Navlakha and Kingsford, 2010)*
- network quality *(Erten et al., 2011)*

Outline

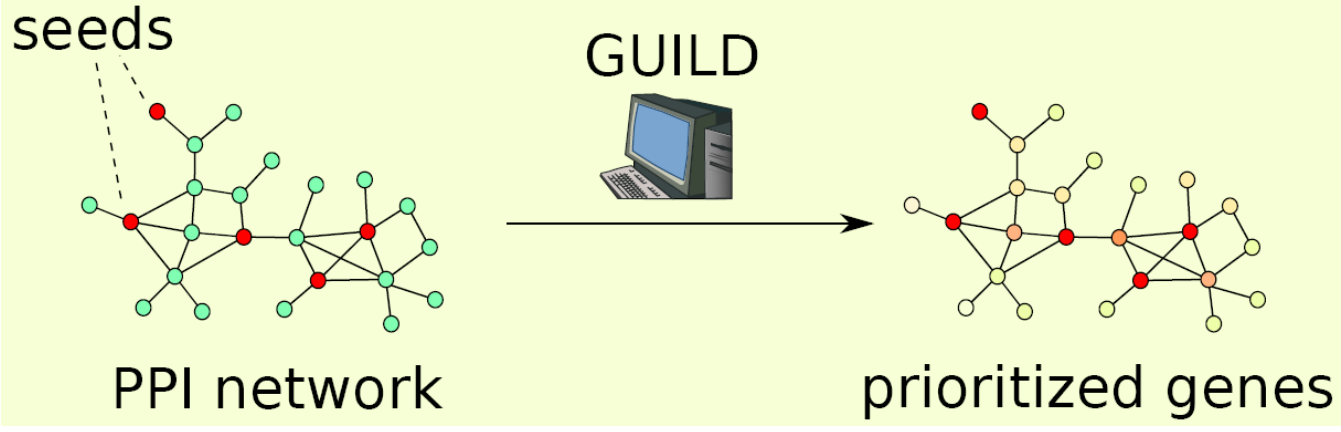
Outline

GUILD framework



- NetScore
- NetZcore
- NetShort
- NetCombo

GUILD framework

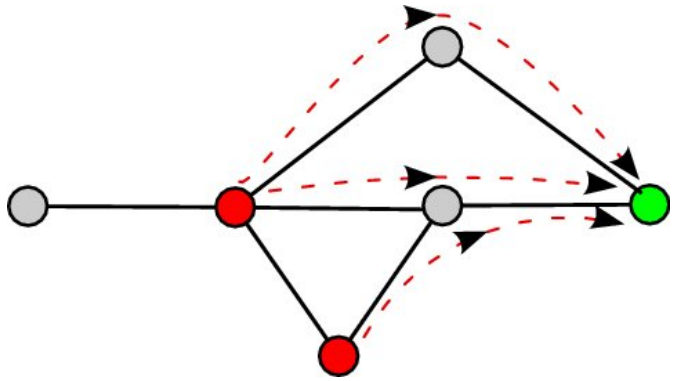


- NetScore
- NetZcore
- NetShort
- NetCombo

- Functional Flow *(Nabieva et al., 2005)*
- Random walk with restart *(Kohler et al., 2008)*
- PageRank with priors *(Chen et al., 2009)*
- Network propagation *(Vanunu et al., 2011)*

NetScore

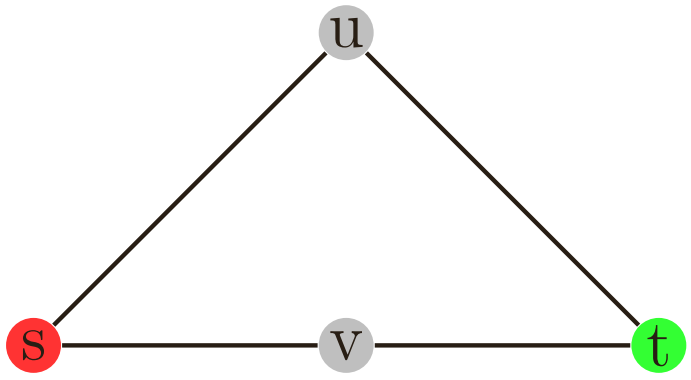
- considers multiple shortest paths in-between the two nodes



$$message = \left\{ \begin{array}{l} source \\ timestamp \\ path\_weight \end{array} \right\}$$

u
m_u
$\vdots$
m_v
$\vdots$

NetScore

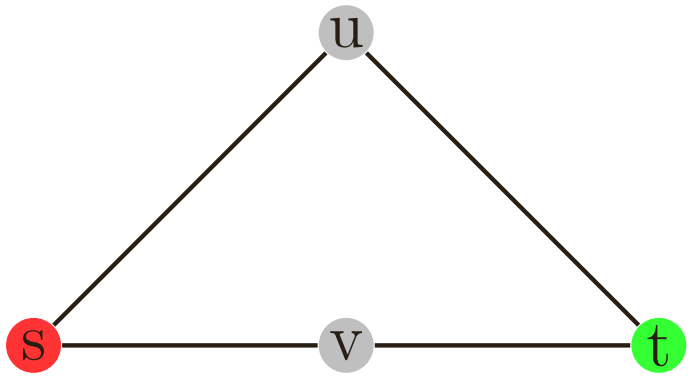


Input: $G = (V, E)$ graph with score property `score`:$V \to [0,1]$, weight property `weight`: $E \to [0,1]$, $nRepetition$, $nIteration$.
Output: $G = (V, E)$ graph with score property `score`$'$:$V \to [0,1]$.

```
for i=1 to nRepetition do
    /* Initialize message arrays                                    */
    foreach u ∈ V do
        m.source ← u // source node id
        m.timestamp ← 0 // the iteration when recieved
        m.path_weight ← 1 // weights of the traveled path
        messages(u) ← { m }
    end
    for j=1 to nIteration do
        /* Digest messages                                          */
        foreach u ∈ V do
            foreach v ∈ {(u, v) ∈ E } do
                foreach m ∈ messages(v) do
                    /* Do not accept messages from the same node
                       recieved during previous iterations          */
                    if AcceptMessage(messages(u), m) then
                        m.timestamp ← j
                        m.path_weight ← m.path_weight * weight(u, v)
                        messages(u) ← messages(u) ∪ m
                end
            end
        end
        /* Update node scores                                       */
        foreach u ∈ V do
            foreach m ∈ messages(u) do
                score(u) ← score(u) + m.path_weight *
                score(m.source)
            end
            score(u) ← score(u) / || messages(u) ||
        end
    end
end
```

s	u	v	t

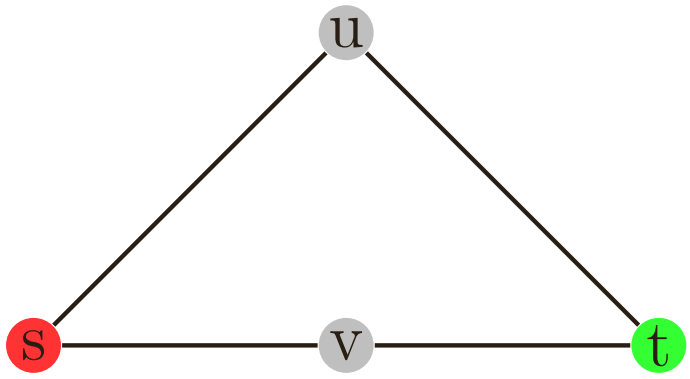
NetScore




```
Input: G = ( V, E) graph with score property score:V → [0, 1],
       weight property weight: E → [0, 1], nRepetition, nIteration.
Output: G = ( V, E) graph with score property score′:V → [0, 1].
for i=1 to nRepetition do
   /* Initialize message arrays                                    */
   foreach u ∈ V do
      m.source ← u // source node id
      m.timestamp ← 0 // the iteration when recieved
      m.path_weight ← 1 // weights of the traveled path
      messages(u) ← { m }
   end
   for j=1 to nIteration do
      /* Digest messages                                          */
      foreach u ∈ V do
         foreach v ∈ {(u, v) ∈ E } do
            foreach m ∈ messages(v) do
               /* Do not accept messages from the same node
                  recieved during previous iterations            */
               if AcceptMessage(messages(u), m) then
                  m.timestamp ← j
                  m.path_weight ← m.path_weight ∗ weight(u, v)
                  messages(u) ← messages(u) ∪ m
               end
            end
         end
      end
      /* Update node scores                                       */
      foreach u ∈ V do
         foreach m ∈ messages(u) do
            score(u) ← score(u) + m.path_weight ∗
            score(m.source)
         end
         score(u) ← score(u) / ‖ messages(u) ‖
      end
   end
end
```

	s	**u**	**v**	**t**
initially	m_s^0	m_u^0	m_v^0	m_t^0

NetScore

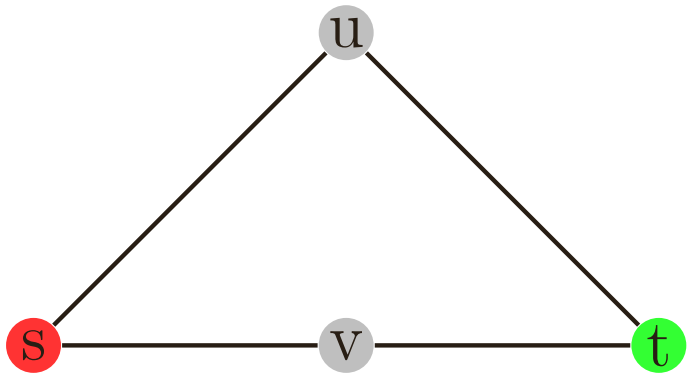



```
Input: G = ( V, E) graph with score property score:V → [0, 1],
       weight property weight: E → [0, 1], nRepetition, nIteration.
Output: G = ( V, E) graph with score property score′:V → [0, 1].
for i=1 to nRepetition do
   /* Initialize message arrays                                    */
   foreach u ∈ V do
      m.source ← u // source node id
      m.timestamp ← 0 // the iteration when recieved
      m.path_weight ← 1 // weights of the traveled path
      messages(u) ← { m }
   end
   for j=1 to nIteration do
      /* Digest messages                                          */
      foreach u ∈ V do
         foreach v ∈ {(u, v) ∈ E } do
            foreach m ∈ messages(v) do
               /* Do not accept messages from the same node
                  recieved during previous iterations            */
               if AcceptMessage(messages(u), m) then
                  m.timestamp ← j
                  m.path_weight ← m.path_weight ∗ weight(u, v)
                  messages(u) ← messages(u) ∪ m
               end
            end
         end
      end
      /* Update node scores                                       */
      foreach u ∈ V do
         foreach m ∈ messages(u) do
            score(u) ← score(u) + m.path_weight ∗
            score(m.source)
         end
         score(u) ← score(u) / ‖ messages(u) ‖
      end
   end
end
```

s	u	v	t
m_s^0	m_u^0	m_v^0	m_t^0
m_u^1			

iteration 1

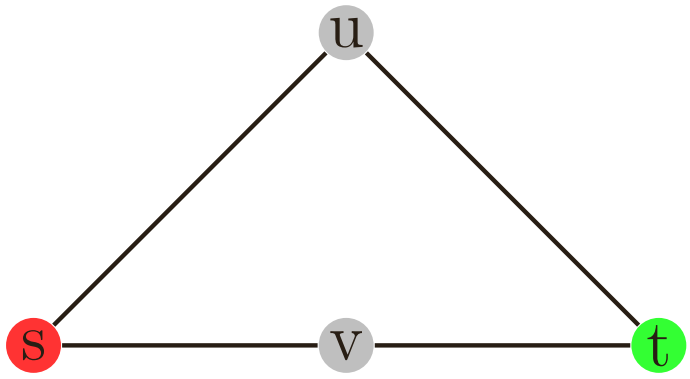
NetScore




```
Input: G = ( V, E) graph with score property score:V → [0, 1],
       weight property weight: E → [0, 1], nRepetition, nIteration.
Output: G = ( V, E) graph with score property score′:V → [0, 1].
for i=1 to nRepetition do
   /* Initialize message arrays                                    */
   foreach u ∈ V do
      m.source ← u // source node id
      m.timestamp ← 0 // the iteration when recieved
      m.path_weight ← 1 // weights of the traveled path
      messages(u) ← { m }
   end
   for j=1 to nIteration do
      /* Digest messages                                          */
      foreach u ∈ V do
         foreach v ∈ {(u, v) ∈ E } do
            foreach m ∈ messages(v) do
               /* Do not accept messages from the same node
                  recieved during previous iterations            */
               if AcceptMessage(messages(u), m) then
                  m.timestamp ← j
                  m.path_weight ← m.path_weight ∗ weight(u, v)
                  messages(u) ← messages(u) ∪ m
               end
            end
         end
      end
      /* Update node scores                                       */
      foreach u ∈ V do
         foreach m ∈ messages(u) do
            score(u) ← score(u) + m.path_weight ∗
            score(m.source)
         end
         score(u) ← score(u) / ‖ messages(u) ‖
      end
   end
end
```

	s	**u**	**v**	**t**
	m_s^0	m_u^0	m_v^0	m_t^0
iteration 1	m_u^1			
	m_v^1			

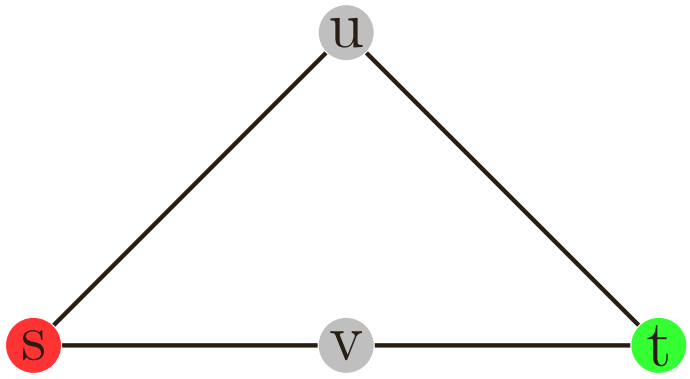
NetScore




```
Input: G = ( V, E) graph with score property score:V → [0, 1],
       weight property weight: E → [0, 1], nRepetition, nIteration.
Output: G = ( V, E) graph with score property score′:V → [0, 1].
for i=1 to nRepetition do
   /* Initialize message arrays                                    */
   foreach u ∈ V do
      m.source ← u // source node id
      m.timestamp ← 0 // the iteration when recieved
      m.path_weight ← 1 // weights of the traveled path
      messages(u) ← { m }
   end
   for j=1 to nIteration do
      /* Digest messages                                          */
      foreach u ∈ V do
         foreach v ∈ {(u, v) ∈ E } do
            foreach m ∈ messages(v) do
               /* Do not accept messages from the same node
                  recieved during previous iterations            */
               if AcceptMessage(messages(u), m) then
                  m.timestamp ← j
                  m.path_weight ← m.path_weight ∗ weight(u, v)
                  messages(u) ← messages(u) ∪ m
               end
            end
         end
      end
      /* Update node scores                                       */
      foreach u ∈ V do
         foreach m ∈ messages(u) do
            score(u) ← score(u) + m.path_weight ∗
            score(m.source)
         end
         score(u) ← score(u) / ‖ messages(u) ‖
      end
   end
end
```

	s	u	v	t
	m_s^0	m_u^0	m_v^0	m_t^0
iteration 1	m_u^1	m_s^1		
	m_v^1	m_t^1		

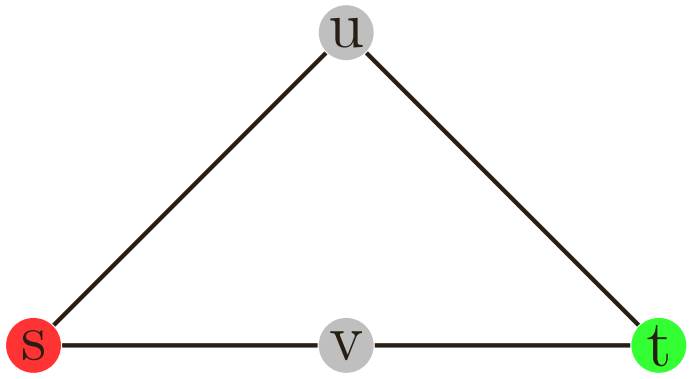
NetScore




```
Input: G = ( V, E) graph with score property score:V → [0, 1],
       weight property weight: E → [0, 1], nRepetition, nIteration.
Output: G = ( V, E) graph with score property score′:V → [0, 1].
for i=1 to nRepetition do
   /* Initialize message arrays                                    */
   foreach u ∈ V do
      m.source ← u // source node id
      m.timestamp ← 0 // the iteration when recieved
      m.path_weight ← 1 // weights of the traveled path
      messages(u) ← { m }
   end
   for j=1 to nIteration do
      /* Digest messages                                          */
      foreach u ∈ V do
         foreach v ∈ {(u, v) ∈ E } do
            foreach m ∈ messages(v) do
               /* Do not accept messages from the same node
                  recieved during previous iterations            */
               if AcceptMessage(messages(u), m) then
                  m.timestamp ← j
                  m.path_weight ← m.path_weight ∗ weight(u, v)
                  messages(u) ← messages(u) ∪ m
               end
            end
         end
      end
      /* Update node scores                                       */
      foreach u ∈ V do
         foreach m ∈ messages(u) do
            score(u) ← score(u) + m.path_weight ∗
            score(m.source)
         end
         score(u) ← score(u) / ‖ messages(u) ‖
      end
   end
end
```

	s	u	v	t
	m_s^0	m_u^0	m_v^0	m_t^0
iteration 1	m_u^1	m_s^1	m_s^1	
	m_v^1	m_t^1	m_t^1	

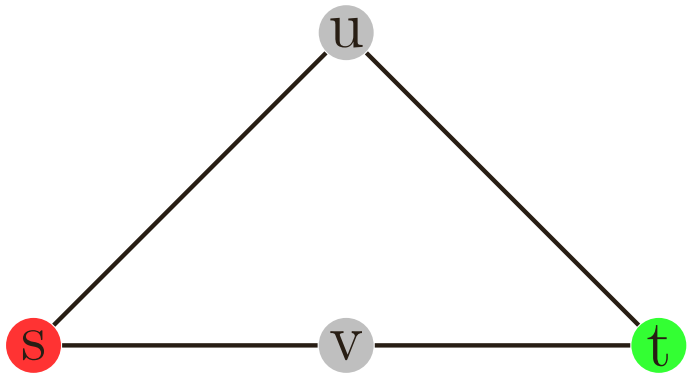
NetScore




```
Input: G = ( V, E) graph with score property score:V → [0, 1],
       weight property weight: E → [0, 1], nRepetition, nIteration.
Output: G = ( V, E) graph with score property score′:V → [0, 1].
for i=1 to nRepetition do
   /* Initialize message arrays                                    */
   foreach u ∈ V do
      m.source ← u // source node id
      m.timestamp ← 0 // the iteration when recieved
      m.path_weight ← 1 // weights of the traveled path
      messages(u) ← { m }
   end
   for j=1 to nIteration do
      /* Digest messages                                          */
      foreach u ∈ V do
         foreach v ∈ {(u, v) ∈ E } do
            foreach m ∈ messages(v) do
               /* Do not accept messages from the same node
                  recieved during previous iterations            */
               if AcceptMessage(messages(u), m) then
                  m.timestamp ← j
                  m.path_weight ← m.path_weight ∗ weight(u, v)
                  messages(u) ← messages(u) ∪ m
               end
            end
         end
      end
      /* Update node scores                                       */
      foreach u ∈ V do
         foreach m ∈ messages(u) do
            score(u) ← score(u) + m.path_weight ∗
            score(m.source)
         end
         score(u) ← score(u) / ‖ messages(u) ‖
      end
   end
end
```

	s	u	v	t
iteration 1	m_s^0	m_u^0	m_v^0	m_t^0
	m_u^1	m_s^1	m_s^1	m_u^1
	m_v^1	m_t^1	m_t^1	

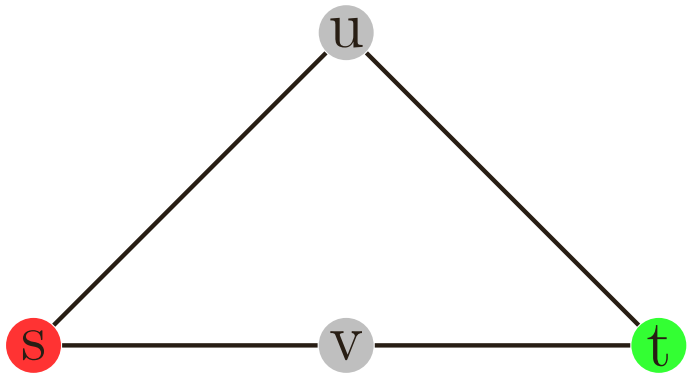
NetScore




```
Input: G = ( V, E) graph with score property score:V → [0, 1],
       weight property weight: E → [0, 1], nRepetition, nIteration.
Output: G = ( V, E) graph with score property score′:V → [0, 1].
for i=1 to nRepetition do
   /* Initialize message arrays                                    */
   foreach u ∈ V do
      m.source ← u // source node id
      m.timestamp ← 0 // the iteration when recieved
      m.path_weight ← 1 // weights of the traveled path
      messages(u) ← { m }
   end
   for j=1 to nIteration do
      /* Digest messages                                          */
      foreach u ∈ V do
         foreach v ∈ {(u, v) ∈ E } do
            foreach m ∈ messages(v) do
               /* Do not accept messages from the same node
                  recieved during previous iterations            */
               if AcceptMessage(messages(u), m) then
                  m.timestamp ← j
                  m.path_weight ← m.path_weight ∗ weight(u, v)
                  messages(u) ← messages(u) ∪ m
               end
            end
         end
      end
      /* Update node scores                                       */
      foreach u ∈ V do
         foreach m ∈ messages(u) do
            score(u) ← score(u) + m.path_weight ∗
            score(m.source)
         end
         score(u) ← score(u) / ‖ messages(u) ‖
      end
   end
end
```

	s	u	v	t
iteration 1	m_s^0	m_u^0	m_v^0	m_t^0
	m_u^1	m_s^1	m_s^1	m_u^1
	m_v^1	m_t^1	m_t^1	m_v^1

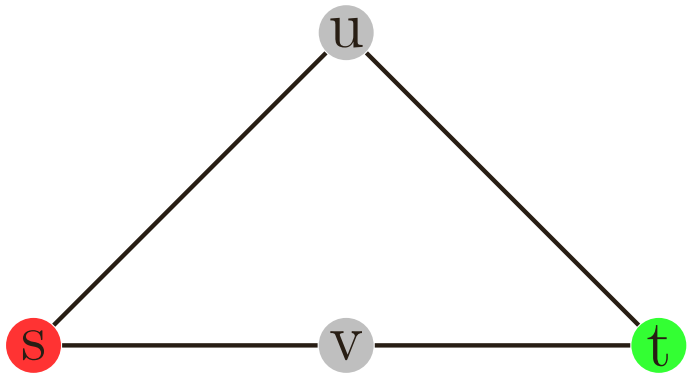
NetScore




```
Input: G = ( V, E) graph with score property score:V → [0, 1],
       weight property weight: E → [0, 1], nRepetition, nIteration.
Output: G = ( V, E) graph with score property score′:V → [0, 1].
for i=1 to nRepetition do
   /* Initialize message arrays                                    */
   foreach u ∈ V do
      m.source ← u // source node id
      m.timestamp ← 0 // the iteration when recieved
      m.path_weight ← 1 // weights of the traveled path
      messages(u) ← { m }
   end
   for j=1 to nIteration do
      /* Digest messages                                          */
      foreach u ∈ V do
         foreach v ∈ {(u, v) ∈ E } do
            foreach m ∈ messages(v) do
               /* Do not accept messages from the same node
                  recieved during previous iterations            */
               if AcceptMessage(messages(u), m) then
                  m.timestamp ← j
                  m.path_weight ← m.path_weight ∗ weight(u, v)
                  messages(u) ← messages(u) ∪ m
               end
            end
         end
      end
      /* Update node scores                                       */
      foreach u ∈ V do
         foreach m ∈ messages(u) do
            score(u) ← score(u) + m.path_weight ∗
            score(m.source)
         end
         score(u) ← score(u) / ‖ messages(u) ‖
      end
   end
end
```

	s	u	v	t
	m_s^0	m_u^0	m_v^0	m_t^0
	m_u^1	m_s^1	m_s^1	m_u^1
	m_v^1	m_t^1	m_t^1	m_v^1
iteration 2	$2m_t^2$			

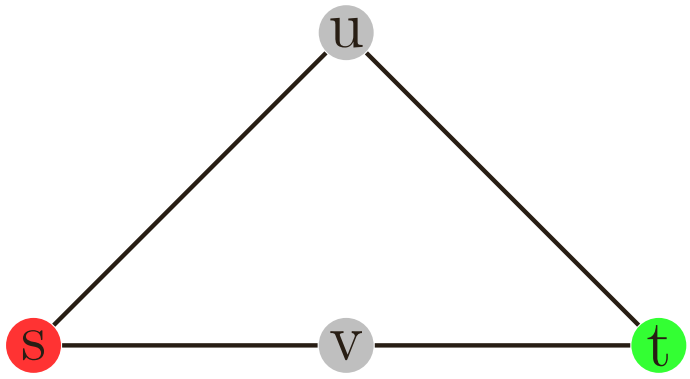
NetScore




```
Input: G = ( V, E) graph with score property score:V → [0, 1],
       weight property weight: E → [0, 1], nRepetition, nIteration.
Output: G = ( V, E) graph with score property score′:V → [0, 1].
for i=1 to nRepetition do
   /* Initialize message arrays                                    */
   foreach u ∈ V do
      m.source ← u // source node id
      m.timestamp ← 0 // the iteration when recieved
      m.path_weight ← 1 // weights of the traveled path
      messages(u) ← { m }
   end
   for j=1 to nIteration do
      /* Digest messages                                          */
      foreach u ∈ V do
         foreach v ∈ {(u, v) ∈ E } do
            foreach m ∈ messages(v) do
               /* Do not accept messages from the same node
                  recieved during previous iterations            */
               if AcceptMessage(messages(u), m) then
                  m.timestamp ← j
                  m.path_weight ← m.path_weight ∗ weight(u, v)
                  messages(u) ← messages(u) ∪ m
               end
            end
         end
      end
      /* Update node scores                                       */
      foreach u ∈ V do
         foreach m ∈ messages(u) do
            score(u) ← score(u) + m.path_weight ∗
            score(m.source)
         end
         score(u) ← score(u) / ‖ messages(u) ‖
      end
   end
end
```

	s	u	v	t
	m_s^0	m_u^0	m_v^0	m_t^0
	m_u^1	m_s^1	m_s^1	m_u^1
iteration 2	m_v^1	m_t^1	m_t^1	m_v^1
	$2m_t^2$	$2m_v^2$	$2m_u^2$	

NetScore

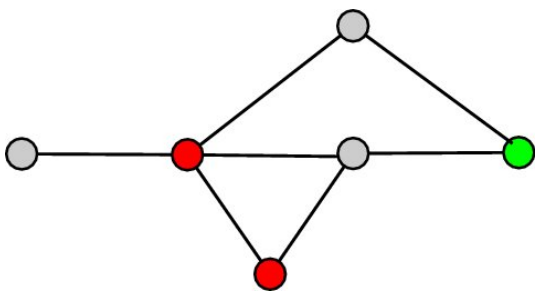


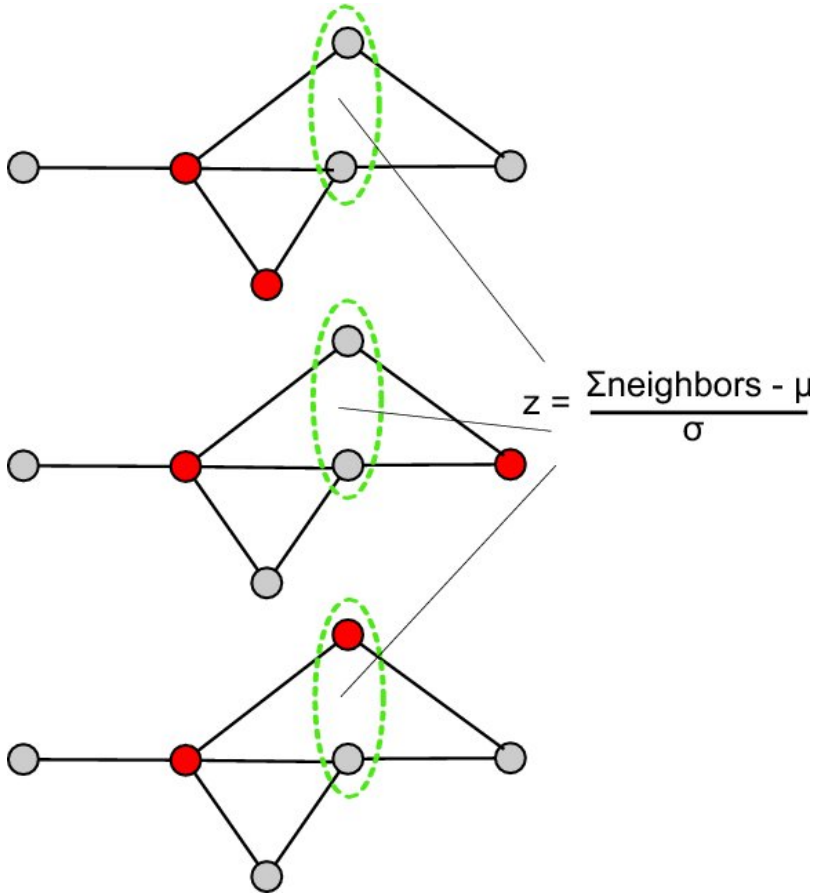

```
Input: G = ( V, E) graph with score property score:V → [0, 1],
       weight property weight: E → [0, 1], nRepetition, nIteration.
Output: G = ( V, E) graph with score property score′:V → [0, 1].
for i=1 to nRepetition do
   /* Initialize message arrays                                    */
   foreach u ∈ V do
      m.source ← u // source node id
      m.timestamp ← 0 // the iteration when recieved
      m.path_weight ← 1 // weights of the traveled path
      messages(u) ← { m }
   end
   for j=1 to nIteration do
      /* Digest messages                                          */
      foreach u ∈ V do
         foreach v ∈ {(u, v) ∈ E } do
            foreach m ∈ messages(v) do
               /* Do not accept messages from the same node
                  recieved during previous iterations            */
               if AcceptMessage(messages(u), m) then
                  m.timestamp ← j
                  m.path_weight ← m.path_weight ∗ weight(u, v)
                  messages(u) ← messages(u) ∪ m
               end
            end
         end
      end
      /* Update node scores                                       */
      foreach u ∈ V do
         foreach m ∈ messages(u) do
            score(u) ← score(u) + m.path_weight ∗
            score(m.source)
         end
         score(u) ← score(u) / ‖ messages(u) ‖
      end
   end
end
```

	s	u	v	t
	m_s^0	m_u^0	m_v^0	m_t^0
	m_u^1	m_s^1	m_s^1	m_u^1
	m_v^1	m_t^1	m_t^1	m_v^1
iteration 2	$2m_t^2$	$2m_v^2$	$2m_u^2$	$2m_s^2$

NetZcore

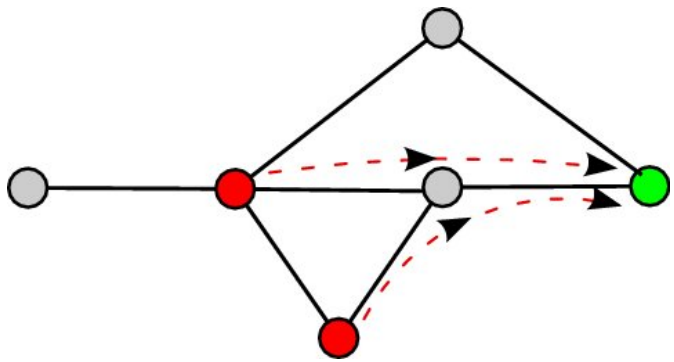
- checks the "significance" of the neighborhood configuration





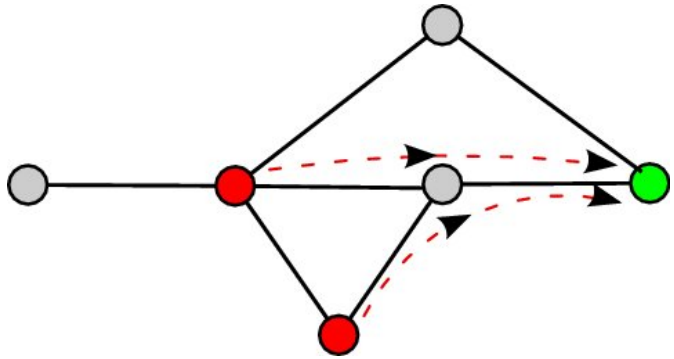
NetShort

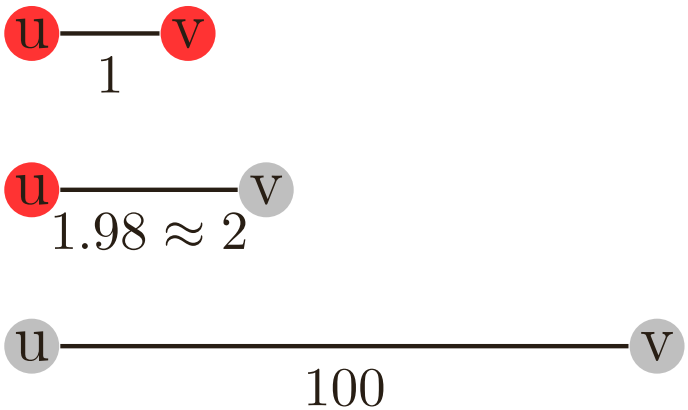
- incorporates "disease-relevance" of a path
- the more seeds the path has, the shorter it is



NetShort

- incorporates "disease-relevance" of a path
- the more seeds the path has, the shorter it is

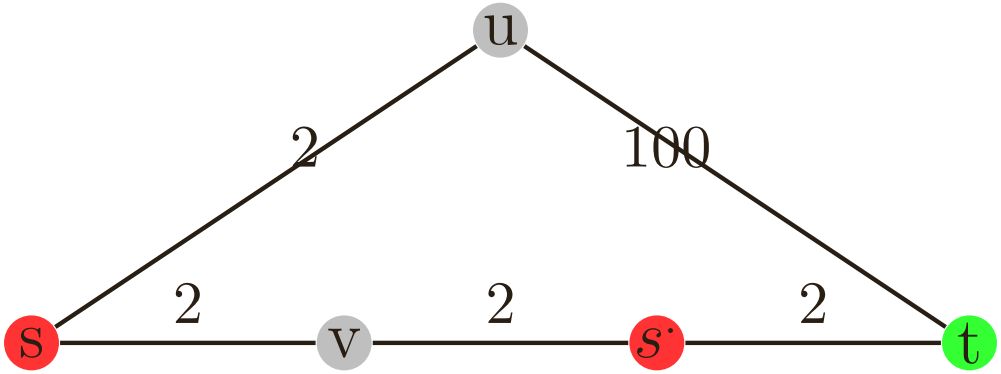


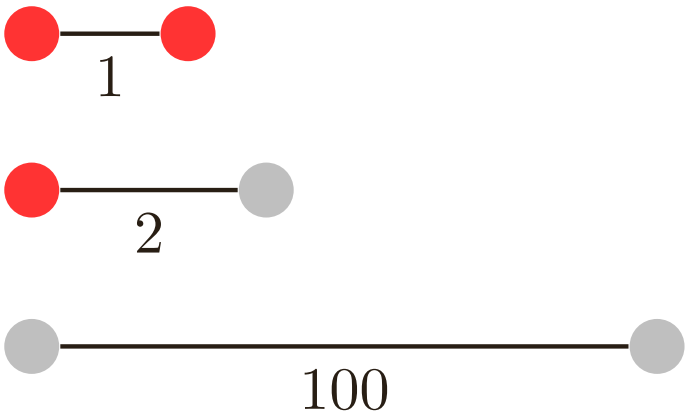


$$score(u) = \begin{cases} 1, & if\ u\ is\ seed \\ 0.01, & otherwise \end{cases}$$

$$weight(u,v) = \frac{1}{(score(u)+score(v))/2}$$

NetShort





$$path\_length(s,t)_{\{s,u,t\}} = 2 + 100 = 102$$

$$path\_length(s,t)_{\{s,v,s\cdot,t\}} = 2 + 2 + 2 = 6$$

NetCombo

- combines the scores of the three methods
- calculates a z-score for each node using the score distribution of each method
- averages the z-score from all three algorithms for each node

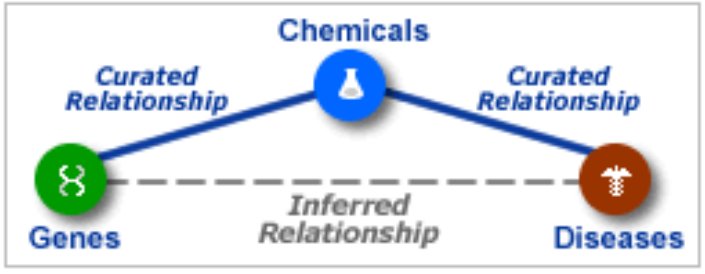
$$score_u^{NetCombo} = \frac{1}{3} * \sum_{method \in \{NetScore, NetZcore, NetCombo\}} \frac{score_u^{method} - \mu^{method}}{\sigma^{method}}$$

Disease-gene Association Data

- OMIM *(Hamosh et al., 2002)*
  - expert curated human disease mutations
  - disorders merged using the first word
  - use merged sets containing at least 5 genes in the PPI network
- Goh *(Goh et al., 2007)*
  - 22 disorder classes w.r.t. the physiological system effected
  - manually group disease-gene associations from OMIM
- Chen *(Chen et al., 2009)*
  - collection of genes for 19 diseases from OMIM and GAD

Disease-gene Association Data

- GAD: Genetic Association Database *(Becker et al., 2004)*
  - disease-gene associations with both high & low significance
- CTD: Comparative Toxicogenomics Database *(Davis et al., 2010)*
  - direct: manually curated
  - indirect: inferred



Creating Inferred Gene-Disease Relationships

image from ctdbase.org

Protein-protein Interaction Data

Protein-protein interaction (PPI) data is spread among various repositories

MIPS

DIP

Human Protein Reference Database

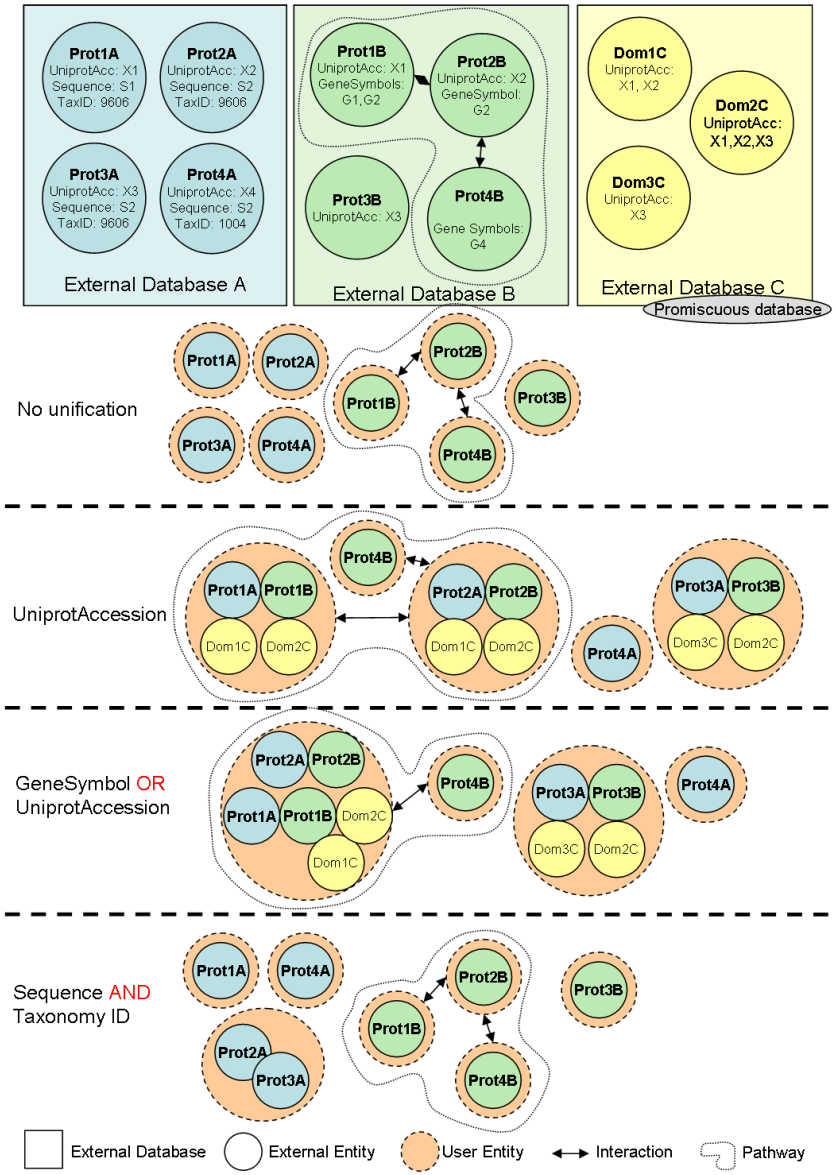
IntAct

STRING

MINT

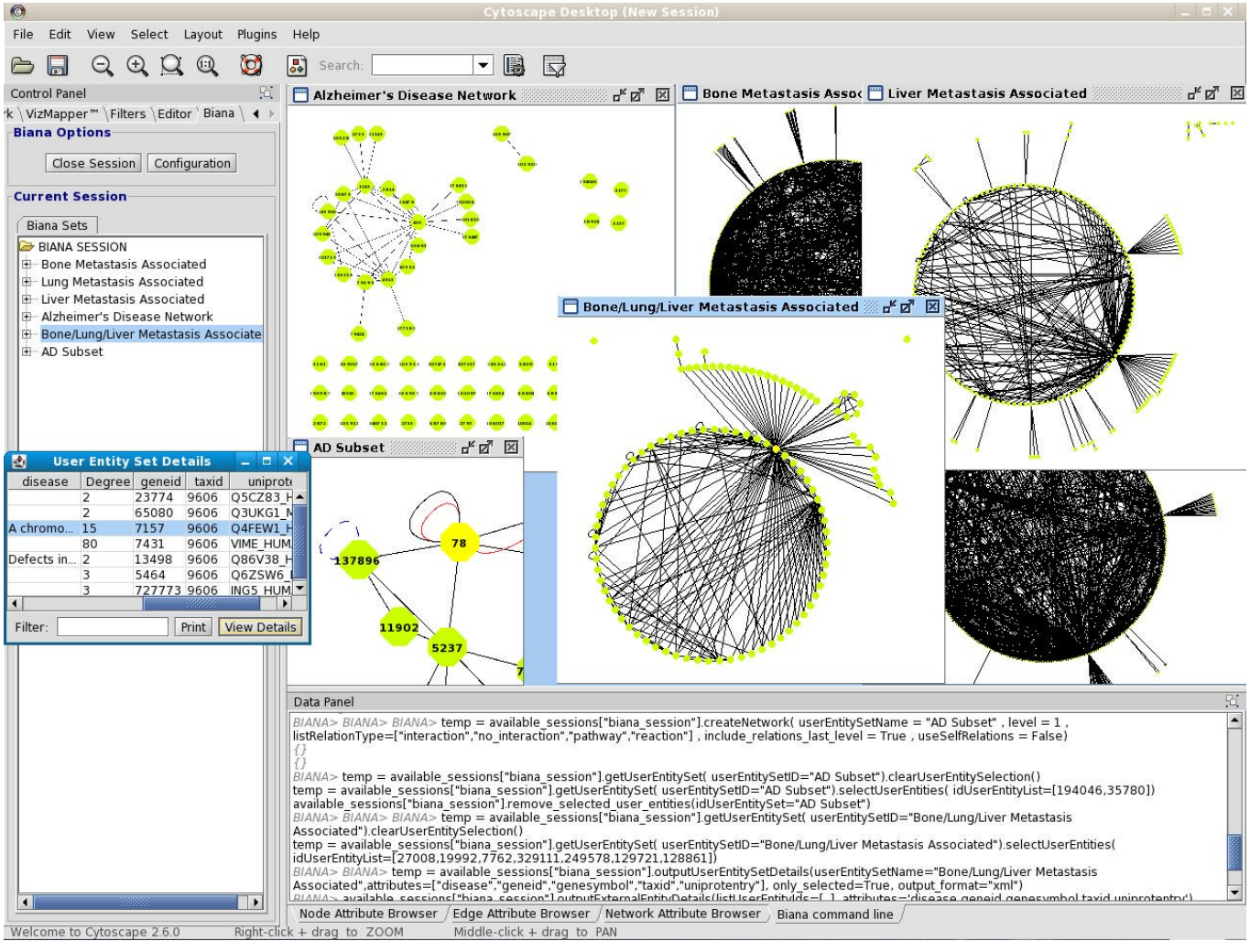
BioGrid

Protein-protein Interaction Data



BIANA
(Garcia-Garcia et al., 2010)

Protein-protein Interaction Data



BIANA (Biological Interactions and Network Analysis)

Garcia-Garcia et al., 2010

Protein-protein Interaction Data

- PPI network
  - human interactome integrated using BIANA

Database	**Retrieval date**
DIP	January, 2009
HPRD	September, 2007
IntAct	January, 2009
MIPS (MPACT)	October, 2008
BIOGRID	January, 2009

Protein-protein Interaction Data

- PPI network
  - human interactome integrated using BIANA

Database	**Retrieval date**
DIP	January, 2009
HPRD	September, 2007
IntAct	January, 2009
MIPS (MPACT)	October, 2008
BIOGRID	January, 2009

- bPPI network
  - PPI network excluding interactions detected by pull down methods

Protein-protein Interaction Data

- PPI network
  - human interactome integrated using BIANA

Database	**Retrieval date**
DIP	January, 2009
HPRD	September, 2007
IntAct	January, 2009
MIPS (MPACT)	October, 2008
BIOGRID	January, 2009

- bPPI network
  - PPI network excluding interactions detected by pull down methods
- Weighted bPPI network
  - bPPI network with edge weights from STRING

Protein-protein Interaction Data

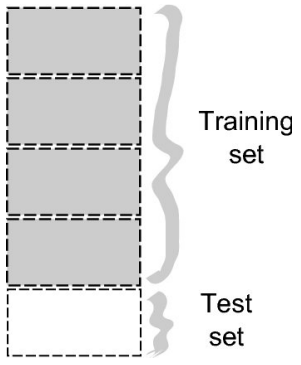
- Goh network
  - two large scale Y2H data sets *(Goh et al., 2007)*
- ENTREZ network
  - PPI interactions from BIND and HPRD

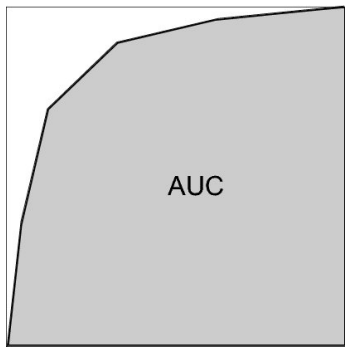
Protein-protein Interaction Data

Network	Nodes	Edges	Avg. degree	Highest degree
PPI	12506	205966	32.9	792
bPPI	11250	59220	10.5	475
Goh	7279	21911	6.0	176
ENTREZ	10119	49826	9.8	333

Evaluation of Classification Performance

- five-fold cross validation
- area under ROC curve (AUC)
- sensitivity at top 1%





Outline

GUILD Predicts Disease-gene Association Better than Existing Network-based Prioritization Methods

Data Set	Metric	NetScore	NetZcore	NetShort	NetCombo	Func. Flow	PageRank	Random Walk	Network Prop.
OMIM	AUC	67.49	62.99	65.63	**72.09**	58.55	57.03	55.36	65.97
	Sens.	20.69	19.62	15.41	21.46	22.31	10.76	14.64	**23.24**
Goh	AUC	**67.32**	61.45	55.36	67.08	54.78	52.39	49.35	54.74
	Sens.	**11.61**	11.05	4.88	11.34	6.22	4.00	5.69	8.66
Chen	AUC	75.92	72.80	63.11	**78.41**	63.56	65.30	61.78	69.07
	Sens.	**18.89**	12.84	9.06	17.51	12.43	6.00	9.64	15.30

GUILD Predicts Disease-gene Association Better than Existing Network-based Prioritization Methods

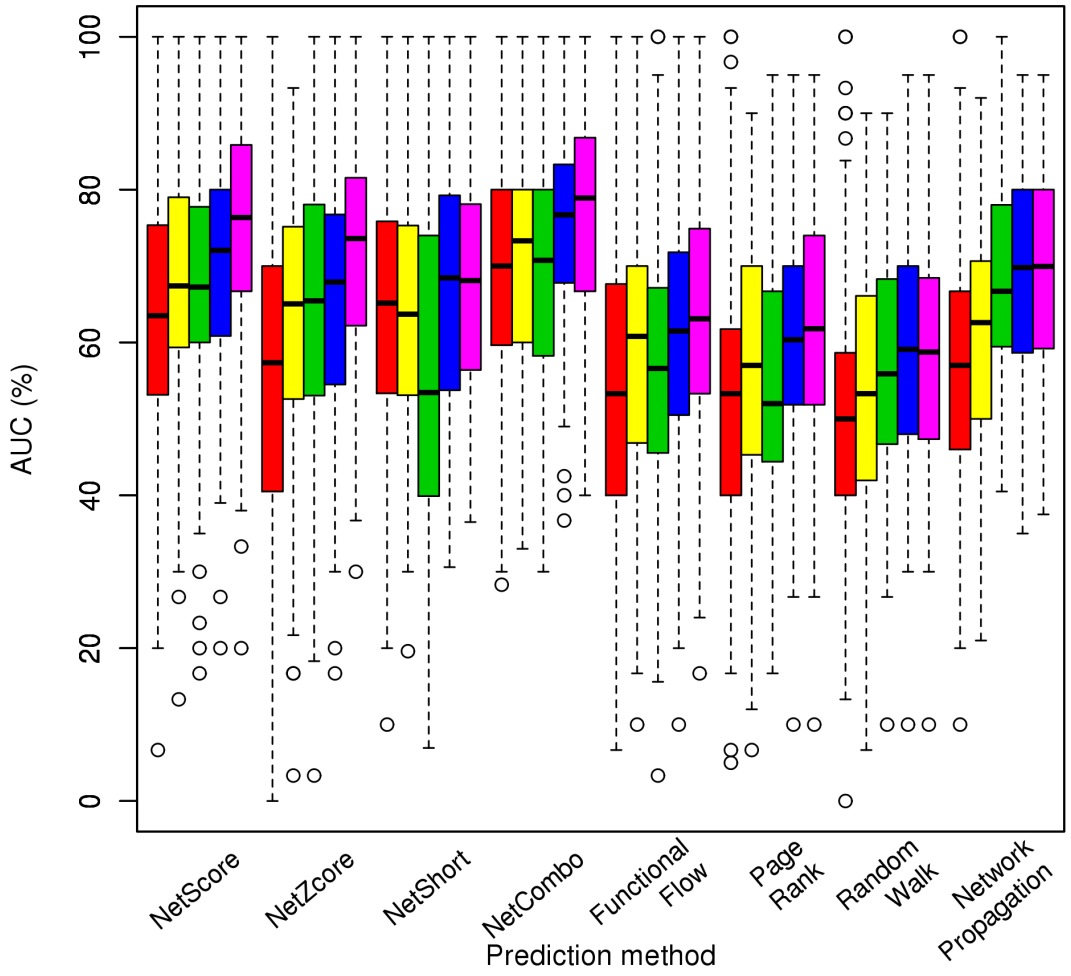
Data Set	Metric	NetScore	NetZcore	NetShort	NetCombo	Func. Flow	PageRank	Random Walk	Network Prop.
OMIM	AUC	67.49	62.99	65.63	**72.09**	58.55	57.03	55.36	65.97
	Sens.	20.69	19.62	15.41	21.46	22.31	10.76	14.64	**23.24**
Goh	AUC	**67.32**	61.45	55.36	67.08	54.78	52.39	49.35	54.74
	Sens.	**11.61**	11.05	4.88	11.34	6.22	4.00	5.69	8.66
Chen	AUC	75.92	72.80	63.11	**78.41**	63.56	65.30	61.78	69.07
	Sens.	**18.89**	12.84	9.06	17.51	12.43	6.00	9.64	15.30

Improvement over existing methods is significant

- for NetCombo in all three datasets
- for NetScore in Goh and Chen datasets

Guney and Oliva, 2012

Prioritization Accuracy Depends on the Interaction Network Quality



Goh
ENTREZ
PPI
bPPI
weighted bPPI

Guney and Oliva, 2012

Prioritization Methods Depend on the Connectedness of Seeds Rather than the Number of Seeds

	Number of seeds	Average number of neighboring seeds	Average shortest path length between seeds
NetScore	-0.01 (0.86)	0.41 (1.1e-6)	-0.43 (2.1e-7)
NetZcore	-0.04 (0.65)	0.45 (6.7e-8)	-0.54 (2.1e-11)
NetShort	-0.38 (7.7e-6)	0.30 (4.8e-4)	-0.65 (6.6-17)
NetCombo	-0.22 (0.01)	0.31 (2.5e-4)	-0.56 (4.9e-12)
Func. Flow	-0.07 (0.43)	0.49 (1.8e-9)	-0.46 (2.4e-8)
PageRank	-0.10 (0.24)	0.53 (8.3e-11)	-0.58 (4.0e-13)
Random Walk	-0.14 (0.12)	0.53 (4.4e-11)	-0.51 (5.1e-10)
Network Prop.	-0.31 (2.5e-4)	0.43 (3.5e-7)	-0.55 (1.3e-11)

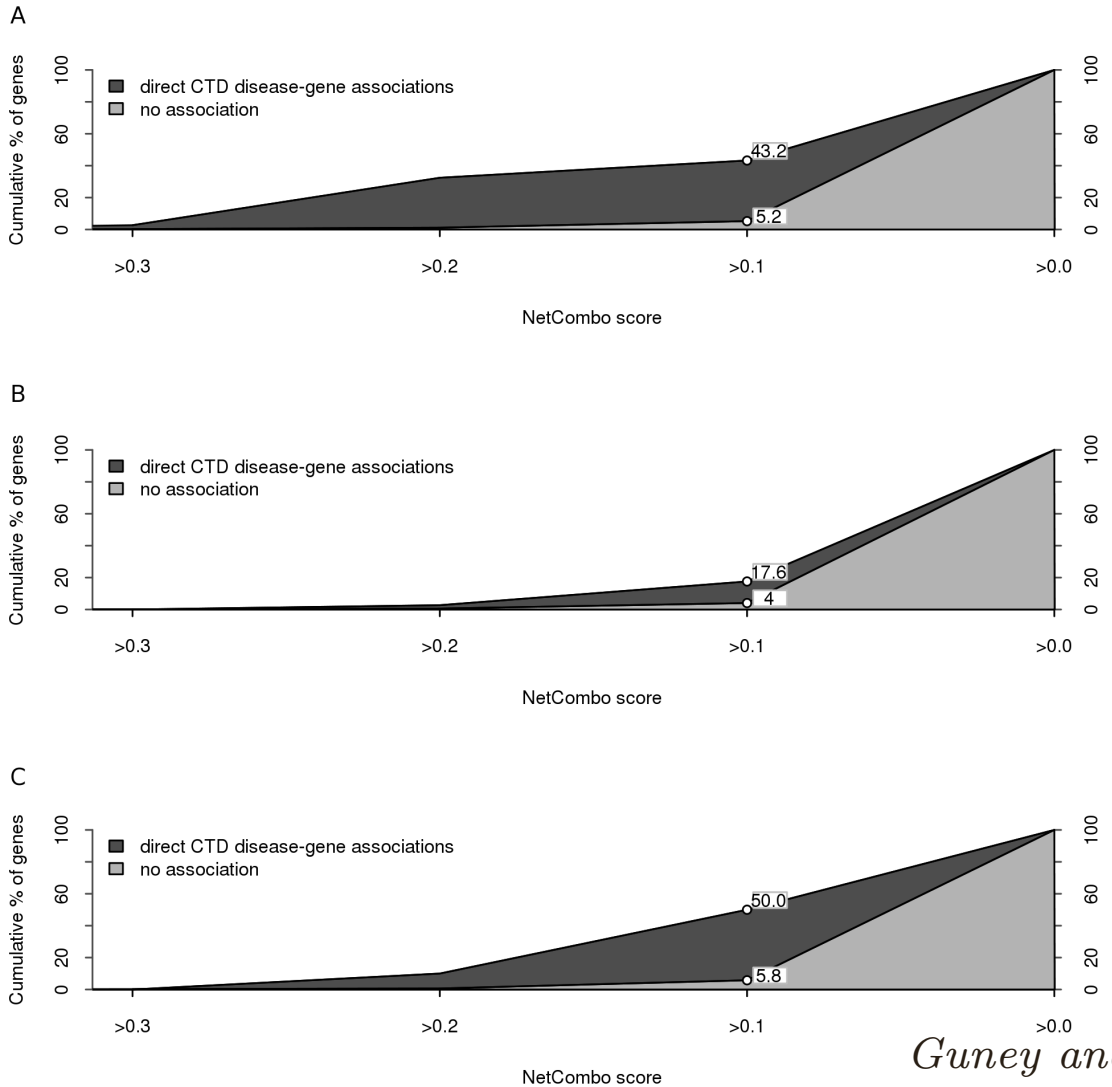
Guney and Oliva, 2012

GUILD Scores Distinguish Disease-Genes from the Rest of the Genes

Alzheimer (AD)

Diabetes

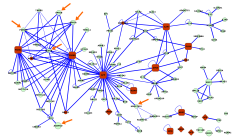
AIDS

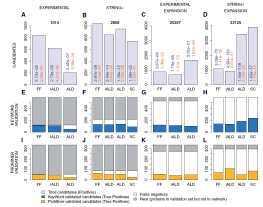


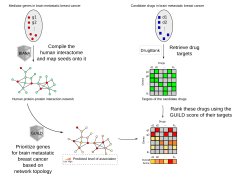
Guney and Oliva, 2012

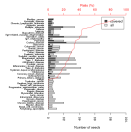
Several Applications of GUILD

- implicating genes in AD *(Guney and Oliva, 2012)*
- extending apoptosis pathway using GUILD *(Planas et al., 2012)*
- prioritizing genes in bone metastatic breast cancer *(Santana et al., submitted)*
- drug repurposing using GUILD
- network-based prioritization of GWAS data
- disease comorbidity through common genes

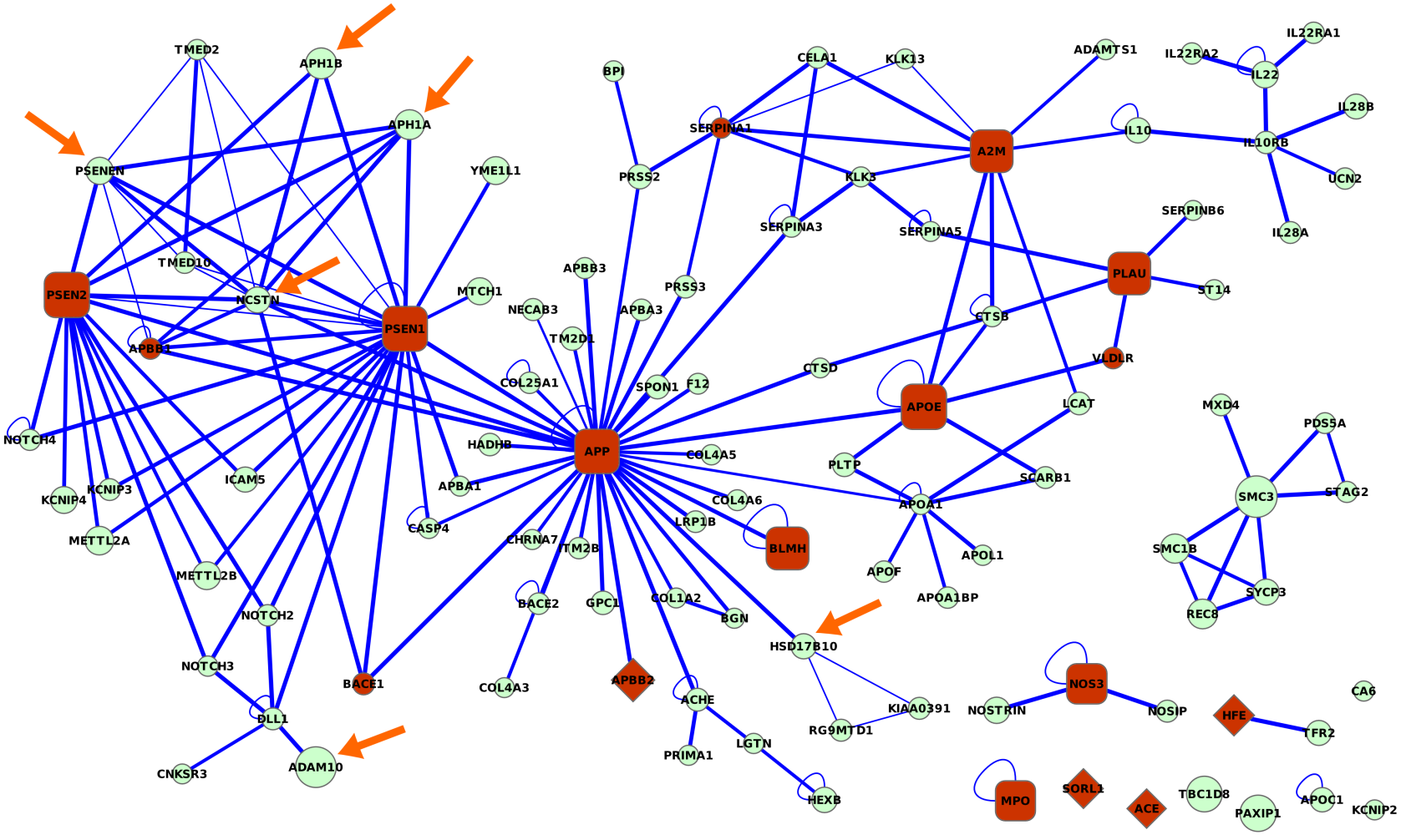








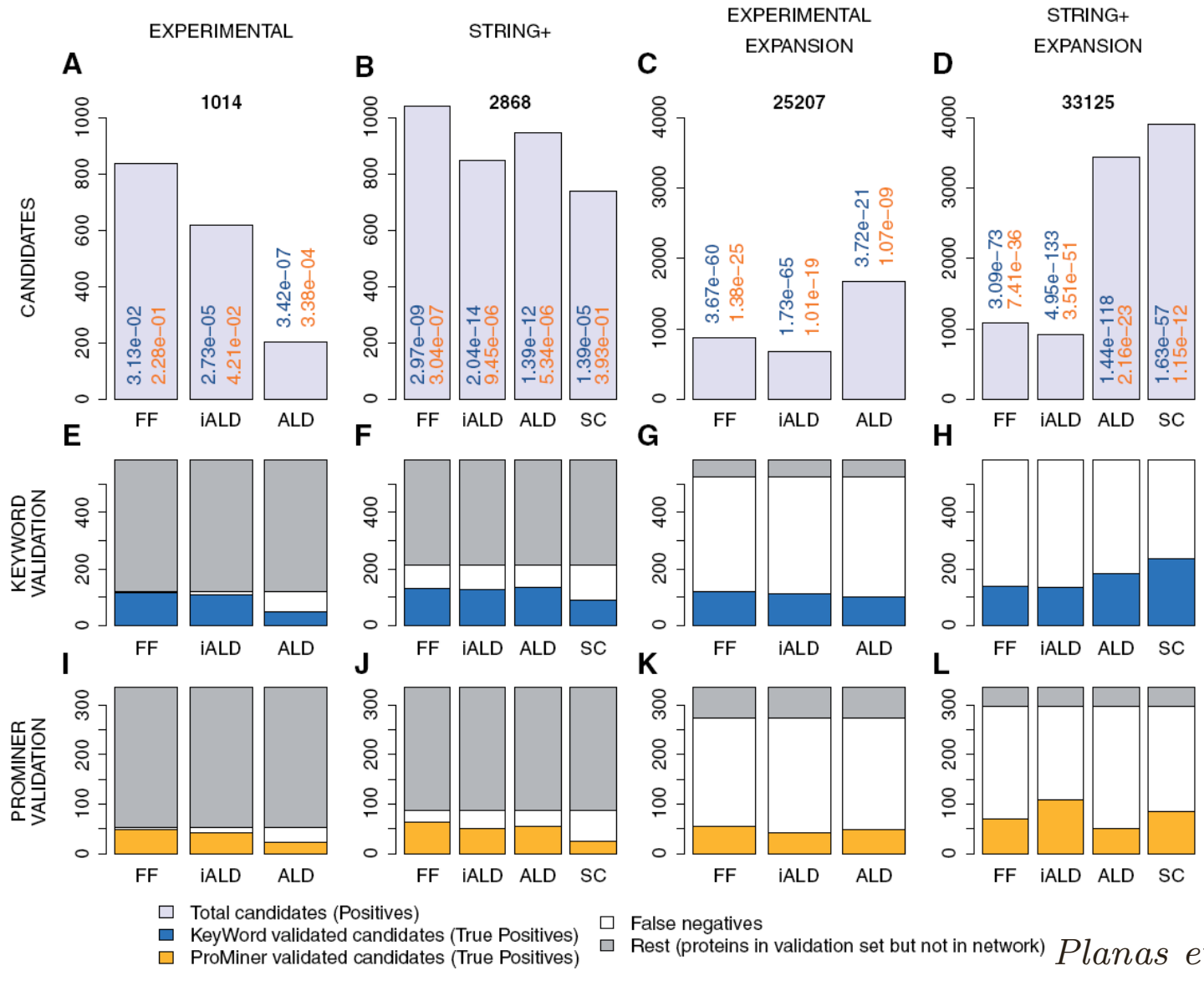
High Scoring Subnetwork Implicated in AD



OMIM & AD-curated ◆ OMIM ● AD-curated

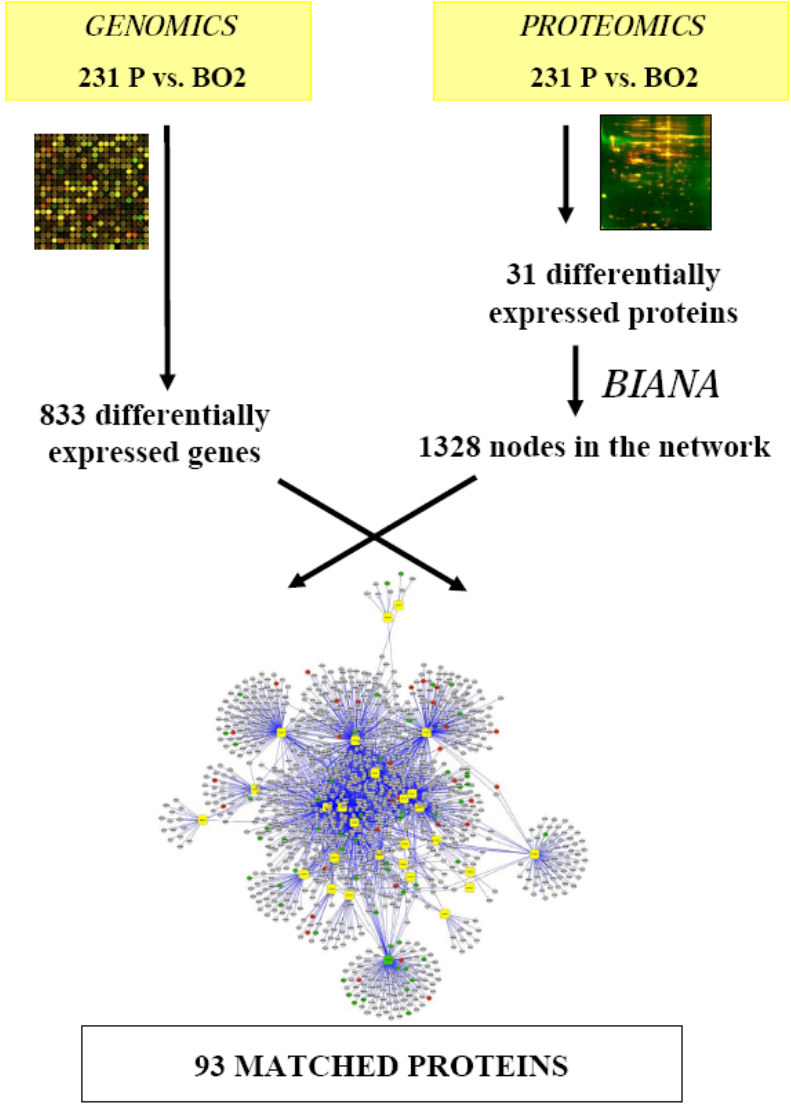
Guney and Oliva, 2012

Extending Apoptosis Pathway Using GUILD

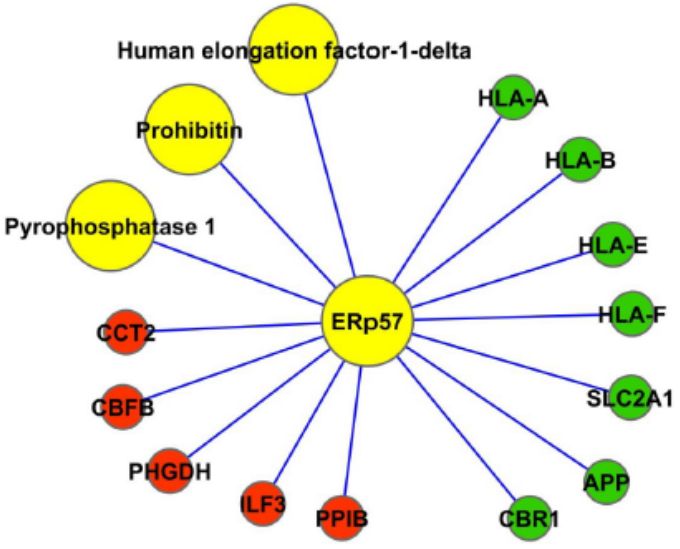


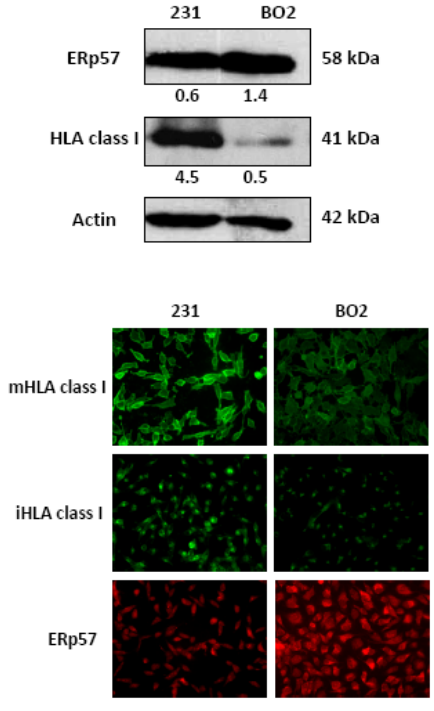
Planas et al., 2012

GUILD implicates ERP57 in Bone Metastatic Breast Cancer



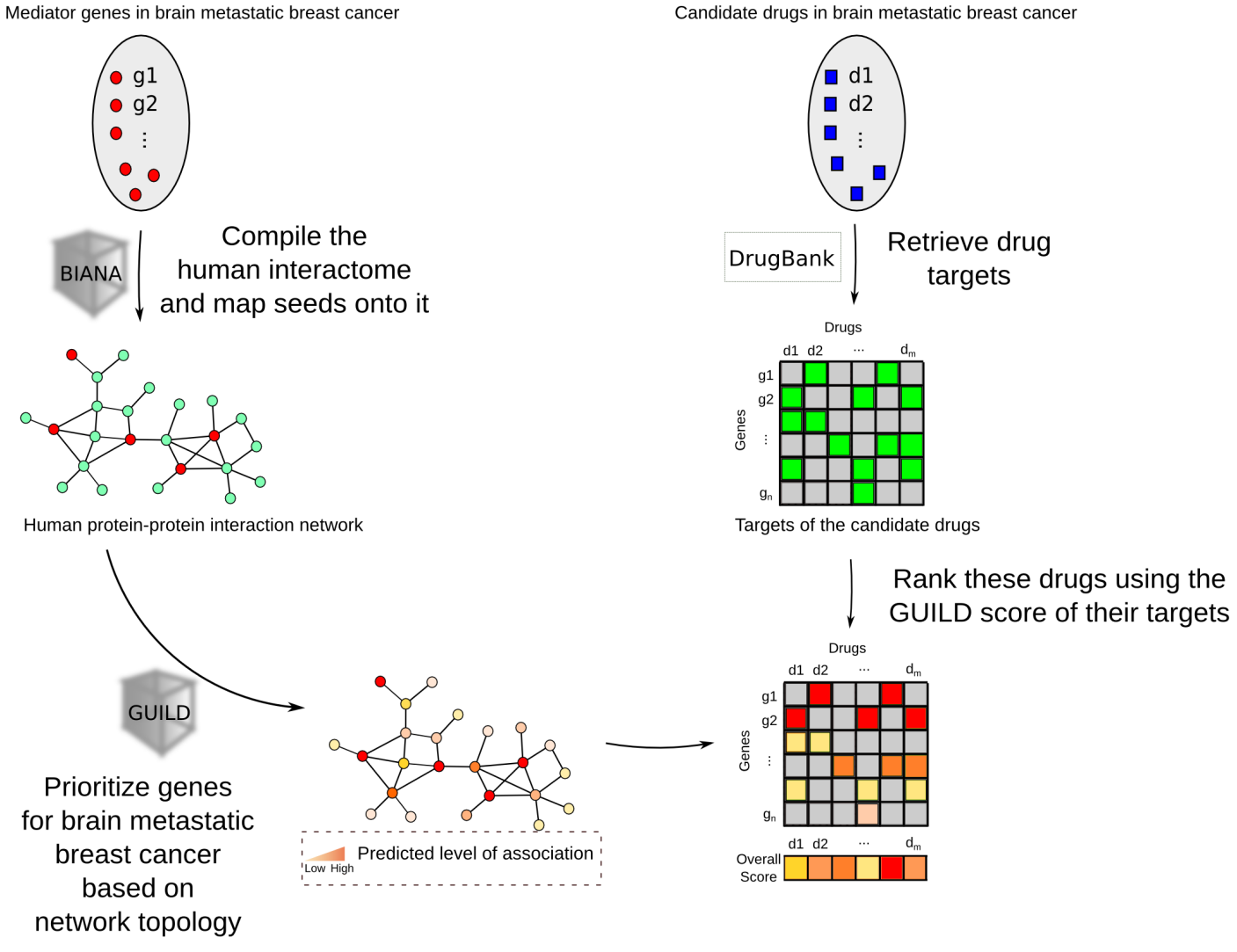
GUILD implicates ERP57 in Bone Metastatic Breast Cancer



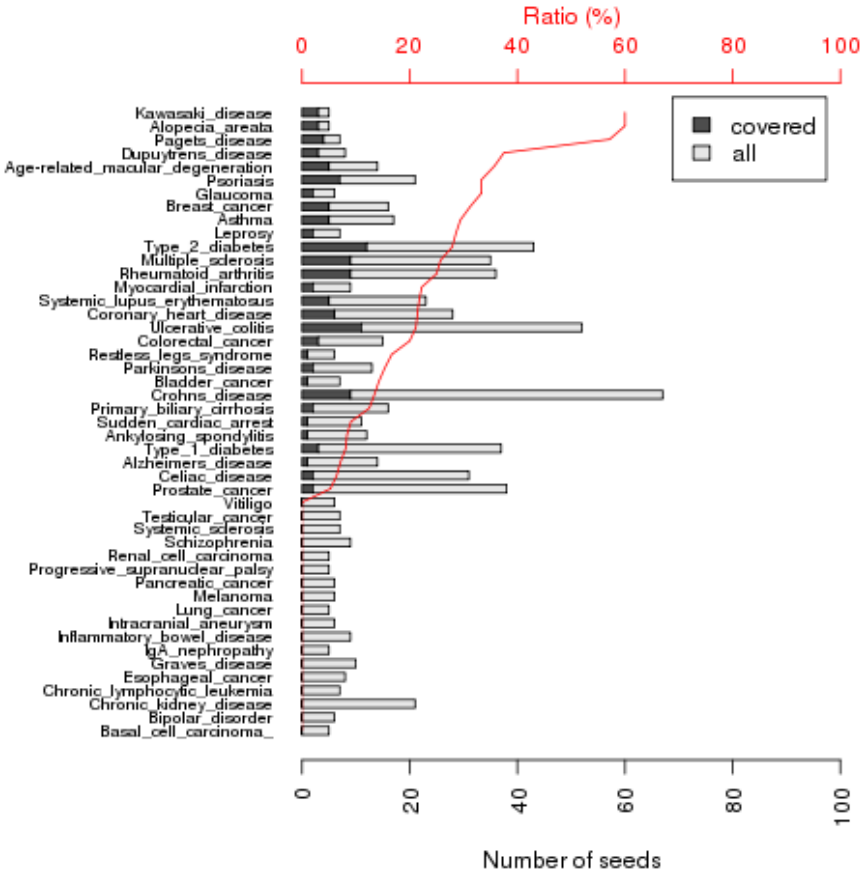


Santana et al., submitted

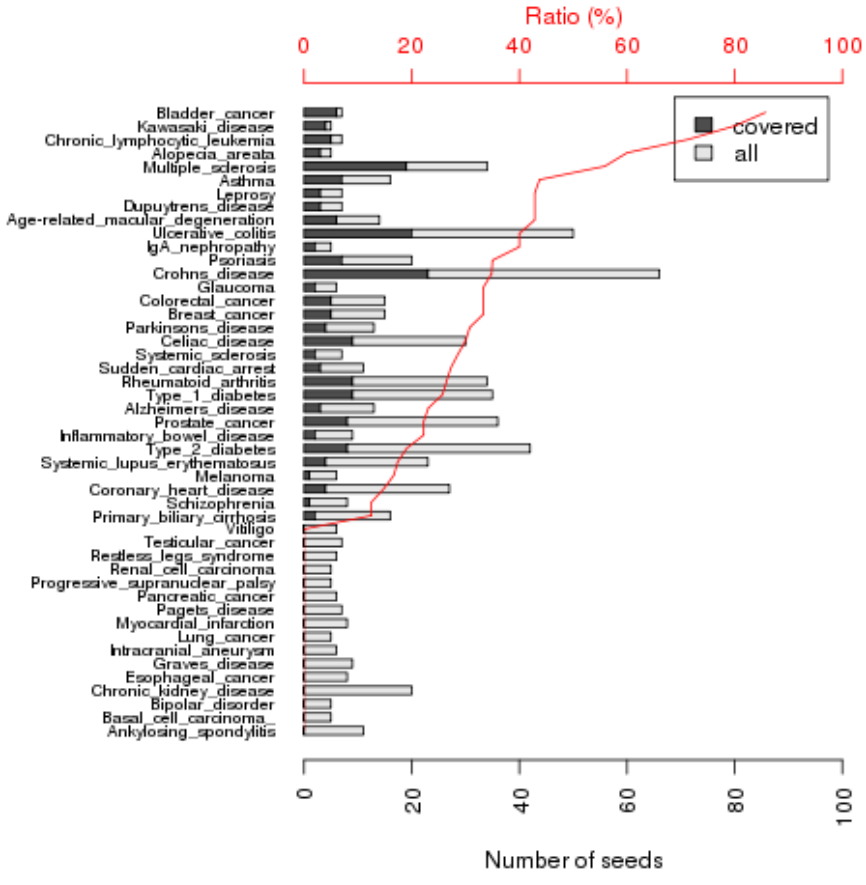
Drug Repurposing using GUILD



Network-based Prioritization of GWAS Data

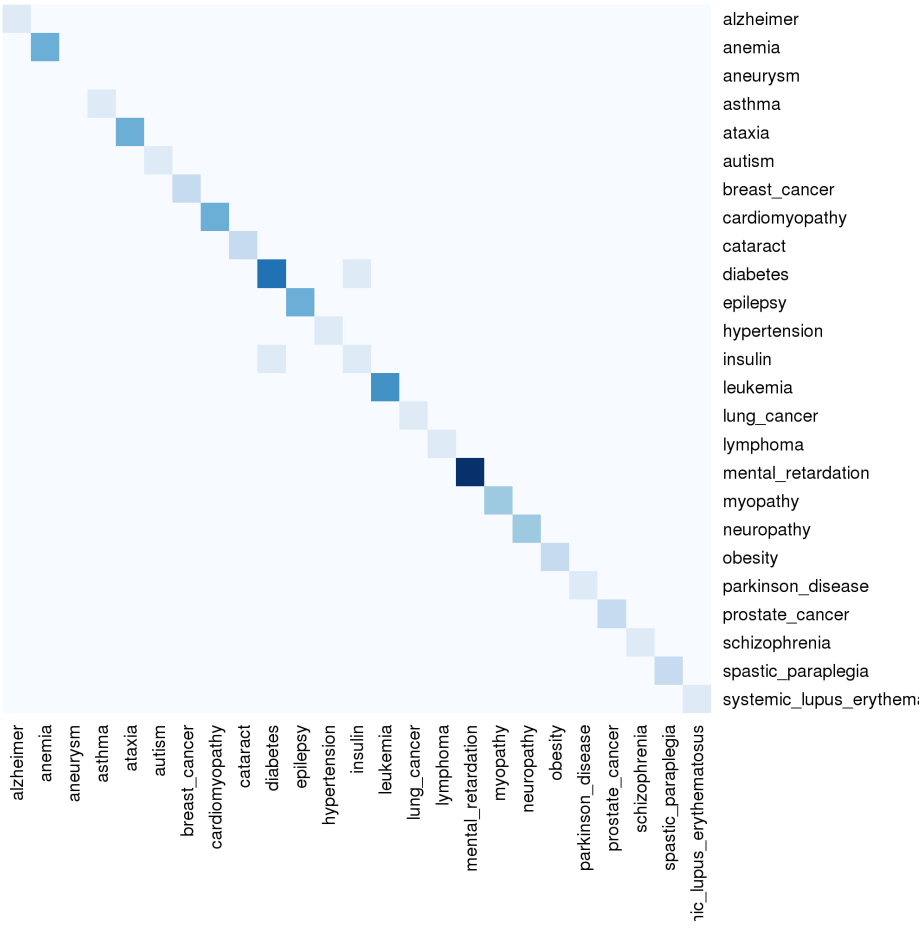


bPPI network

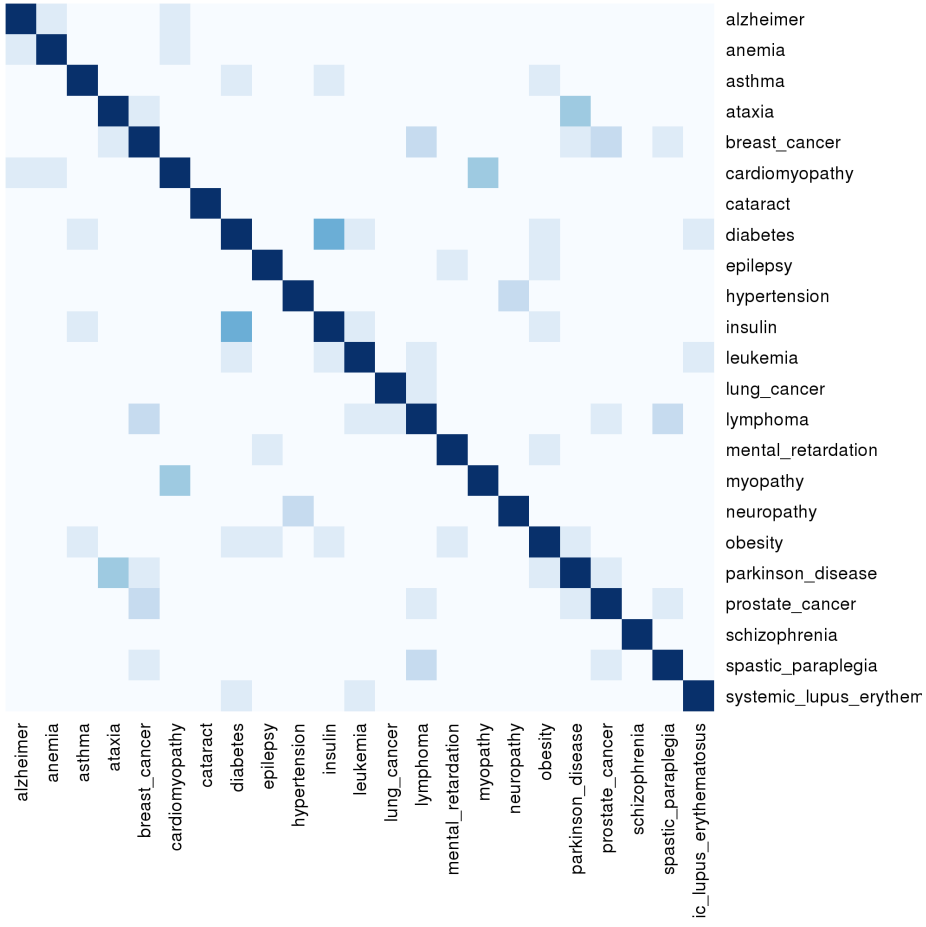


HumanNet

Disease Comorbidity through Common Genes

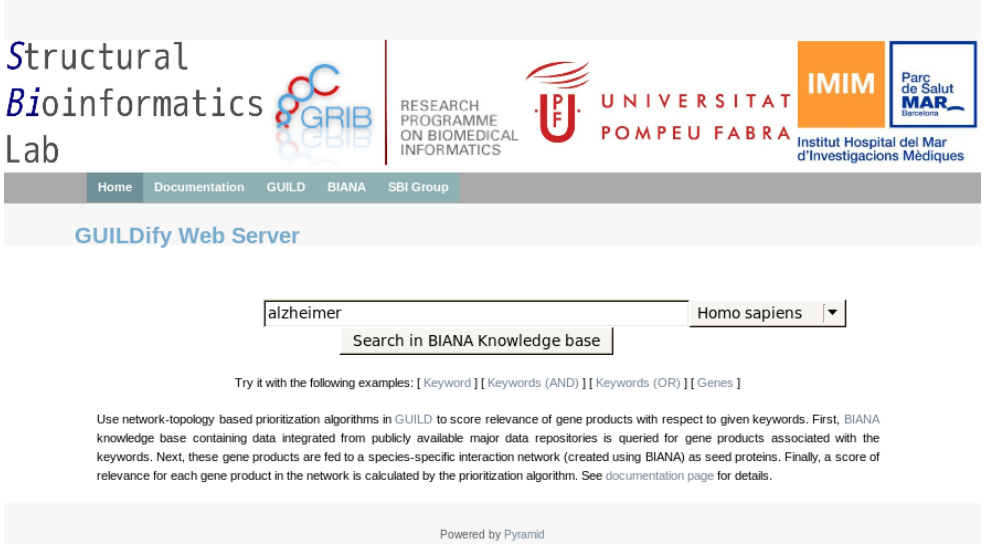


OMIM seeds



High scoring subnetwork

GUILD Web Server



- free text search on UniProt, OMIM databases and GO for a given species
- matching genes are retrieved and used as seeds
- the PPI network is compiled from DIP, HPRD, IntAct, MINT, MPact, BioGRID, BIND

GUILDify

GUILDify Web Server - Matching Proteins

35 BIANA entries are found for the query ***alzheimer*** in *Homo sapiens*. 2 of these entries have no interactions (not in the network).

GUILDify! using selected entries below [Options]

Options

☑ NetScore (Repetition: 3 ▾ Iteration: 2 ▾)

☐ NetZcore (Iteration: 5 ▾)

☐ NetShort

[Hide]

BIANA entries in the network

[Select All / None]

Keep	Gene ID	UniProt ID	Gene Symbol	Description
☑	2	P01023	A2M	{alzheimer disease, susceptibility to}, 104300 (3) omim
☑	26574	Q9NY61	AATF	May function as a general inhibitor of the histone deacetylase HDAC1. Binding to the pocket region of RB1 may displace HDAC1 from RB1/E2F complexes, leading to activation of E2F target genes and cell cycle progression. Conversely, displacement of HDAC1 from SP1 bound to the CDKN1A promoter leads to increased expression of this CDK inhibitor and blocks cell cycle progression. Also antagonizes PAWR mediated induction of aberrant amyloid peptide production in alzheimer disease (presenile and senile dementia), although the molecular basis for this phenomenon has not been described to date. swissprot
☑	1636	P12821	ACE	{alzheimer disease, susceptibility to}, 104300 (3) omim
☑	9507	O75173	ADAMTS4	Cleaves aggrecan, a cartilage proteoglycan, and may be involved in its turnover. May play an important role in the destruction of aggrecan in arthritic diseases. Could also be a critical factor in the exacerbation of neurodegeneration in alzheimer disease. Cleaves aggrecan at the '392-Glu-|-Ala-393' site. swissprot
☑	323	Q92870	APBB2	{alzheimer disease, late-onset}, 104300 (3) omim
☑	348	P02649	APOE	alzheimer disease swissprot Defects in APOE are a cause of hyperlipoproteinemia type 3 (HLPP3) [MIM:107741]; also known as familial dysbetalipoproteinemia. Individuals with HLPP3 are clinically characterized by xanthomas, yellowish lipid deposits in the palmar crease, or less specific on tendons and on elbows. The disorder rarely manifests before the third decade in men. In women, it is usually expressed only after the menopause. The vast majority of the patients are homozygous for APOE*2 alleles. More severe cases of HLPP3 have also been observed in individuals heterozygous for rare APOE variants. The influence of APOE on lipid levels is often suggested to have major implications for the risk of coronary artery disease (CAD). Individuals carrying the common APOE*4 variant are at higher risk of CAD. Genetic variations in APOE are associated with alzheimer disease type 2 (AD2) [MIM:104310]. It is a late-onset neurodegenerative disorder characterized by progressive dementia, loss of cognitive abilities, and deposition of fibrillar amyloid proteins as intraneuronal neurofibrillary tangles, extracellular amyloid plaques and vascular amyloid deposits. The major constituent of these plaques is the neurotoxic amyloid-beta-APP 40-42 peptide (s), derived proteolytically from the transmembrane precursor protein APP by sequential secretase processing. The cytotoxic C-terminal fragments (CTFs) and the caspase-cleaved products such as C31 derived from APP, are also implicated in neuronal death. Note=The APOE*4 allele is genetically associated with the common late onset familial and sporadic forms of alzheimer disease. Risk for AD increased from 20% to 90% and mean age at onset decreased from 84 to 68 years with increasing number of APOE*4 alleles in 42 families with late onset AD. Thus APOE*4 gene dose is a major risk factor for late onset AD and, in these families, homozygosity for APOE*4 was virtually sufficient to cause AD by age 80. The mechanism by which APOE*4 participates in pathogenesis is not known. Defects in

GUILDify

GUILDify Web Server - Results

Calculated scores for proteins in the BIANA ***Homo sapiens*** interactome: [Download all scores] [Download seed proteins]

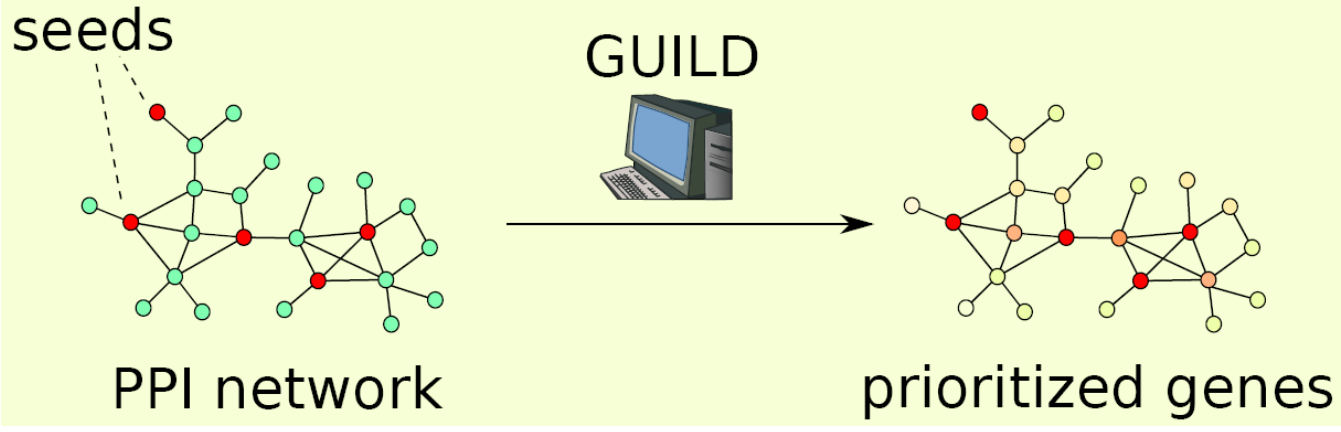
Proteins **21** - **40** of 11238 << Previous Next >>

Gene ID	UniProt ID	Gene Symbol	GUILD Score	Description
1509	P07339	seed: CTSD	0.7507	Cathepsin D Cathepsin D light chain Cathepsin D heavy chain
22943	O94907	seed: DKK1	0.7494	Dickkopf-related protein 1
4137	P10636	seed: MAPT	0.7436	Microtubule-associated protein tau
4353	P05164	seed: MPO	0.7216	Myeloperoxidase Myeloperoxidase 89 kDa myeloperoxidase 84 kDa myeloperoxidase Myeloperoxidase light chain Myeloperoxidase heavy chain
10059	O00429	seed: DNM1L	0.7209	Dynamin-1-like protein
9520	P55786	seed: NPEPPS	0.7116	Puromycin-sensitive aminopeptidase
22883	O94985	seed: CLSTN1	0.7097	Calsyntenin-1 Soluble Alc-alpha CTF1-alpha
3077	Q30201	seed: HFE	0.7075	Hereditary hemochromatosis protein
26574	Q9NY61	seed: AATF	0.7056	Protein AATF
27122	Q9UBP4	seed: DKK3	0.7002	Dickkopf-related protein 3
55103	Q86X27	seed: RALGPS2	0.6962	Ras-specific guanine nucleotide-releasing factor RalGPS2
2186	Q12830	seed: BPTF	0.6880	Nucleosome-remodeling factor subunit BPTF
4973	P78380	seed: OLR1	0.6875	Oxidized low-density lipoprotein receptor 1 Oxidized low-density lipoprotein receptor 1, soluble form
5408	P54317	PNLIPRP2	0.6623	Pancreatic lipase-related protein 2
83999	Q96MU8	KREMEN1	0.6362	Kremen protein 1
1208	P04118	CLPS	0.5428	Colipase
8513	P07098	DKFZp666P126, LIPF	0.5245	Gastric triacylglycerol lipase
10102	P43897	TSFM	0.4981	Elongation factor Ts, mitochondrial
339175	Q96IZ6	METTL2A	0.4968	Methyltransferase-like protein 2A
55798	Q6P1Q9	METTL2B	0.4968	Methyltransferase-like protein 2B

Outline

Outline

Investigating Robustness of Prioritization Methods

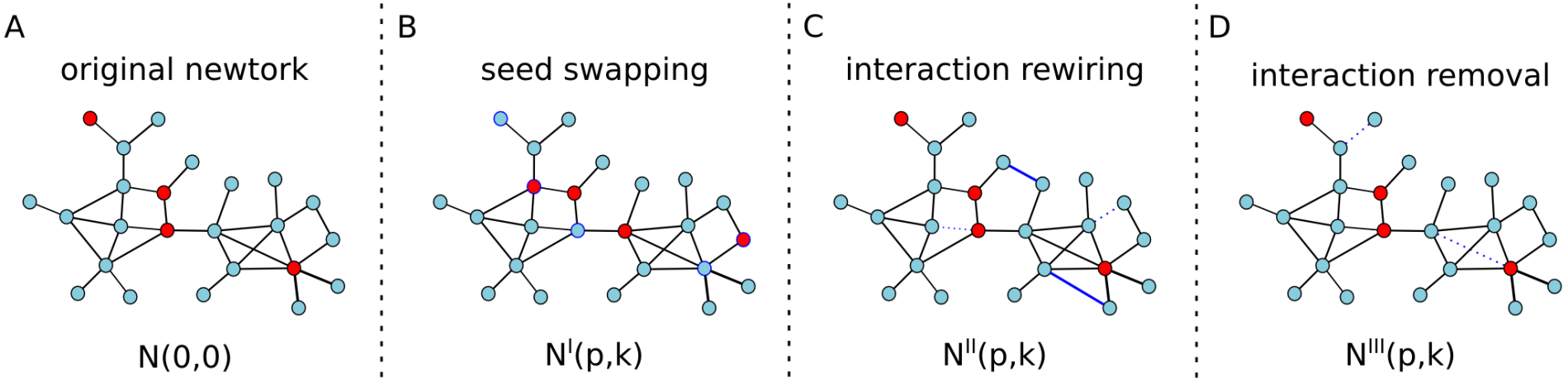


Evaluation of network-based prioritization methods upon perturbations

- five-fold AUC performance
- bPPI network
- 19 disease phenotypes in OMIM → seeds

Investigating Robustness of Prioritization Methods

Three types of perturbation (at various levels; 10% to 80%)



$$AUC^D_{perturbed}(p) = \frac{1}{n}\sum_{k=1}^{n} AUC^D(N(p,k))$$

Identification of Disease Modules

image from stanford.edu/~thkim7

- **high-scoring subnetwork**: proteins at top 5% and their interactions
- **disease modules**: modules in the high-scoring subnetwork identified by MCL *(Enright et al., 2002)* that contain at least 2 seeds

Functional Enrichment Analysis

- **GO term enrichment**: FuncAssoc2 web service *(Berriz et al., 2009)*
  - all the proteins in the network used as background
  - proteins mapped to gene symbols using UniProt
  - consider GO terms with *adjusted p-value* ≤ 0.05
- **seed GO terms**: GO terms enriched among seeds of the disease
- **non-redundant GO terms**: GO terms at least 90% semantically similar to another GO term are removed using GOSemSim package *(Yu et al., 2010)*

Differential Network Analysis

- **differential network**: get interactions from the networks 80% of whose interactions were randomly removed
  - interactions common to two best performing (in terms of AUC) randomly perturbed networks
  - excluding interactions common to two worst performing randomly perturbed networks

Investigating Network-centric Tolerance of Diseases

Network-centric tolerance of a disease

Incapacity to effect the disease with interventions in the underlying network

- Intervention: perturbations introduced in the interaction network
- Evaluation: recoverability of disease genes through prioritization

Investigating Network-centric Tolerance of Diseases

Network-centric tolerance of a disease

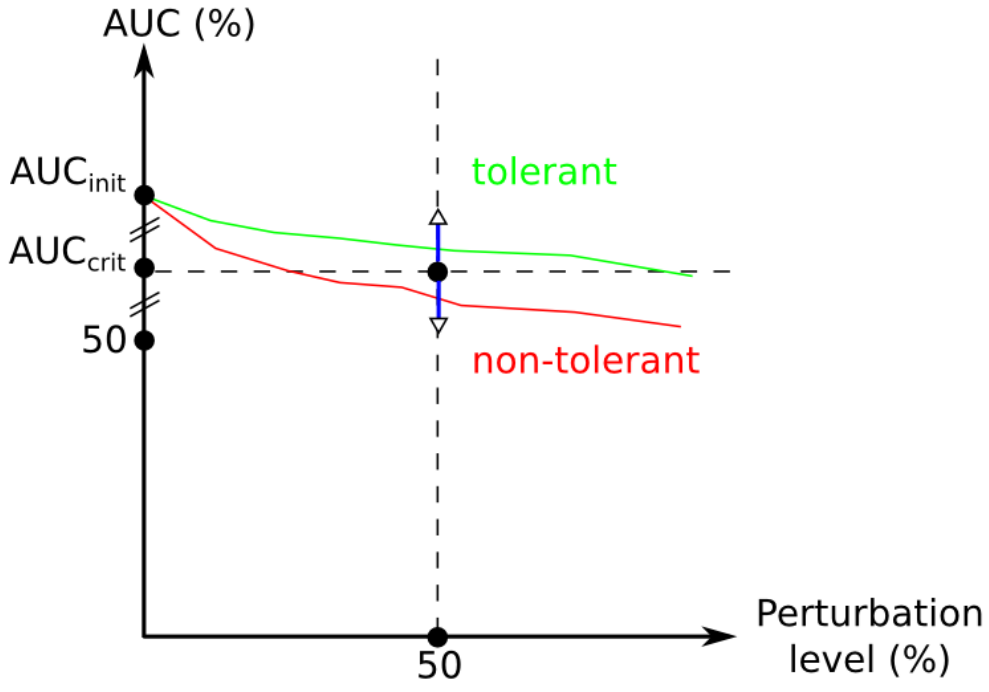
Incapacity to effect the disease with interventions in the underlying network

- Intervention: perturbations introduced in the interaction network
- Evaluation: recoverability of disease genes through prioritization

Categorizing diseases w.r.t. network-centric tolerance

Whether the disease can tolerate perturbing 50% of the interactions

Investigating Network-centric Tolerance of Diseases



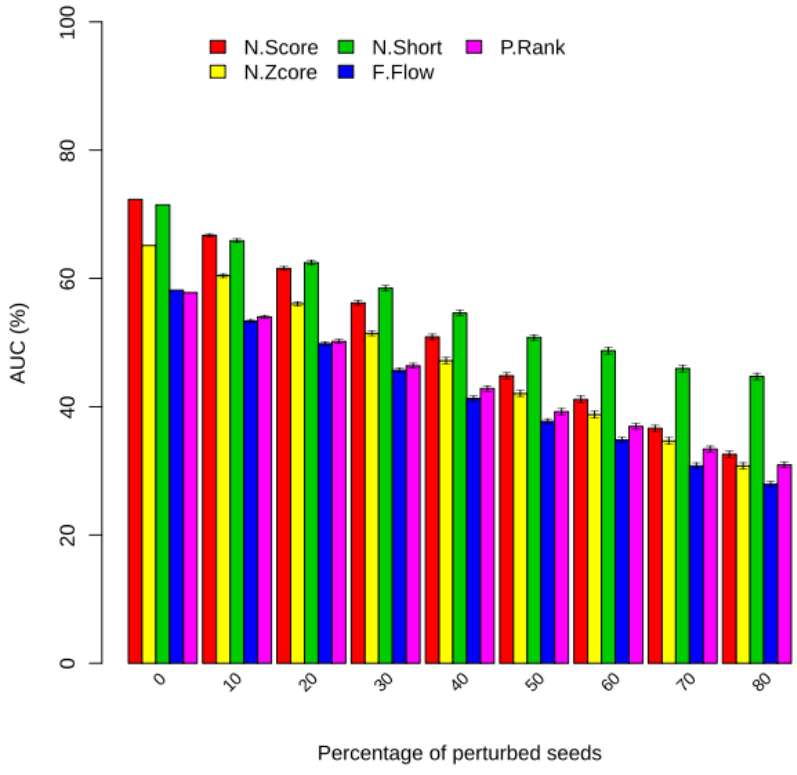
$$AUC^D_{tolerance} = \frac{AUC_{perturbed}(0)^D - 0.5}{2}$$

$$AUC^D_{critical} = AUC^D_{perturbed}(0) - AUC^D_{tolarence}$$

$$category(D) = \begin{cases} tolerant, & if\ AUC^D_{perturbed}(50) > AUC^D_{critical} \\ non\text{-}tolerant, & otherwise \end{cases}$$

Outline

Dependence on the Seed Set

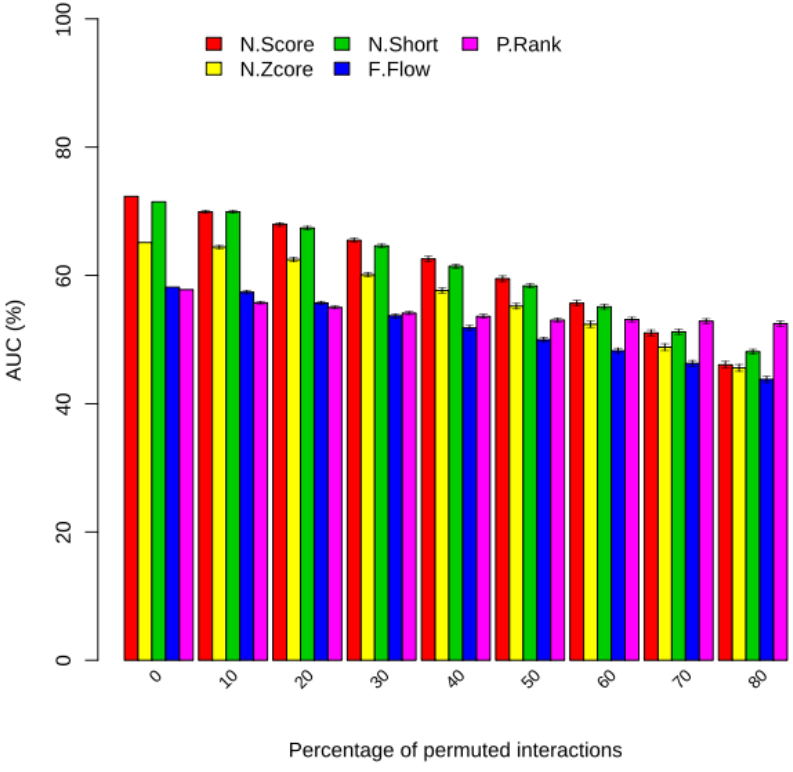


NetScore
NetZcore
NetShort
Functional Flow
Page Rank

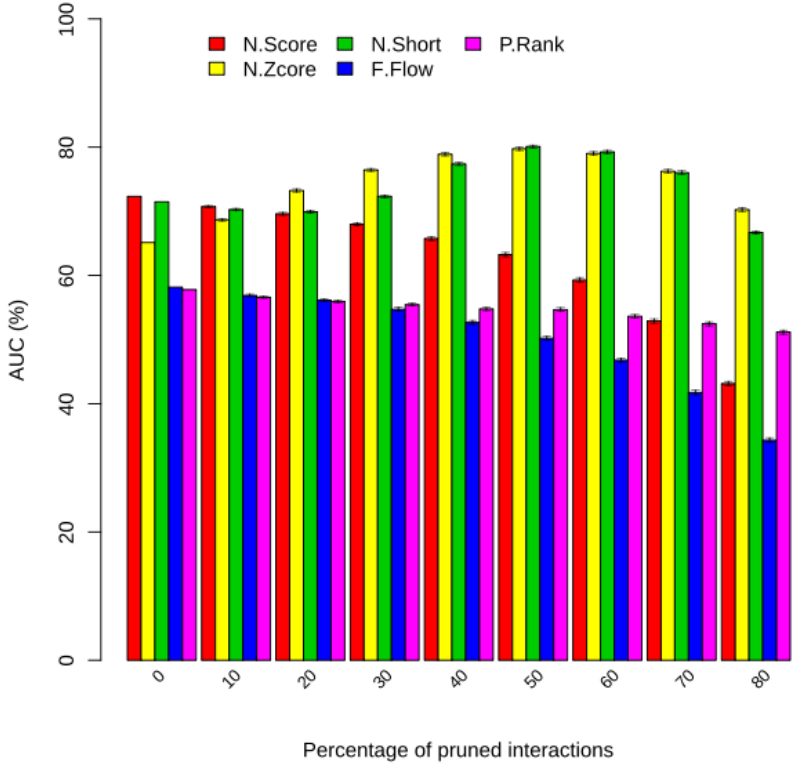
seed swapping

Guney and Oliva, submitted

Dependence Against Underlying Network



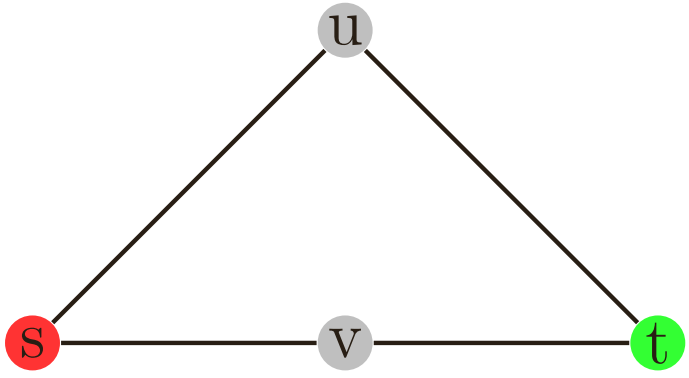
interaction rewiring



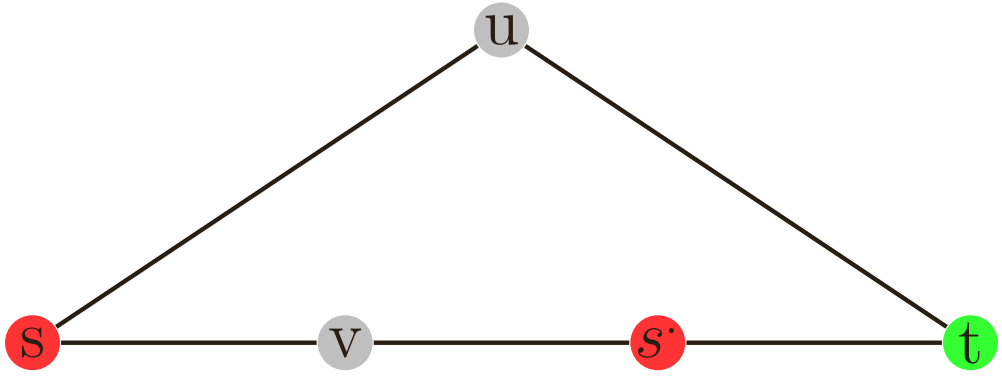
interaction removal

Guney and Oliva, submitted

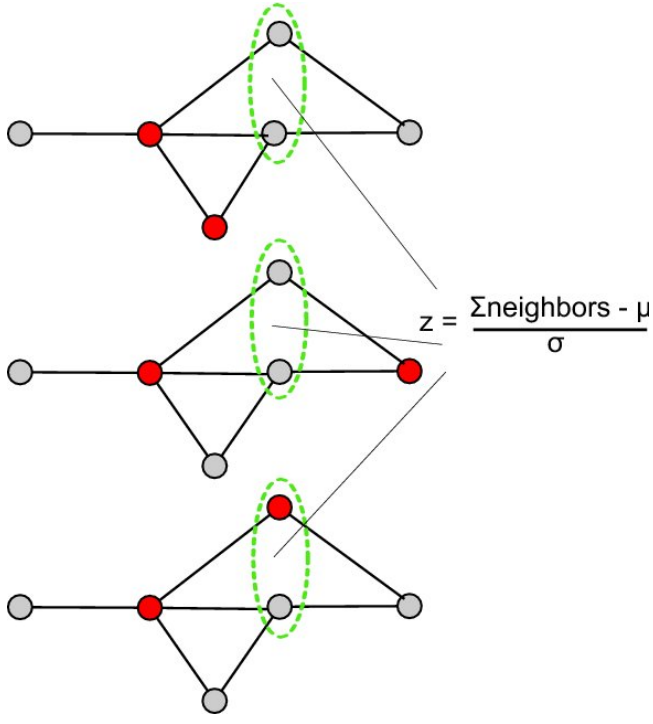
Existance of Alternative Routes Connecting Disease-genes



NetScore

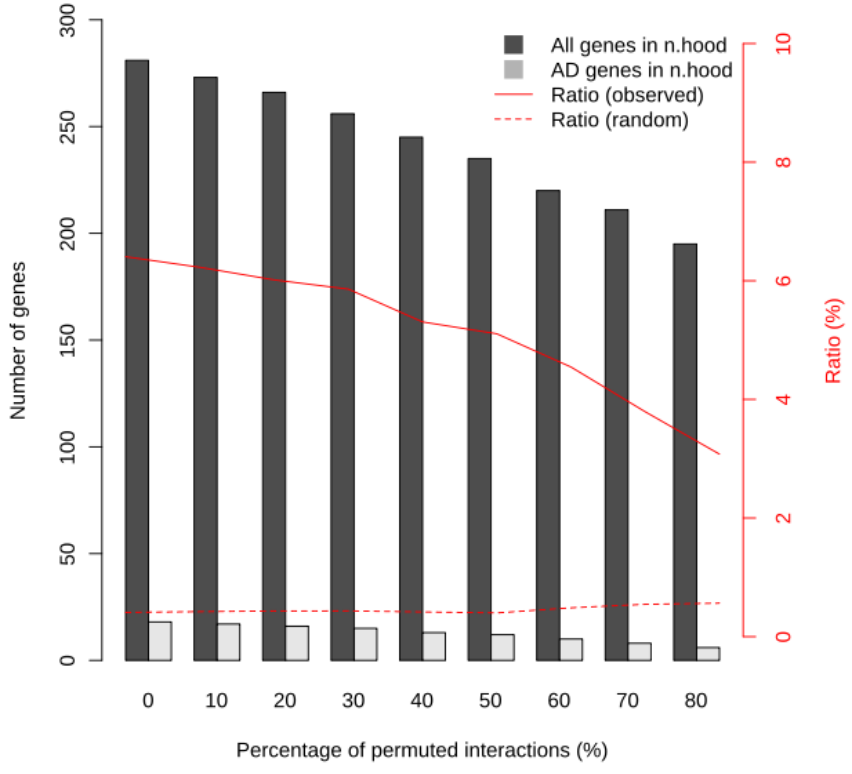


NetShort

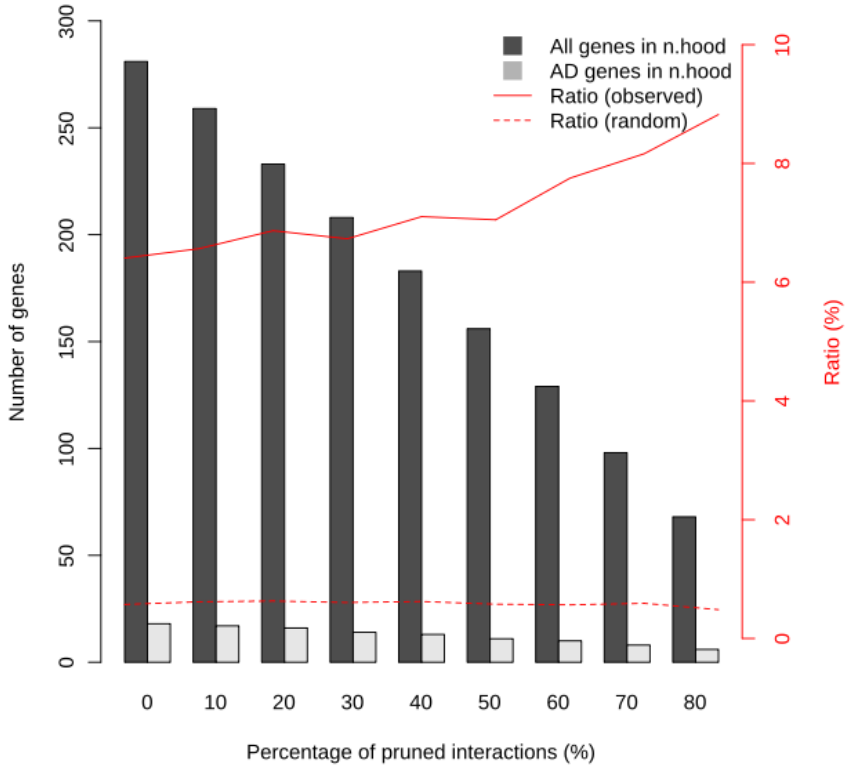


NetZcore

The Case of Alzheimer's Disease



interaction rewiring



interaction removal

AD data from Krauthammer et al., 2004

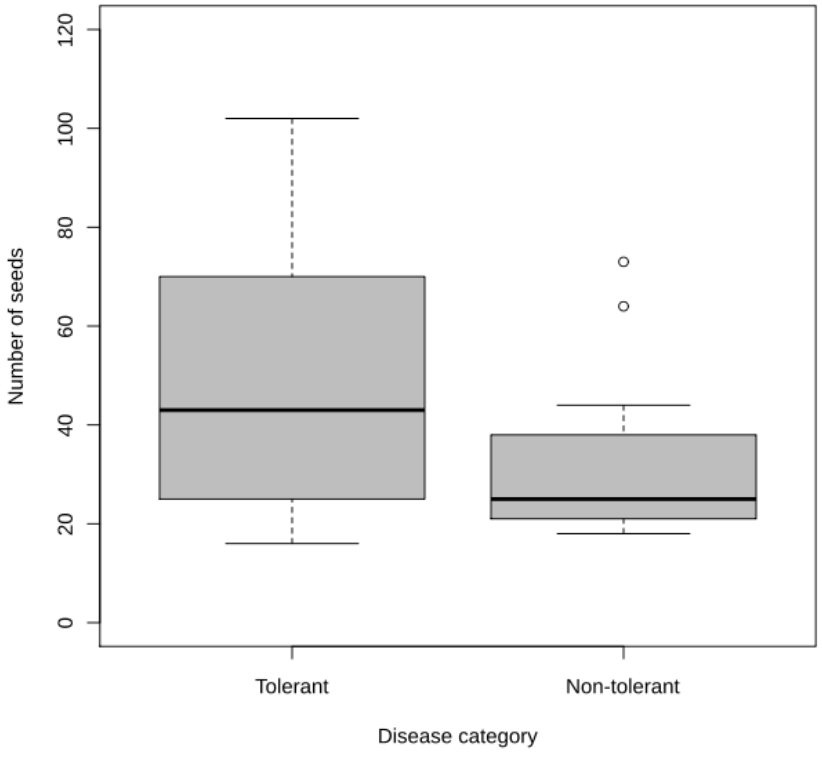
Categorizing Network-centric Tolerance of Diseases

NetScore

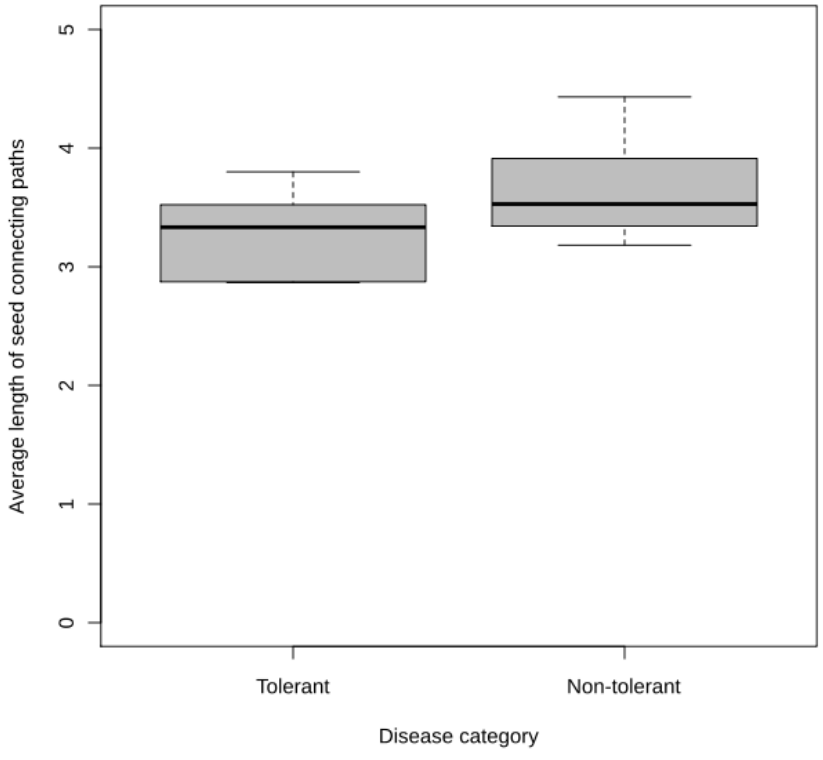
- good overall AUC performance
- not robust to perturbations in the network

Pathophenotype	AUC (%)				Pathophenotype	AUC (%)			
	org.	crit.	perm.	del.		org.	crit.	perm.	del.
alzheimer	78.3	64.2	62.5	62.8	lung cancer	85.0	67.5	65.8	68.4
anemia	70.3	60.2	56.4	57.9	lymphoma	79.7	64.9	62.3	71.8
ataxia	62.6	56.3	54.2	53.8	mental retardation	56.3	53.2	46.6	45.3
breast cancer	76.7	63.4	70.7	75.8	myopathy	86.0	68.0	67.3	72.0
cardiomyopathy	69.5	59.8	65.0	70.5	**obesity**	72.0	61.0	67.4	70.4
cataract	72.0	61.0	53.9	52.8	**parkinson disease**	80.0	65.0	70.9	78.5
diabetes	61.4	55.7	58.4	63.4	prostate cancer	68.0	59.0	52.7	62.7
epilepsy	62.1	56.1	47.4	47.4	schizophrenia	53.3	51.7	40.9	42.1
hypertension	70.0	60.0	47.7	51.8	systemic lupus erythematosus	86.3	68.2	64.2	72.7
leukemia	84.6	67.3	75.8	81.6					

Comparison Between Diseases

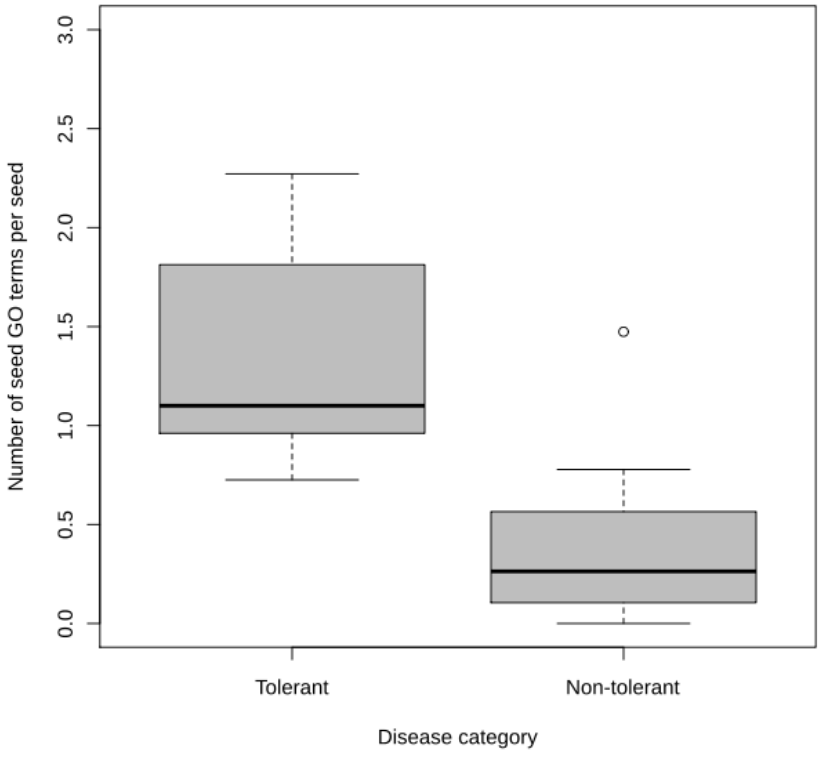


# of seeds
($P = 0.27$)

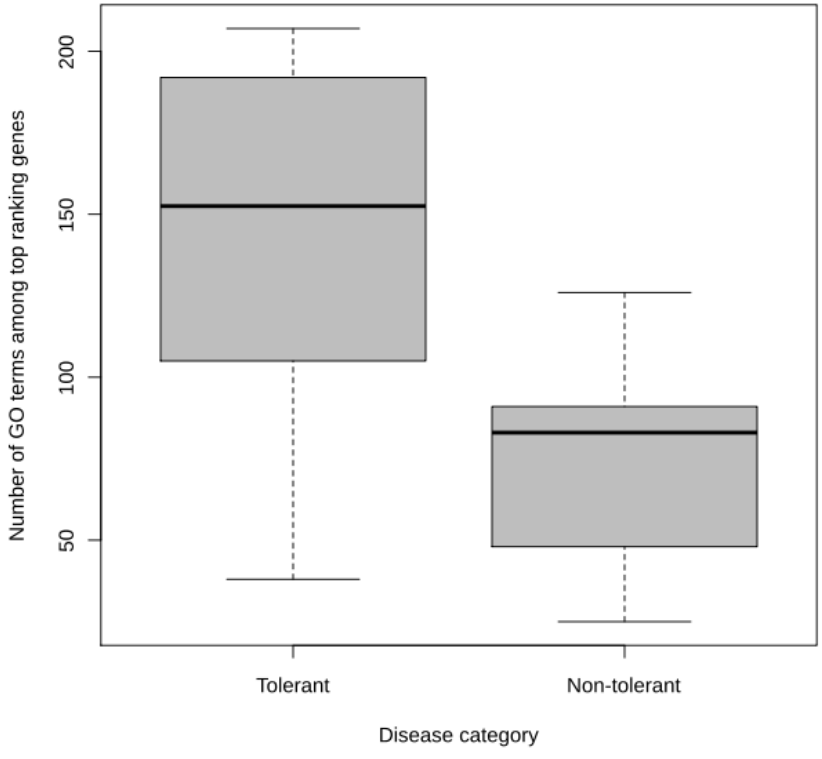


path length between seeds
($P = 0.13$)

Comparison Between Diseases

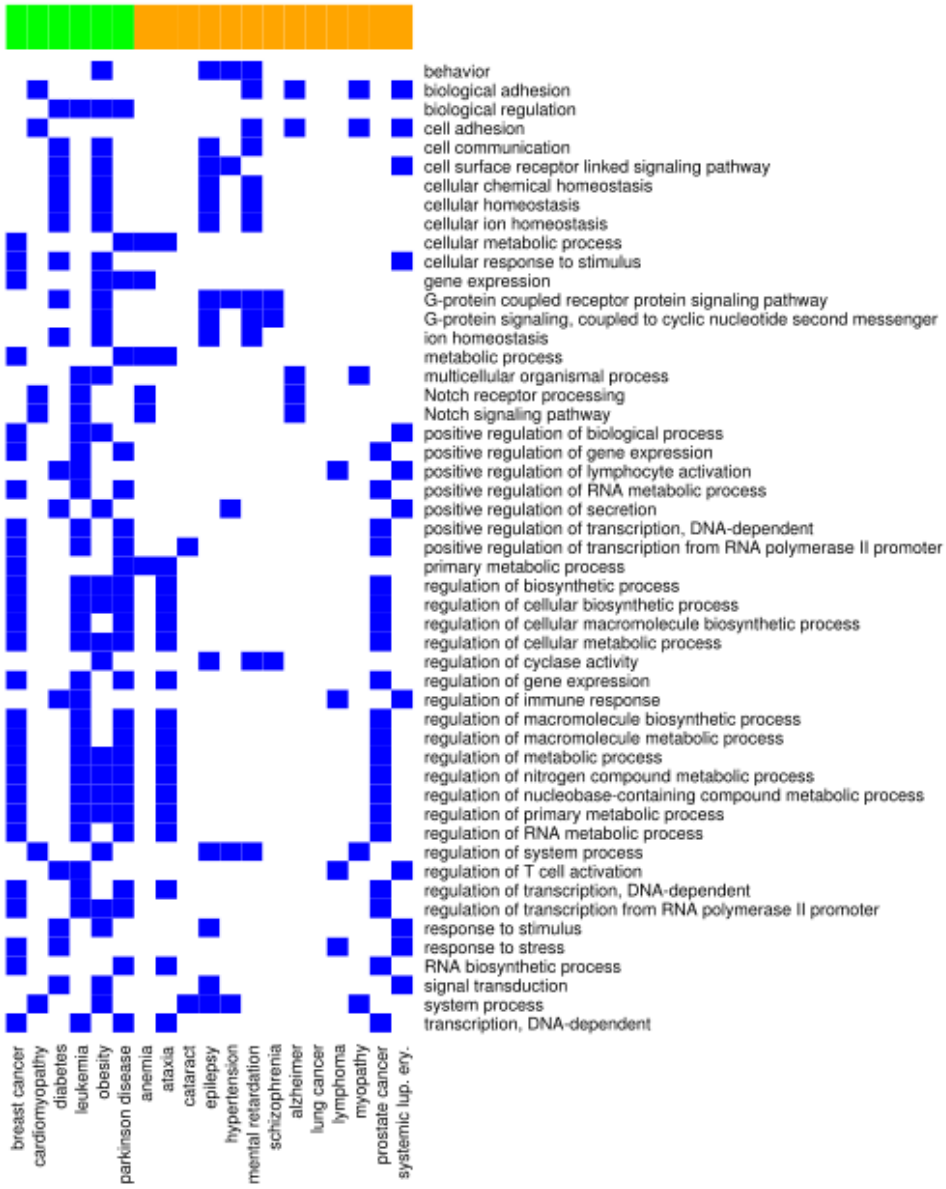


# of seed GO terms per seed
($P = 3.29e^{-3}$)



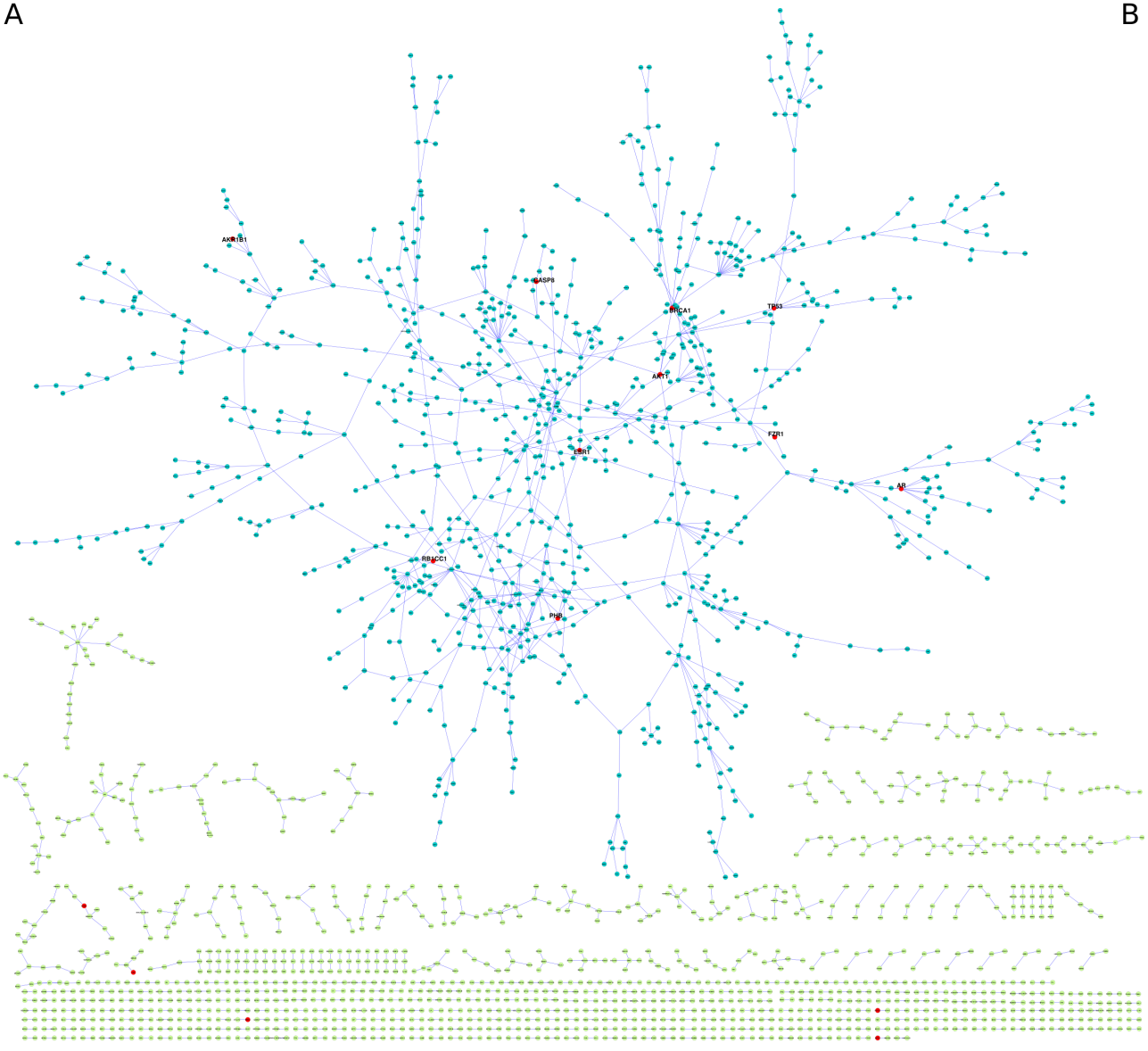
# of GO terms in the high-scoring subnetwork
($P = 0.03$)

GO Terms Common to at Least 4 Diseases

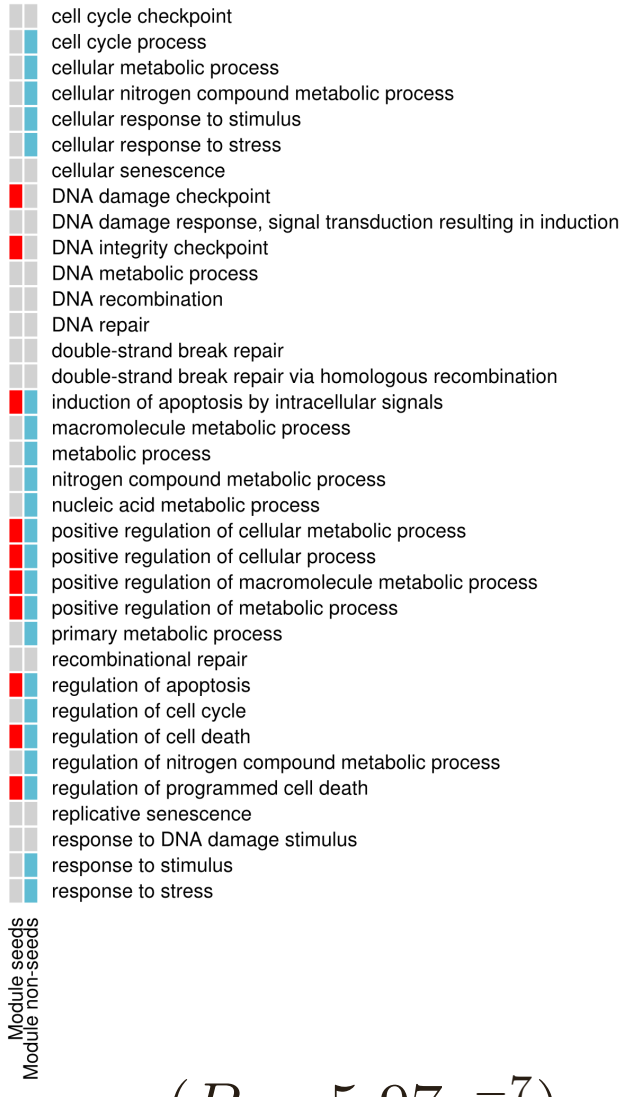


Differential Network in Breast Cancer

A



B



$(P = 5.97e^{-7})$

Outline

Conclusions

- GUILD exploits the global topology to rank all the genes in the network
- Prioritization performance is correlated with the connectedness of the seeds
- Existence of alternative paths connecting seeds give rise to robustness of the prioritization methods
- Functional diversity plays an important role in defining tolerance against perturbations in the network

Acknowledgements

UPF
Baldo Oliva
Javier Garcia
Joan Planas

Koc U.
Attila Gursoy
Ozlem Keskin
Nurcan Tuncbag

Collaborations
Angels Sierra, IDIBELL
Arcadi Navarro, UPF
Janusz Dutkowski, USCD
David Ochoa, CNB-CSIC

sbi.imim.es

Structural Bioinformatics Lab
GRIB
RESEARCH PROGRAMME ON BIOMEDICAL INFORMATICS
UNIVERSITAT POMPEU FABRA
IMIM
Parc de Salut MAR Barcelona
Institut Hospital del Mar d'Investigacions Mèdiques

Agència de Gestió d'Ajuts Universitaris i de Recerca
MINISTERIO DE EDUCACIÓN Y CIENCIA

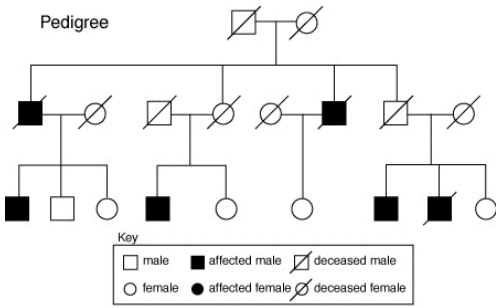
Questions & Comments

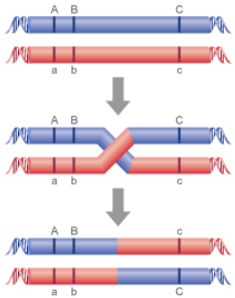
Thank you for your attention!

Appendix

- Supplementary Background
- Supplementary Methods
- Supplementary Results
- Bonus Projects
- Future Directions

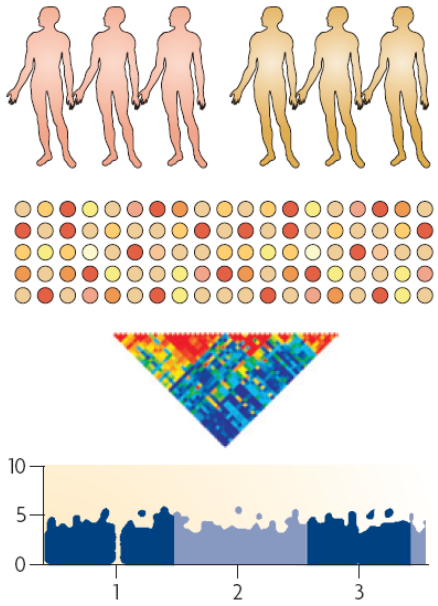
Associating Human Genetic Variants with Diseases





linkage analysis

Broeckel and Schork, 2004, Carlson et al., 2004



association studies

Hirschhorn et al., 2002, Altshuler et al., 2008, McCarthy et al., 2008

images from Kingsmore et al., 2008, genome.wellcome.ac.uk and wikipedia.com

Associating Human Genetic Variants with Diseases

- confounding genetic factors
- problems with reproducibility
- small effect sizes
- variants not necessarily correspond coding sequence

Missing heritability

Frazer et al., 2009

Disease-gene Association Databases

Database	URL
Short variations	
1000 Genomes	`www.1000genomes.org`
dbSNP	`www.ncbi.nlm.nih.gov/projects/SNP`
HapMap	`www.hapmap.org`
Structural variations	
dbVar	`www.ncbi.nlm.nih.gov/dbvar`
DGV	`projects.tcag.ca/variation`
DGVa	`www.ebi.ac.uk/dgva`
General variants associated with phenotypes	
HGMD	`www.hgmd.org`
OMIM	`www.omim.org`
SwissVar	`swissvar.expasy.org`
GWAS and other association studies	
dbGaP	`www.ncbi.nlm.nih.gov/gap`
EGA	`www.ebi.ac.uk/ega`
GAD	`geneticassociationdb.nih.gov`
NHGRI GWAS Catalog	`www.genome.gov/gwastudies`
Cancer genes and variants	
ICGC	`www.icgc.org`
COSMIC	`sanger.ac.uk/genetics/CGP/cosmic`
Cancer Gene Census	`sanger.ac.uk/genetics/CGP/Census`
Cancer Gene Index	`ncicb.nci.nih.gov/NCICB/projects/cgdcp`
TCGA	`cancergenome.nih.gov`

Disease-gene Association Databases

Database	URL
Pharmacogenomic genes and variants	
DrugBank	drugbank.ca
PharmGKB	www.pharmgkb.org
CTD	ctdbase.org
KEGG	www.genome.jp/kegg
Crowdsourced genes and variants	
Gene Wiki	en.wikipedia.org/wiki/Portal:Gene_Wiki
SNPedia	www.snpedia.com
WikiGenes	www.wikigenes.org
Computationally-derived / meta databases	
GeneCards	www.genecards.org
PhenoGO	www.phenogo.org
PhenomicDB	www.phenomicdb.de
DisGeNet	ibi.imim.es/DisGeNET/DisGeNETweb.html

Capriotti et al., 2012

Disease-gene Prioritization

- **Sequence, structure**
  - sequence conservation
  - coding region length
  - structural domains
  - sequence motifs
  - chromosomal location
  - protein localization
- **Non-human data (Orthology)**
  - information from other species
- **Ontologies**
  - gene, disease, phenotype and anatomic ontologies

Disease-gene Prioritization

- **Literature**
  - text-mining
- **Mutations**
  - existing knowledge / predictions on functional and structural effects of SNPs
- **Pathway involvement**
  - pathways
  - experimental / predicted PPIs
  - functional association links such as gene co-expression

Existing Approaches in Disease-gene Prioritization

Method	URL	Description
aGeneApart	www.esat.kuleuven.be/ageneapart	L
BITOLA	ibmi.mf.uni-lj.si/bitola	L
CAESAR	polaris.med.unc.edu/projects/caesar	ESPNOML
CANDID	dsgweb.wustl.edu/hutz/candid.html	ESPNL
DADA	compbio.case.edu/dada	PL
DomainRBF	bioinfo.au.tsinghua.edu.cn/domainRBF/gene	SOML
ENDEAVOR	www.esat.kuleuven.be/endeavour	ESPNOL
G2D	www.ogic.ca/projects/g2d_2	ESPOL
GeneDistiller	www.genedistiller.org	ESPNOL
GeneProspector	www.hugenavigator.net	SNML
GeneSeeker	www.cmbi.kun.nl/GeneSeeker	NL
GeneWanderer	compbio.charite.de/genewanderer/GeneWanderer	PNML
Genie	cbdm.mdc-berlin.de/tools/genie	ESPNL
Gentrepid	www.gentrepid.org	ESPL
MedSim	www.funsimmat.de	SPNOL

Existing Approaches in Disease-gene Prioritization

Method	URL	Description
MimMiner	www.cmbi.ru.nl/MimMiner	SL
PGMapper	www.genediscovery.org/pgmapper	ESPL
PhenoPred	www.phenopred.org	SPO
PINTA	www.esat.kuleuven.be/pinta	EP
PRINCE	www.cs.tau.ac.il/~bnet/software/PrincePlugin	EP
PolySearch	wishart.biology.ualberta.ca/polysearch	L
PosMed	omicspace.riken.jp/PosMed	L
PROSPECTR	www.genetics.med.ed.ac.uk/prospectr	SNML
SNPs3D	www.snps3d.org	SPNOML
SUSPECTS	www.genetics.med.ed.ac.uk/suspects	ESPNML
ToppGene	toppgene.cchmc.org	ESPNOL
TOM	www-micrel.deis.unibo.it/~tom	EP
VAAST	www.yandell-lab.org/software/vaast.html	EM

Capriotti et al., 2012

Network-based Prioritization of GWAS Data

Common drawbacks in GWAS;

- most SNPs are not causal variants
- rare variants are typically missed

Mapping SNPs to the genes (˜functional units) and apply network-based prioritization methods

- *Lee et al. 2011*, Chron's disease and type 2 diabetes
- *Akula et al. 2011*, height and Chron's disease

Trade-off Between Robustness and Fragility in Biological Systems

Biological systems are robust, however

- prone to be disrupted by certain types of rare but specialized perturbations

Trade-off Between Robustness and Fragility in Biological Systems

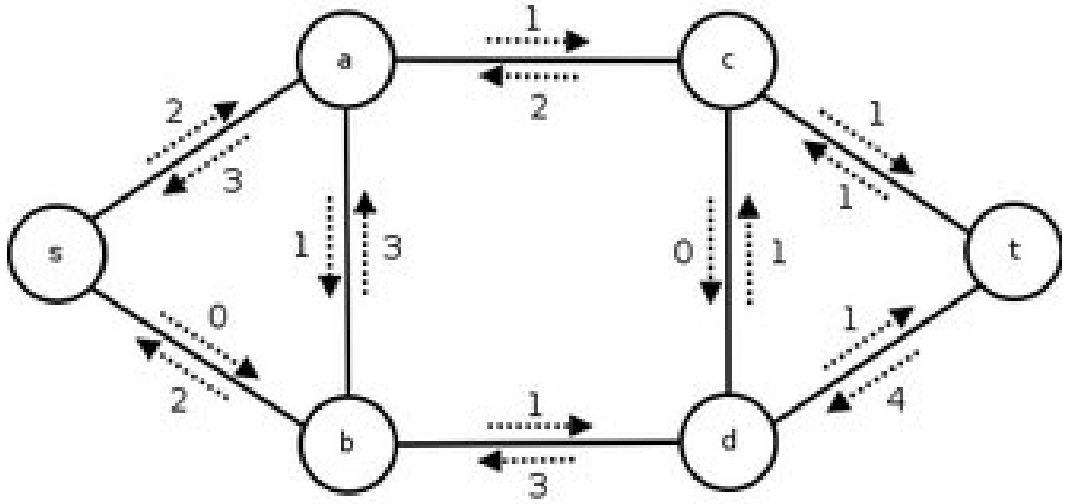
Biological systems are robust, however

- prone to be disrupted by certain types of rare but specialized perturbations
- these perturbations leverage the fragility of the system

Carlson and Doyle, 2002, Kitano, 2004

Back

Functional Flow



- Propagates annotation over the edges of the network from higher scored nodes towards lower scored nodes

Nabieva et al., 2005, Bioinformatics

Random walk with restart

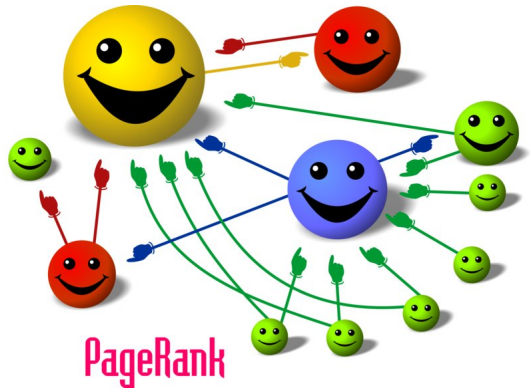
- simulates a random surfer model on the network where number of links a node has defines the transition (surfing) probability to that node:

$$p_{t+1} = (1 - r) * W p_t + r * p_0$$

where p_t is a vector where each element i holds the probability of being at node i of the network at time step t, W is the column normalized adjacency matrix and r is the restart probability.

Kohler et al., 2008, Am. J. Hum. Genet.

PageRank with priors



- similar to random walk where a random surfer is more likely to end up in initially relevant nodes:

$$PR_{t+1}(u) = (1-d) * PR_0(u) + d \sum_{v \in Neighbors(u)} \frac{PR_t(v)}{degree(v)}$$

where u is current node in consideration, v is a node that gives link to u, d is *damping factor*

Chen et al., 2009, BMC Bioinformatics

Network propagation

- modifies random walk with restart algorithm such that the link weight is normalized not only by number of outgoing edges but also by number of incoming edges:

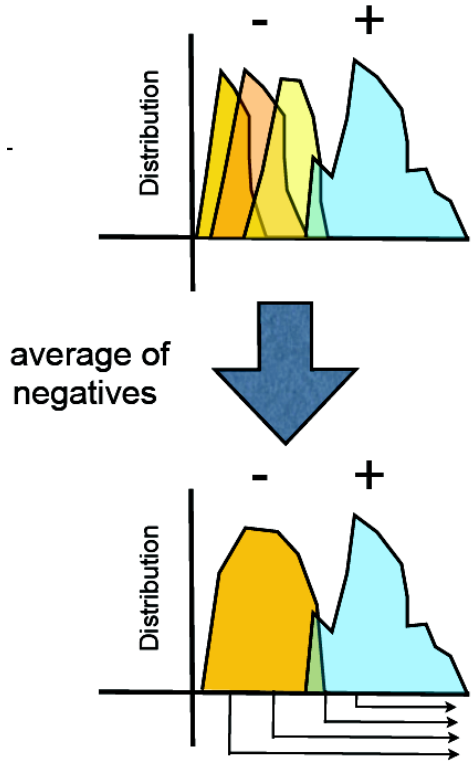
$$p_{t+1} = (1 - r) * W' p_t + r * p_0$$

W is the adjacency matrix whose elements are normalized by the square root of the multiplication of node degrees of the nodes that define the edge at that cell.

Vanunu et al., 2011, Plos Comp. Bio.

Evaluation of Classification Performance

- no negative data



Optimization of the Parameters of the Algorithms

Five-fold cross validation AUC using text-mining data

- Aneurysm
- Breast Cancer

SCAIView *(Hofmann-Apitius et al., 2008)*

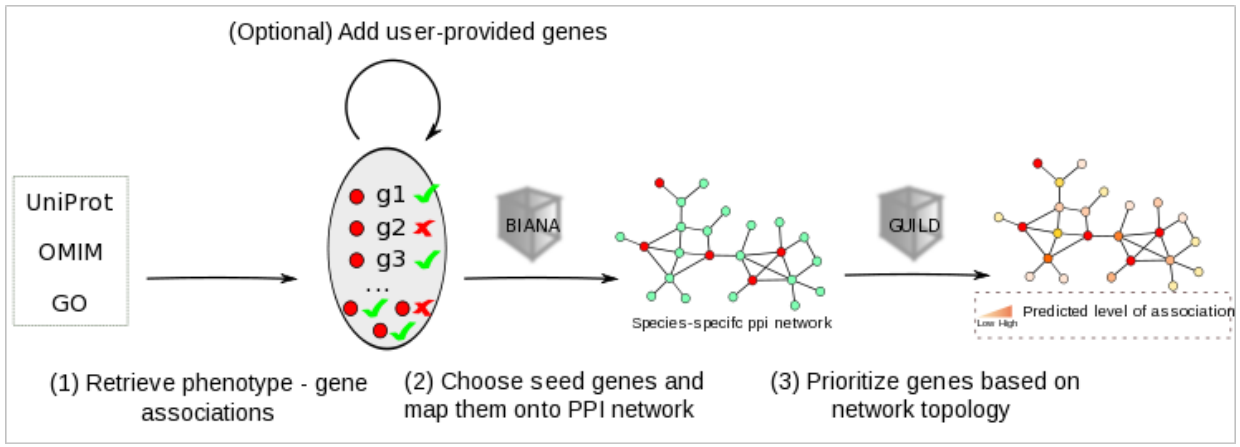
GO Term Enrichment Analysis

- FuncAssoc2 *(Berriz et al., 2009)*
- proteins mapped to gene symbols using UniProt
- all the proteins in the network used as background
- consider GO terms with *adjusted p-value* ≤ 0.05

GUILD Framework

- NetScore, NetZcore, NetShort, Functional Flow and PageRank are implemented with C++ using BOOST Graph Library (BGL), with the exception of FFlow which uses a self-implemented graph algorithm
- Random walk with restart and network propagation are implemented in R
- Experimentation setup is written in Python
- R is used for statistical analysis (with ROCR package for ROC curve analysis) *(Sing et al., 2005)*

Phenotypic Characterization of Genes through Biological Data Integration



Back

GUILD Predicts Disease-gene Association Better than Existing Network-based Prioritization Methods

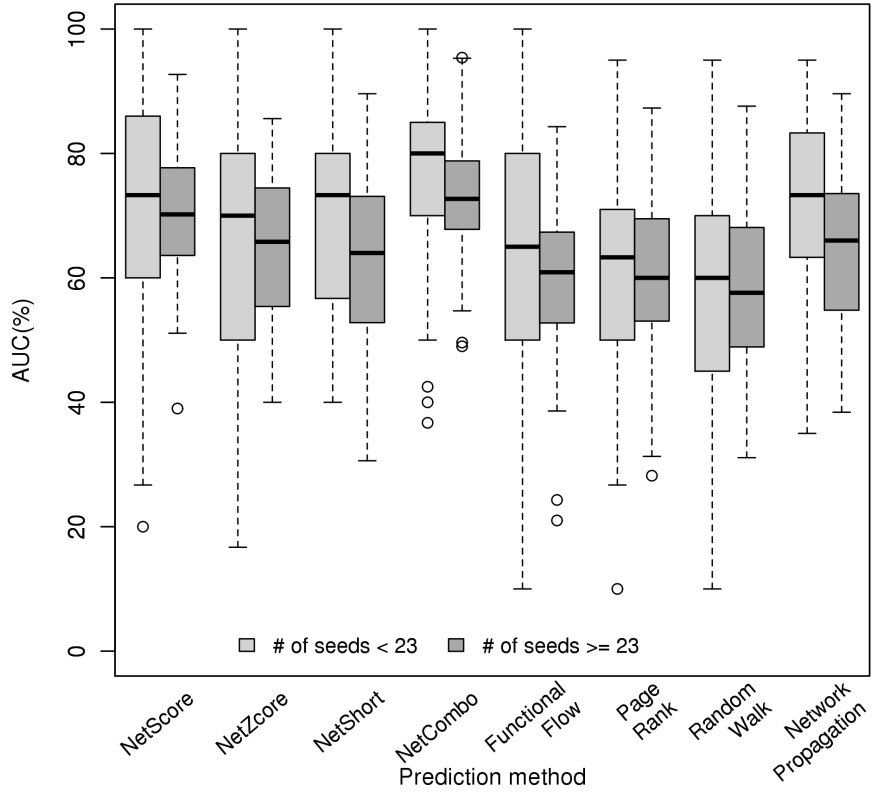
Data Set	NetScore	NetZcore	NetShort	NetCombo	Func. Flow	PageRank	Random Walk	Network Prop.
OMIM	75.50	73.11	69.19	74.57	72.67	70.70	70.82	72.51
Goh	73.64	70.84	66.27	73.05	68.00	68.62	69.09	70.56
Chen	81.27	78.75	68.66	80.59	74.88	76.14	75.81	78.14

Weighted bPPI (bPPI with edge weights from STRING)

OMIM	73.78	72.03	68.69	**76.51**	62.48	61.73	58.09	69.71
Goh	**72.09**	66.92	59.81	71.97	59.37	56.01	51.02	57.46
Chen	83.07	80.69	69.16	**84.30**	69.29	72.09	66.40	73.44
Overall	74.67	71.45	66.09	**76.51**	62.83	62.08	57.70	66.53

Guney and Oliva, 2012

Prioritization Methods Depend on the Connectedness of Seeds Rather than the Number of Seeds



Guney and Oliva, 2012

GUILD Scores Distinguish Disease-Genes from the Rest of the Genes

CTD

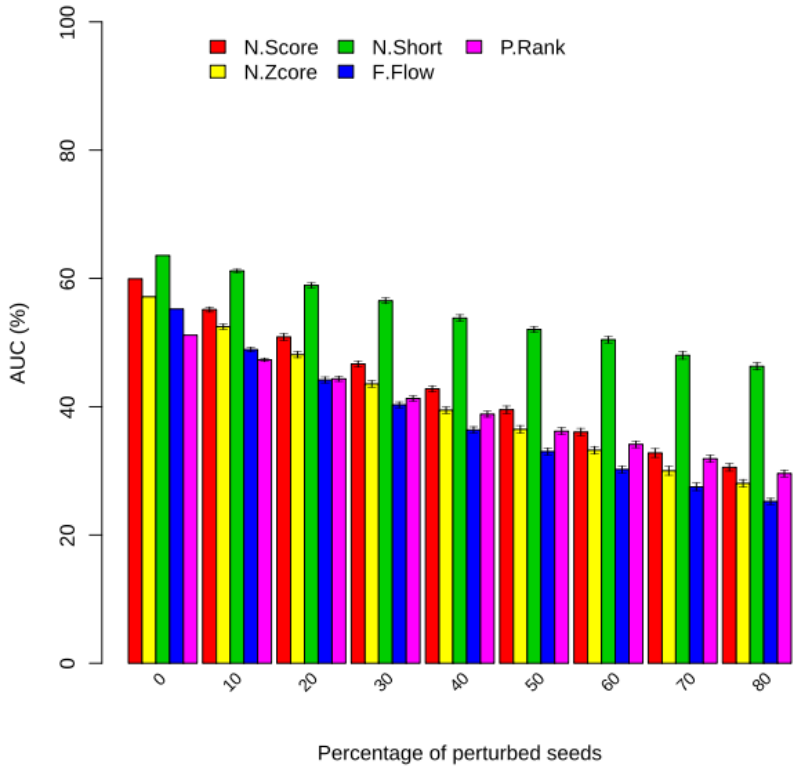
	Average score of direct association group	Average score of indirect association group	Average score of no-association group	P-value of direct vs indirect ($P \leq p$)	P-value of direct vs no-association ($P \leq p$)
AD	0.134 (0.093)	0.069 (0.046)	0.054 (0.038)	5.8e-5	2.6e-6
Diabetes	0.089 (0.085)	0.058 (0.035)	0.049 (0.028)	1.3e-3	5.2e-5
AIDS	0.123 (0.075)	0.079 (0.032)	0.071 (0.029)	1.3e-3	2.6e-4

GAD

	Number of top-ranking genes (predictions)	Number of top-ranking genes in GAD	Number of GAD genes in the network	Number of genes in the network	P-value ($P \leq p$)
AD	89	13	107	9469	1.6e-11
Diabetes	56	5	183	9432	4.5e-03
AIDS	102	3	11	9477	1.9e-04

Guney and Oliva, 2012

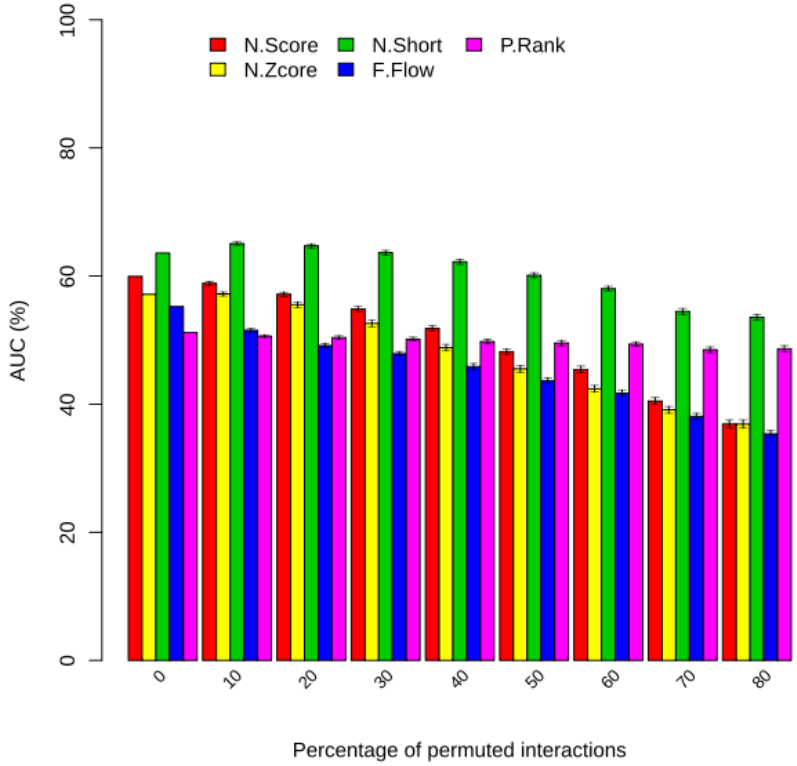
Dependence on the Seed Set - Goh Network



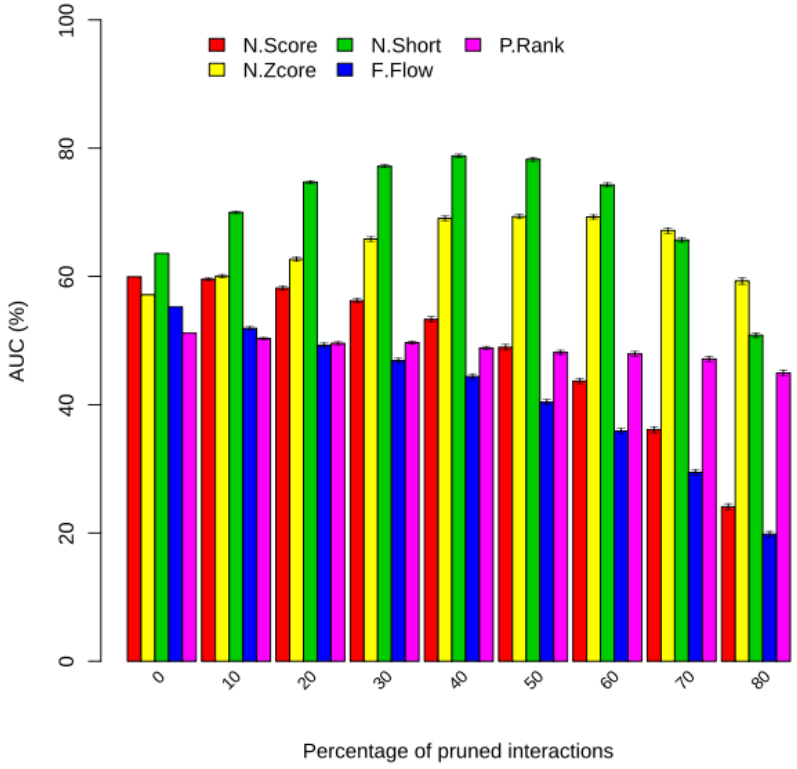
- NetScore
- NetZcore
- NetShort
- Functional Flow
- Page Rank

seed swapping

Dependence Against Underlying Network - Goh Network

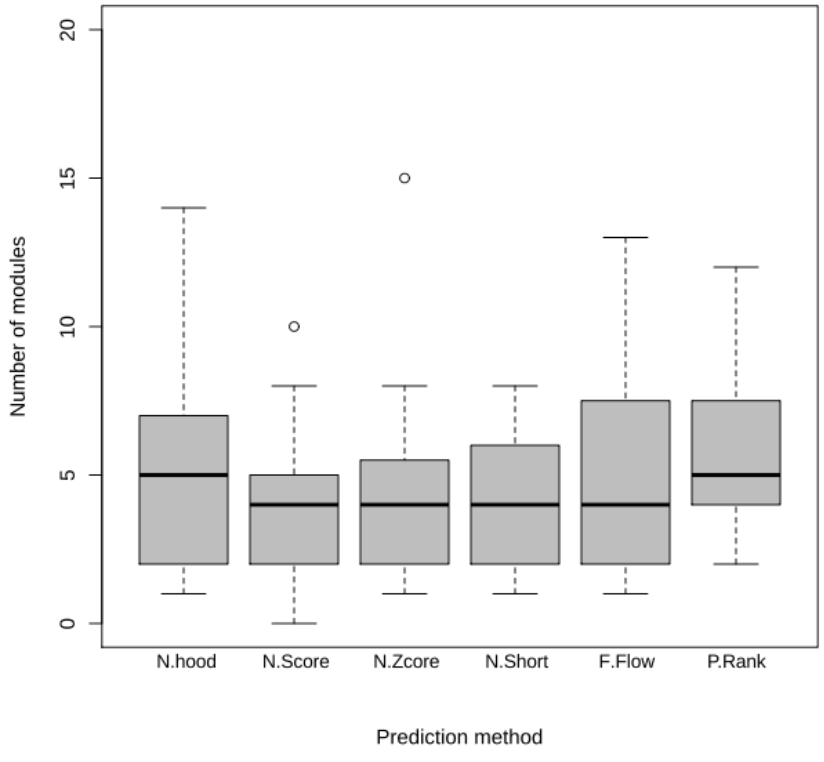


interaction rewiring

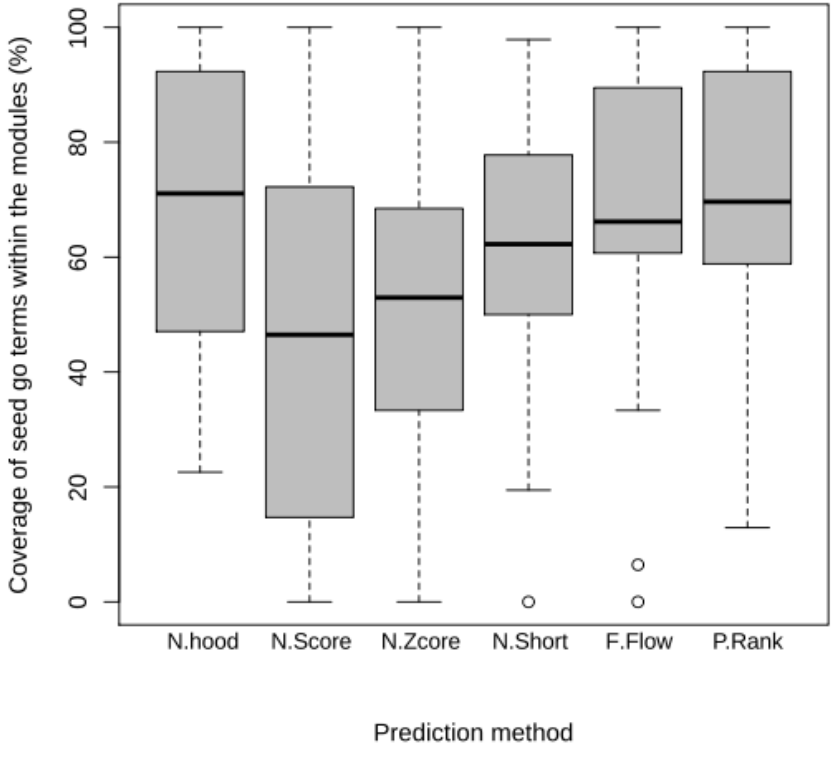


interaction removal

Modularity and Functional Enrichment of High Scoring Subnetworks

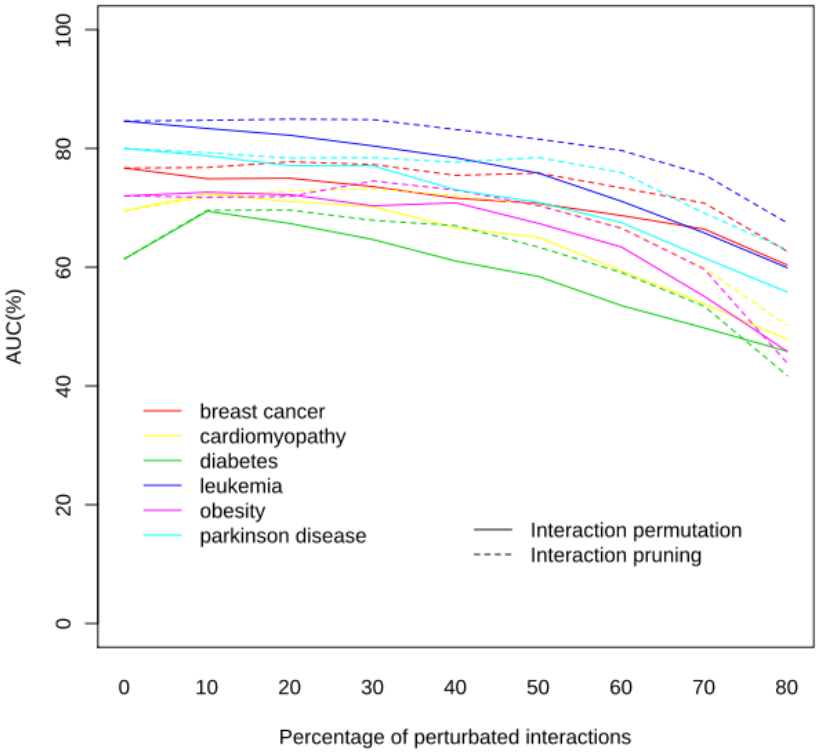


# of modules

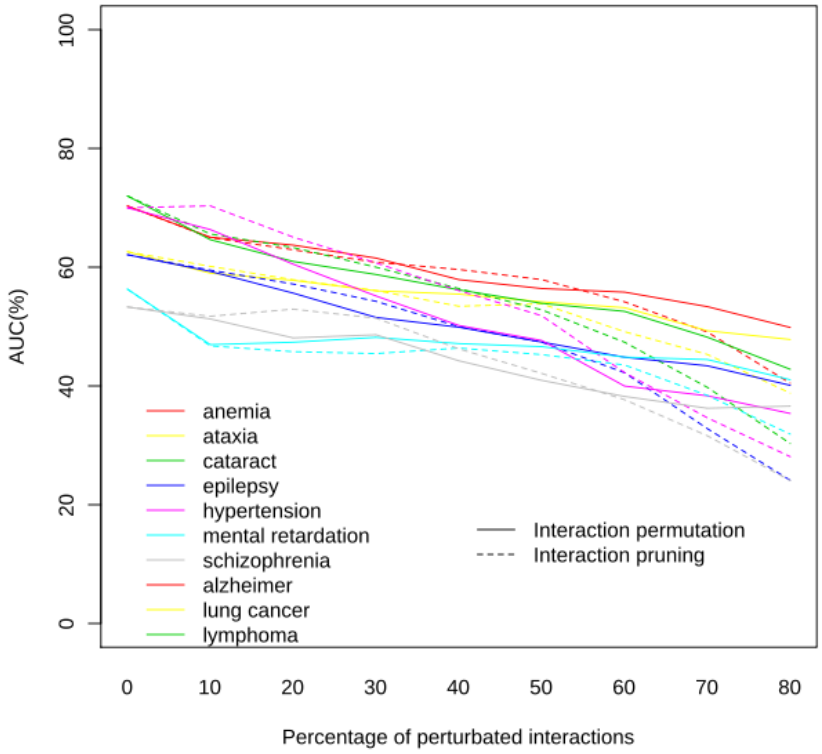


# of seed GO terms in modules

Categorizing Network-centric Tolerance of Diseases

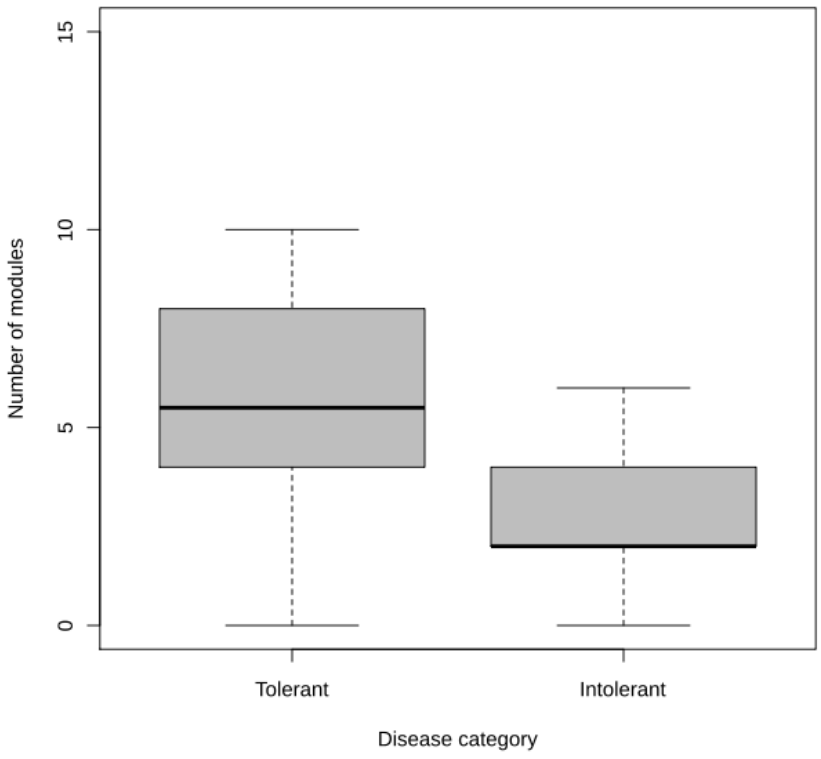


robust

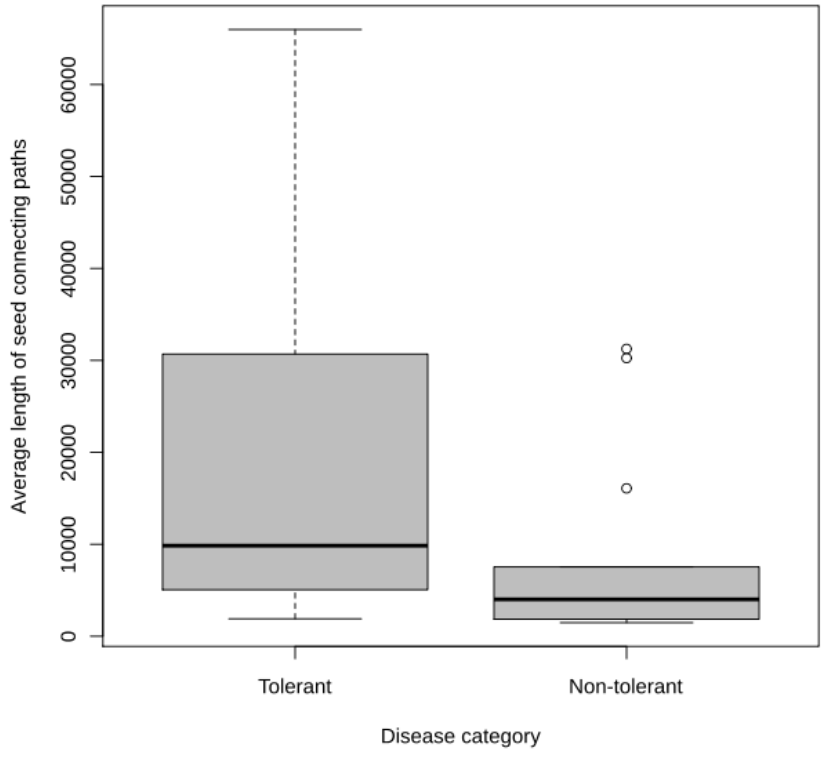


non-robust

Comparison Between Diseases

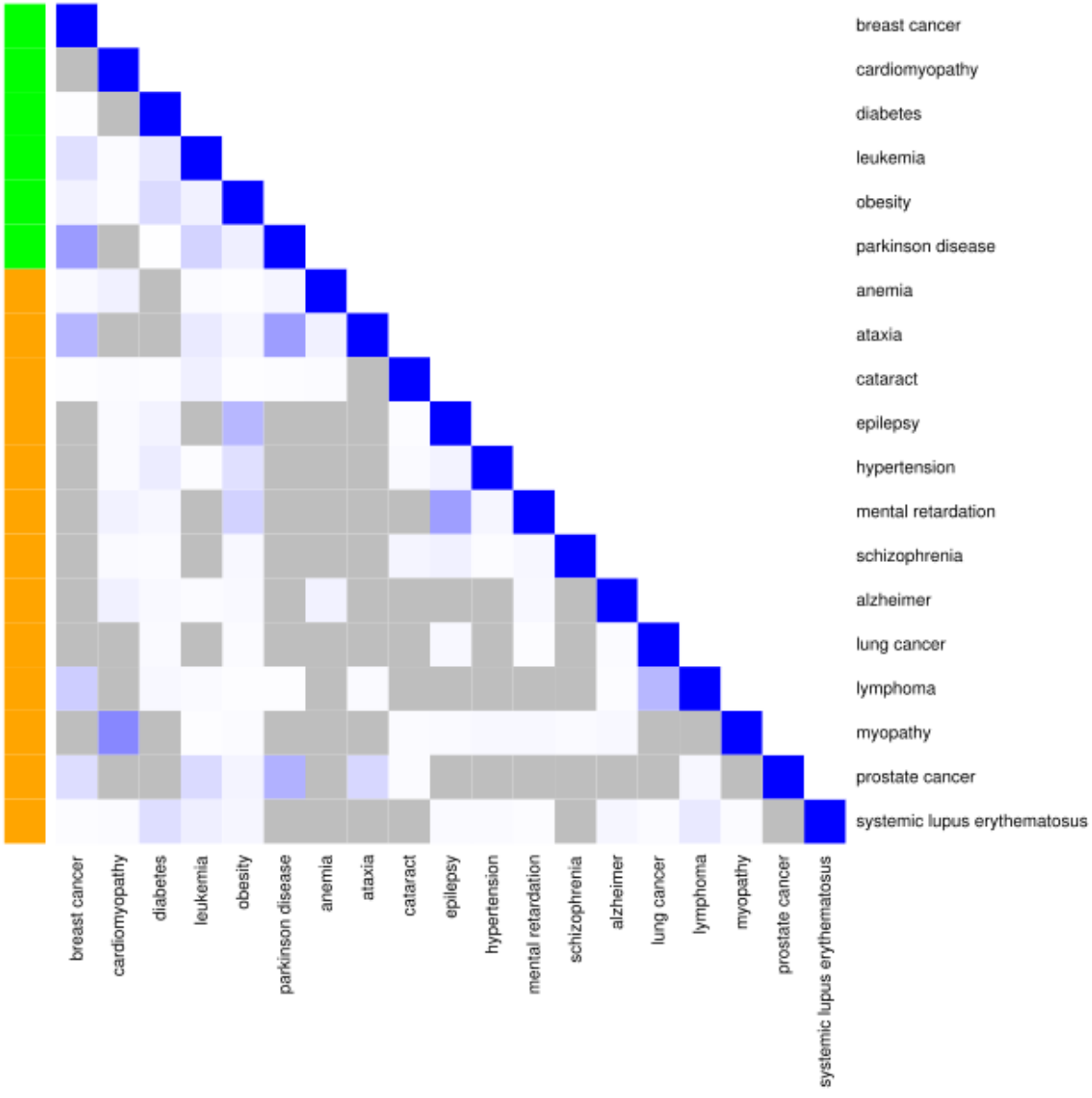


# of modules



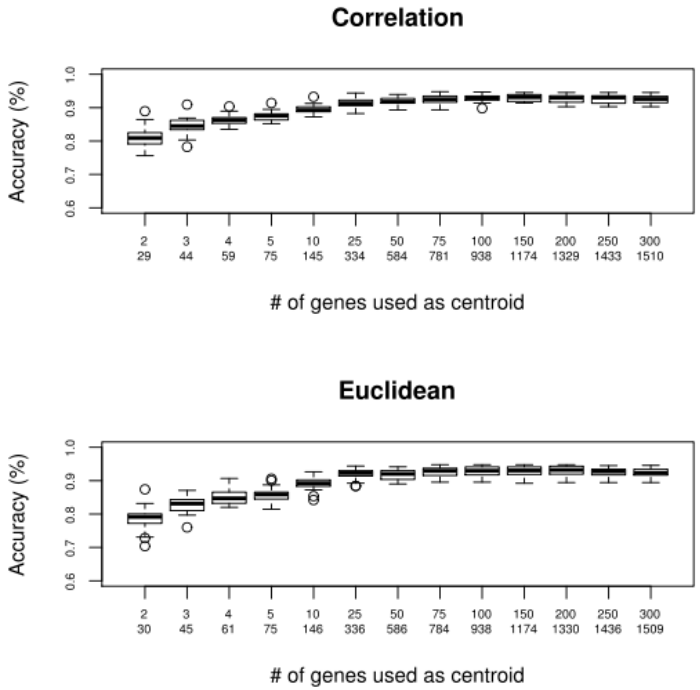
# of alternative paths

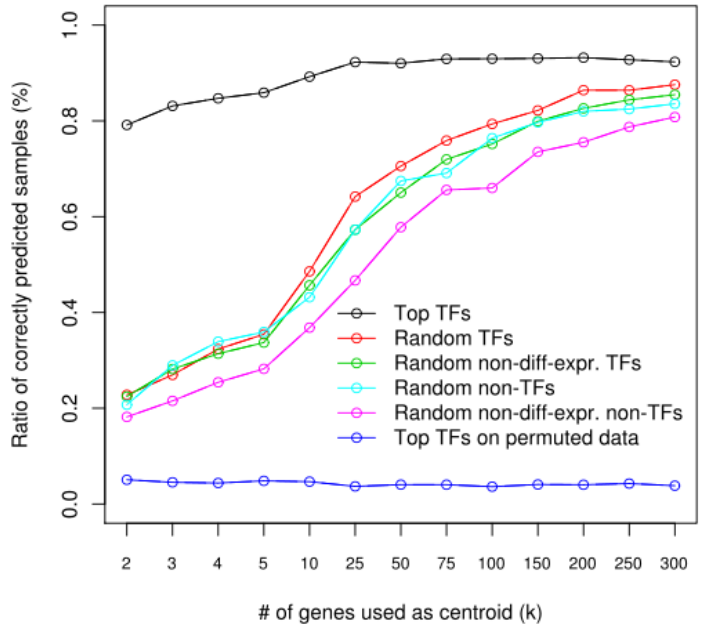
Functional Similarity Between Diseases



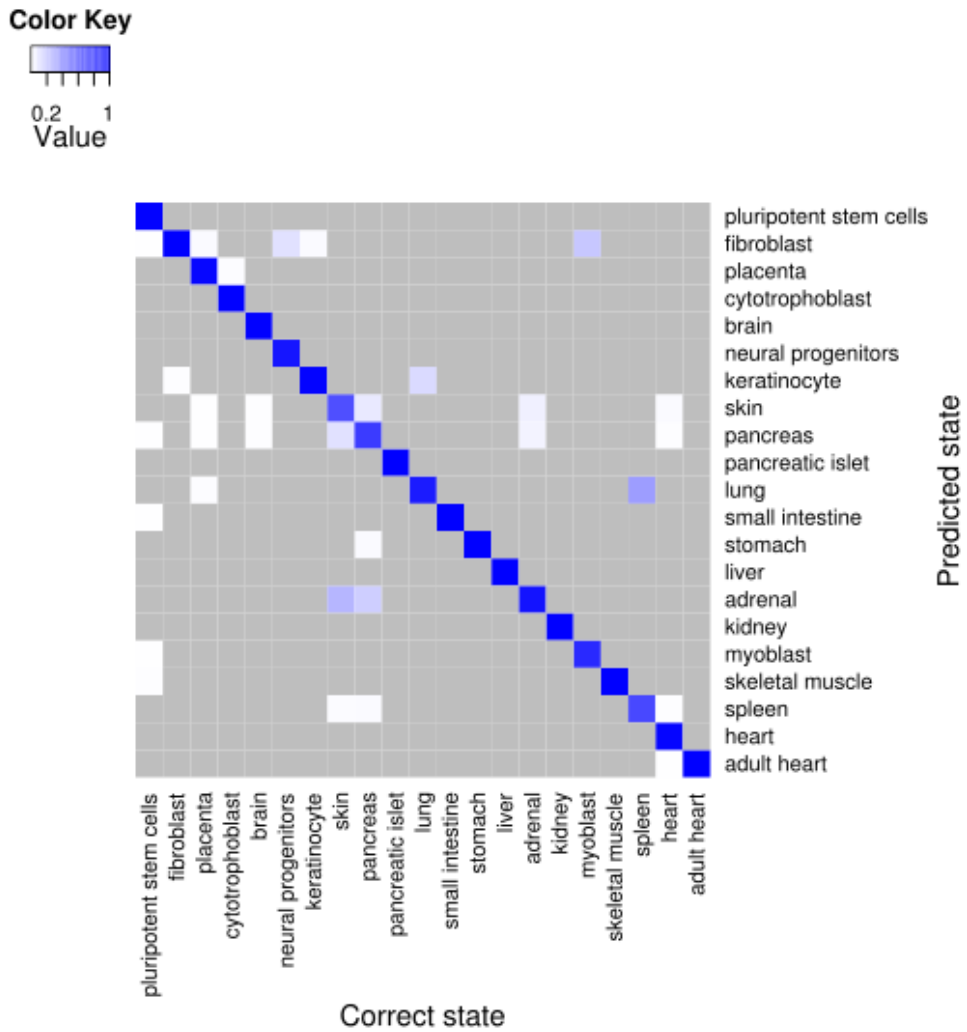
Back

Transcription Factors Mediating Human Cell Differentiation

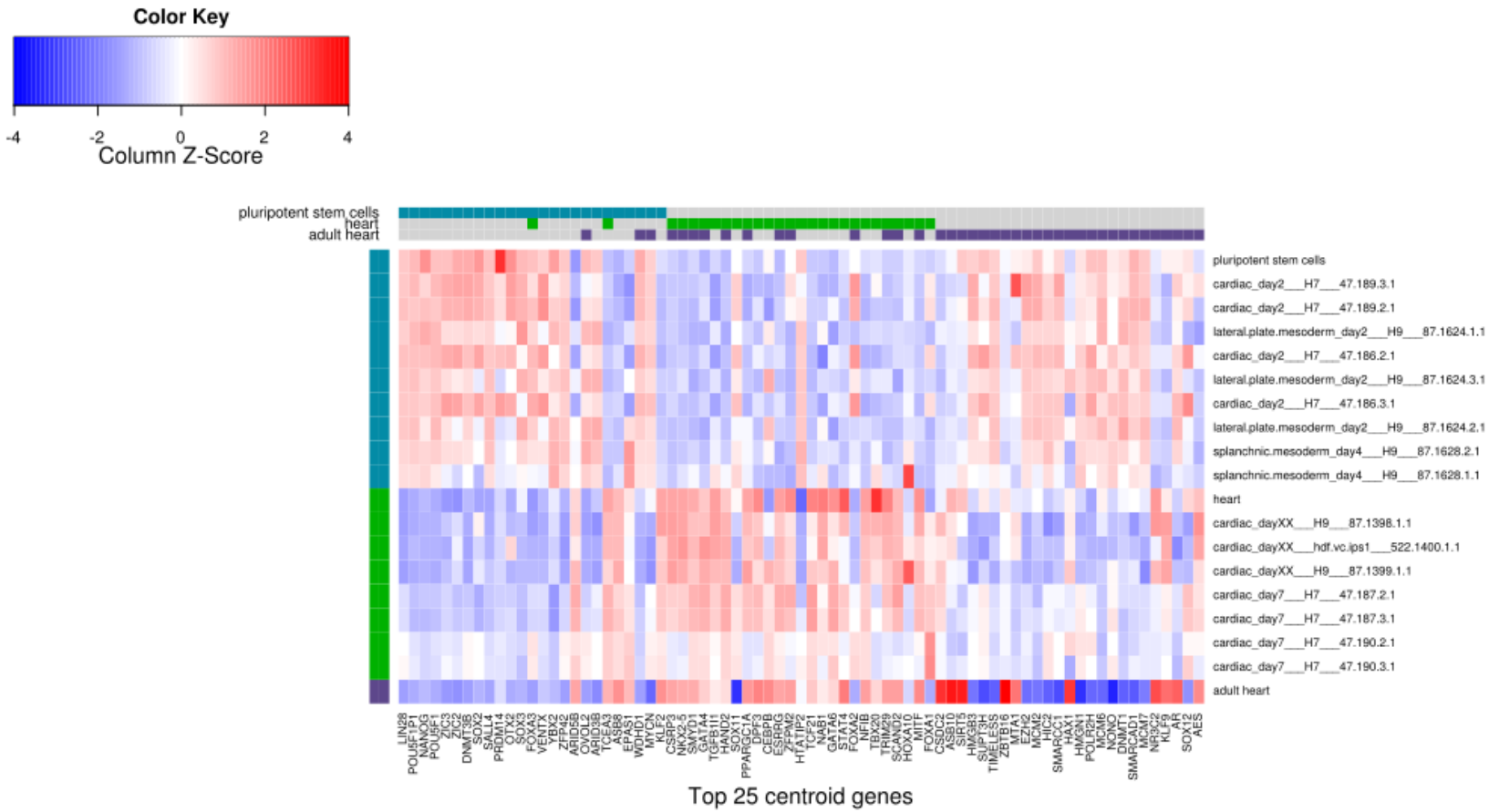




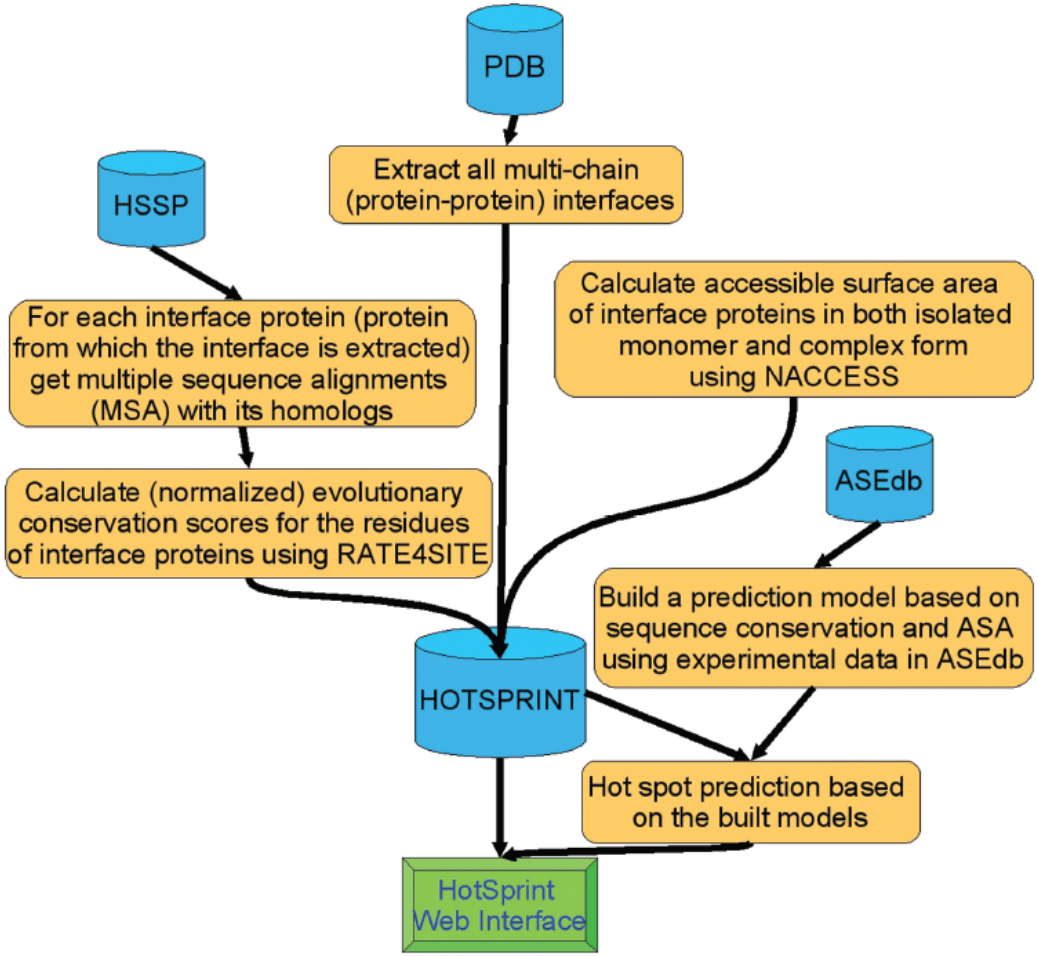
Transcription Factors Mediating Human Cell Differentiation



Transcription Factors Mediating Human Cell Differentiation



Prediction of Hot Spot at Protein-Protein Interfaces



Guney et al. 2008

Prediction of Hot Spot at Protein-Protein Interfaces

Interface 1yp2AB:

- *Overall Characteristics*

Protein 1yp2:	CRYSTAL STRUCTURE OF POTATO TUBER ADP-GLUCOSE PYROPHOSPHORYLASE

Interface 1yp2AB	# of Residues	# of Hot Spots	# of Conserved Residues	Avg. Conservation Score	Buried ASA (A^{o2})
Chain A	34	16	16	6	1401
Chain B	33	15	15	6	1410
Total	67	31	31	6	2811

External Links:	1yp2AB @ DAPPI (Interaction information)

- *Residues of the Interface*

Legend:

C: Contacting Interface Residue

N: Neighboring Interface Residue

Interface Residue | Conserved Residue | Interface Hot Spot

1yp2A

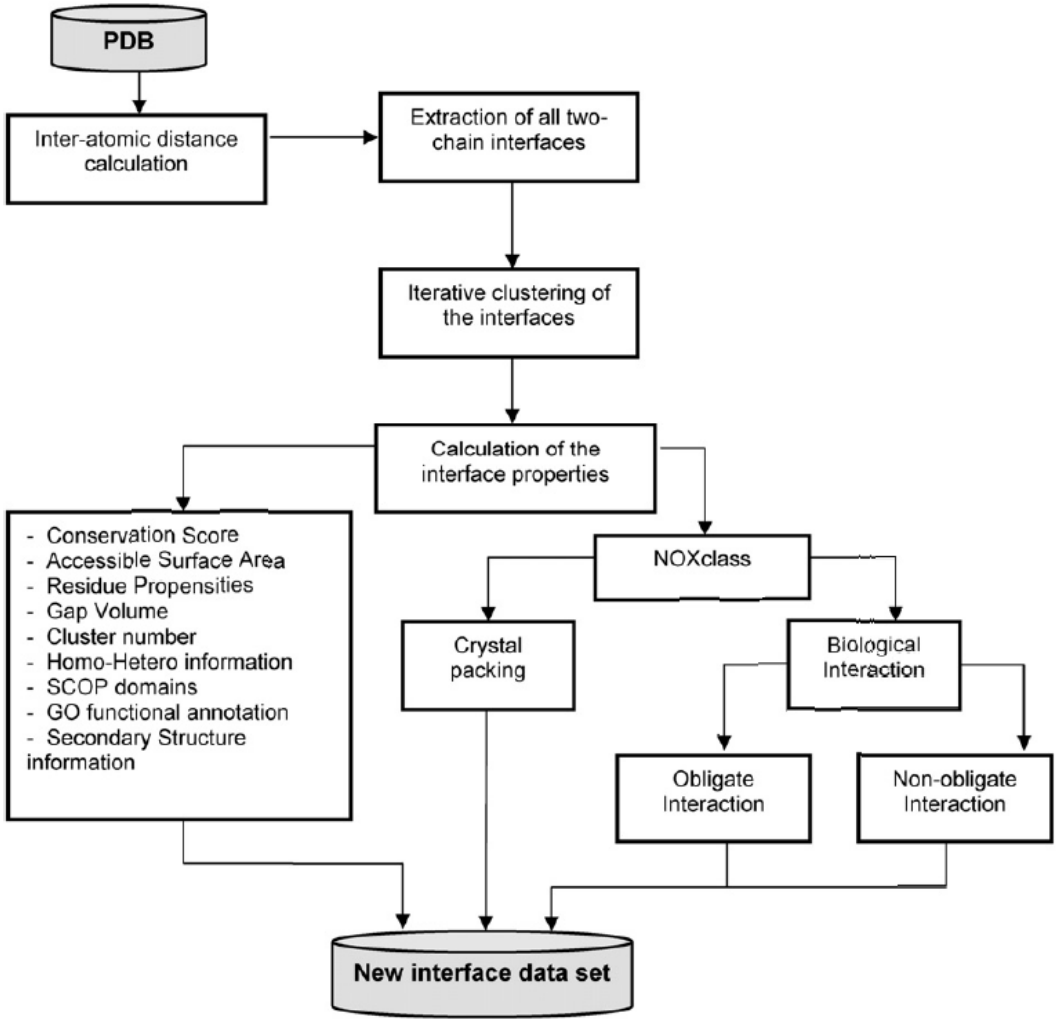
Position	Name	Cons. Score	ASA in Chain (A^{o2})	ASA in Complex (A^{o2})	Type	Hot Spot
47	PRO	7	6.74	6.74	N	
48	LEU	5	2.14	2.14	N	
49	GLY	7	3.86	0.46	C	*
50	ALA	8	13.5	9.05	C	*
51	ASN	5	28.55	4.18	C	
52	TYR	7	37.2	7.7	C	*

1yp2B

Position	Name	Cons. Score	ASA in Chain (A^{o2})	ASA in Complex (A^{o2})	Type	Hot Spot
47	PRO	7	6.96	6.96	N	
48	LEU	5	2.44	2.44	N	
49	GLY	7	5.21	0.12	C	*
50	ALA	8	11.82	8.18	C	*
51	ASN	5	28.13	4.15	C	
52	TYR	7	36.35	7.54	C	*

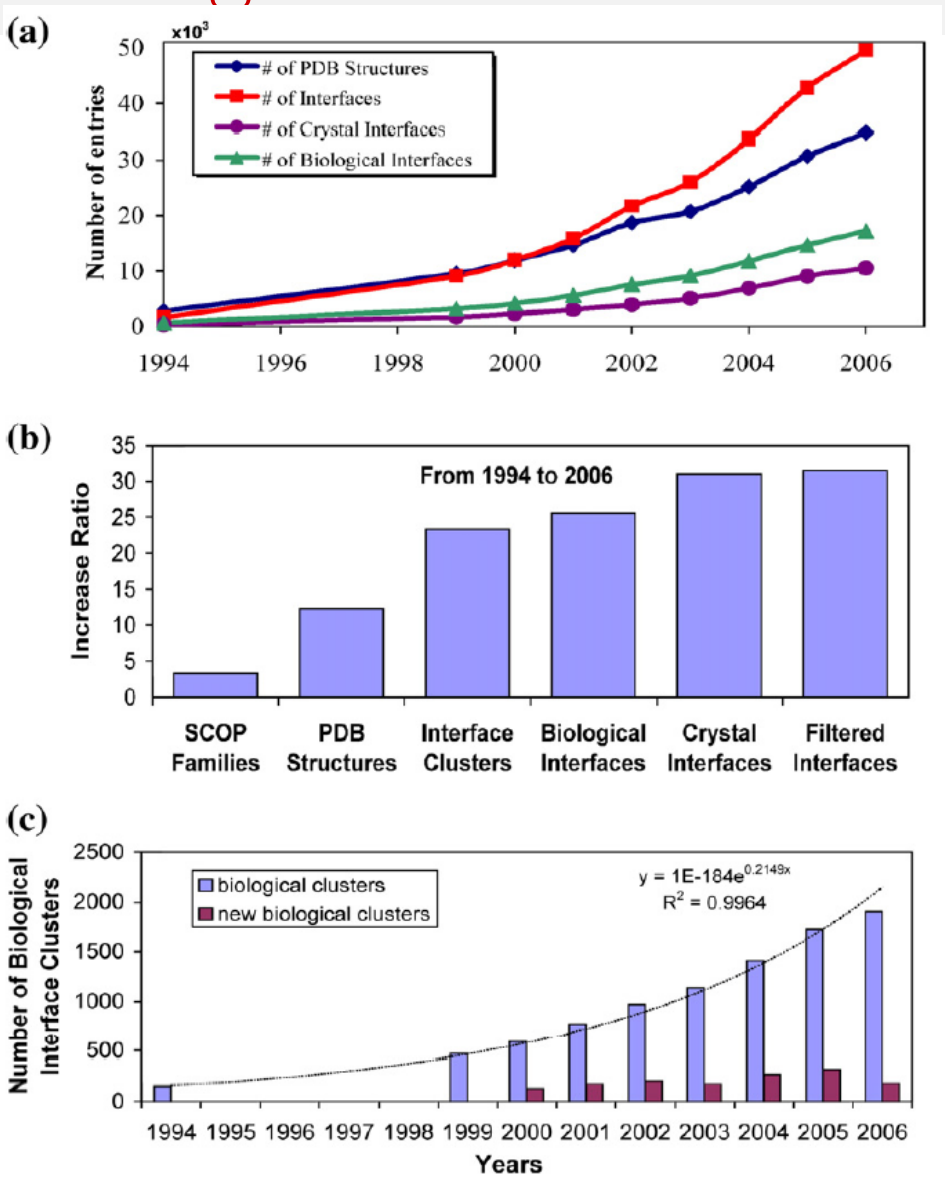
Guney et al., 2008

Functional Coverages of Protein-Protein Interfaces



Tuncbag et al., 2008

Functional Coverages of Protein-Protein Interfaces



Tuncbag et al., 2008

Back

Towards Context-Specific Networks and Their Implications for Network Medicine

Mapping and analysis of chromatin state dynamics in nine human cell types

Jason Ernst[1,2], Pouya Kheradpour[1,2], Tarjei S. Mikkelsen[1], Noam Shoresh[1], Lucas D. Ward[1,2], Charles B. Epstein[1], Xiaolan Zhang[1], Li Wang[1], Robbyn Issner[1], Michael Coyne[1], Manching Ku[1,3,4], Timothy Durham[1], Manolis Kellis[1,2]* & Bradley E. Bernstein[1,3,4]*

Focusing on cell-type-specific patterns of promoters and enhancers, we define multicell activity profiles for chromatin state, gene expression, regulatory motif enrichment and regulator expression. We use correlations between these profiles to link enhancers to putative target genes, and predict the cell-type-specific activators and repressors that modulate them. The resulting annotations and regulatory predictions have implications for the interpretation of genome-wide association studies. Top-scoring disease single nucleotide polymorphisms are frequently positioned within enhancer elements specifically active in relevant cell types, and in some cases affect a motif instance for a

Towards Context-Specific Networks and Their Implications for Network Medicine

Mapping and analysis of chromatin state dynamics in nine human cell types

Jason Ernst[1,2], Pouya Kheradpour[1,2], Tarjei S. Mikkelsen[1], Noam Shoresh[1], Lucas D. Ward[1,2], Charles B. Epstein[1], Xiaolan Zhang[1], Li Wang[1], Robbyn Issner[1], Michael Coyne[1], Manching Ku[1,3,4], Timothy Durham[1], Manolis Kellis[1,2]* & Bradley E. Bernstein[1,3,4]*

Focusing on cell-type-specific patterns of promoters and enhancers, we define multicell activity profiles for chromatin state, gene expression, regulatory motif enrichment and regulator expression. We use correlations between these profiles to link enhancers to putative target genes, and predict the cell-type-specific activators and repressors that modulate them. The resulting annotations and regulatory predictions have implications for the interpretation of genome-wide association studies. Top-scoring disease single nucleotide polymorphisms are frequently positioned within enhancer elements specifically active in relevant cell types, and in some cases affect a motif instance for a

Tissue-Specific Functional Networks for Prioritizing Phenotype and Disease Genes

Yuanfang Guan[1,2], Dmitriy Gorenshteyn[3,4], Margit Burmeister[1,5], Aaron K. Wong[3], John C. Schimenti[6], Mary Ann Handel[7], Carol J. Bult[7], Matthew A. Hibbs[7,8]*, Olga G. Troyanskaya[3,9]*

phenotypes. Our results demonstrate that prediction performance is significantly improved through using the tissue-specific networks as compared to the global functional network. We used a testis-specific functional relationship network to

Towards Context-Specific Networks and Their Implications for Network Medicine

Adding Protein Context to the Human Protein-Protein Interaction Network to Reveal Meaningful Interactions

Martin H. Schaefer[1], Tiago J. S. Lopes[2], Nancy Mah[1], Jason E. Shoemaker[2], Yukiko Matsuoka[2,3], Jean-Fred Fontaine[1], Caroline Louis-Jeune[4], Amie J. Eisfeld[5], Gabriele Neumann[5], Carol Perez-Iratxeta[4], Yoshihiro Kawaoka[2,5,6], Hiroaki Kitano[2,7,8,9], Miguel A. Andrade-Navarro[1]*

Towards Context-Specific Networks and Their Implications for Network Medicine

Adding Protein Context to the Human Protein-Protein Interaction Network to Reveal Meaningful Interactions

Martin H. Schaefer[1], Tiago J. S. Lopes[2], Nancy Mah[1], Jason E. Shoemaker[2], Yukiko Matsuoka[2,3], Jean-Fred Fontaine[1], Caroline Louis-Jeune[4], Amie J. Eisfeld[5], Gabriele Neumann[5], Carol Perez-Iratxeta[4], Yoshihiro Kawaoka[2,5,6], Hiroaki Kitano[2,7,8,9], Miguel A. Andrade-Navarro[1]*

SAPIN: Structural Analysis for Protein Interaction Networks

Jae-Seong Yang[1], Anne Campagna[1], Javier Delgado[1], Peter Vanhee[1], Luis Serrano[1,2,*] and Christina Kiel[1,*]

identified parts, and (iii) the identification of compatible and mutually exclusive interactions at the network level.

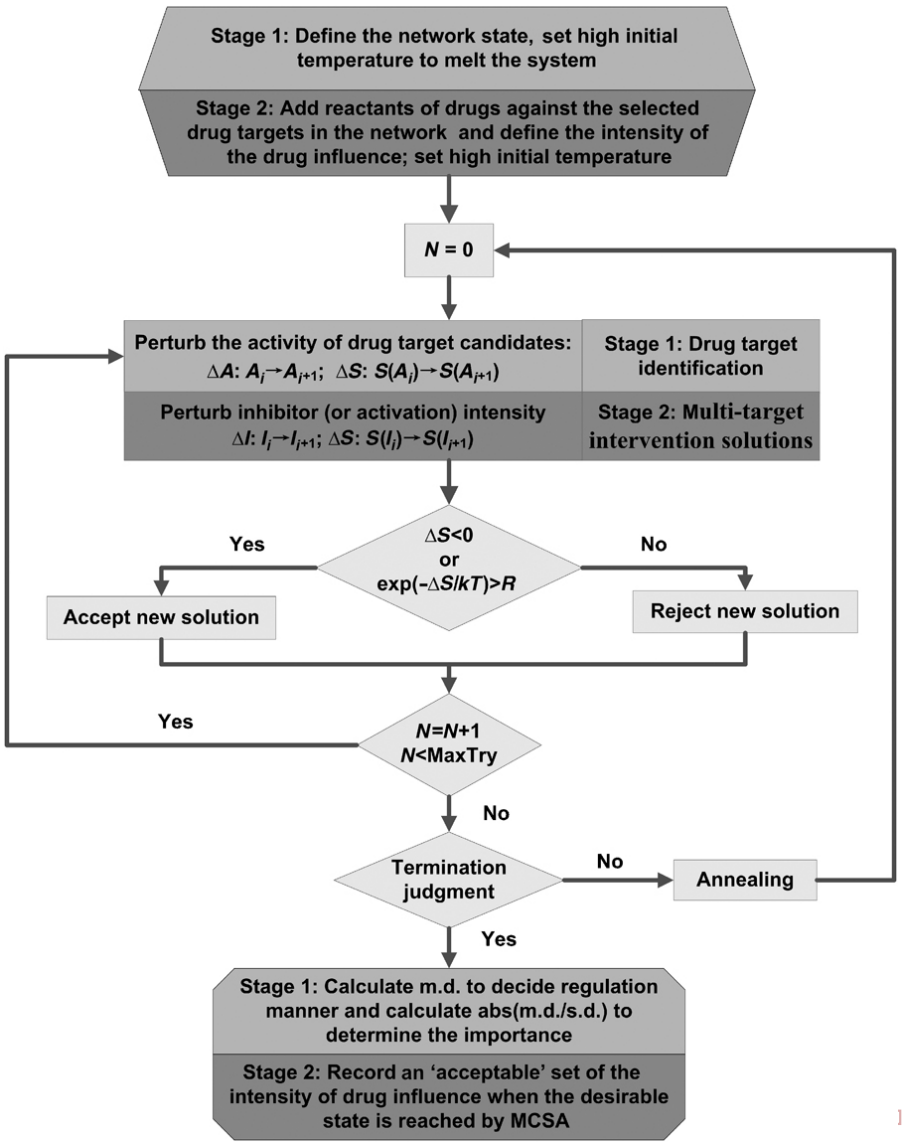
Towards Context-Specific Networks and Their Implications for Network Medicine

- Integration
  - proteomics
  - genetics
  - epigenetics
- Analysis
  - disease modules
  - common dysregulated pathways
  - optimum set of drugs & their targets

Network (Poly)Pharmacology

- optimal drug combinations *(Vazquez, 2009, Wang et al., 2012)*
- disease comorbidity *(Lee et al., 2008, Park et al., 2009)*

Finding Multiple Target Optimal Intervention in Disease-related Molecular Network



Yang et al. 2008

Back

Manuscripts Produced During the Development of the PhD Thesis

Journal Articles

- **Guney E.**, Oliva B. Exploiting Protein-Protein Interaction Networks for Genome-wide Disease-Gene Prioritization., *PLOS One*, 2012, 10.1371/journal.pone.0043557.
- Garcia-Garcia J, **Guney E**, Aragues R, Planas-Iglesias J, Oliva B. BIANA: A software framework for compiling biological interactions and analyzing networks. *BMC Bioinformatics*, 2010, 11:56.
- Planas-Iglesias J, **Guney E**, Garcia-Garcia J, Robertson KA, Raza S, Freeman TC, Ghazal P and Oliva B. Extending signalling pathways with protein-interaction networks. Application to apoptosis. *OMICS: A journal of Integrative Biology*, 2012, 16(5): 245-256.
- Garcia-Garcia J, Bonet J, **Guney E**, Fornes O, Planas-Iglesias J, Oliva B. Networks of Protein-Protein Interactions: from uncertainty to molecular details. *Molecular Informatics*, 2012, 31(5): 342-362.

Book chapter

- **Guney E**, Sanz R, Angels S, Oliva B. Understanding Cancer Progression Using Protein Interaction Networks. *Systems Biology in Cancer Research and Drug Discovery, Part 2*, Springer, 2012, Pages 167-195.

Manuscripts Produced During the Development of the PhD Thesis

In preparation / submitted

- **Guney E**, Oliva B. Robustness analysis of network-based disease gene prioritization methods points to redundancy and functional decoupling in the human interactome. *Submitted*.
- Santana-Codina N, Carretero R, Sanz-Pamplona R, Cabrera T, **Guney E**, Oliva B, Clezardin P, Olarte OE, Loza P, Mendez-Lucas A, Perales JC, and Sierra A. A transcriptome-proteome integrated network identifies ERp57 as a hub that mediates bone metastasis. *Submitted*.
- **Guney E**, Garcia-Garcia J, Oliva B. GUILDify: A web server for phenotypic characterization of genes through biological data integration and network-based prioritization algorithms. *In preparation*.

Abstracts

- **Guney E**, Oliva B. Toward PWAS: discovering pathways associated with human disorders. *BMC Bioinformatics*, 2011, 12(Suppl 11):A12

Manuscripts Produced During the Development of the Masters Thesis

Journal Articles

- **Guney E**, Tuncbag N, Keskin O, Gursoy A. HotSprint: database of computational hot spots in protein interfaces. *Nucleic Acids Res.*, 2008, 36:D662D666.
- Tuncbag N, Gursoy A, **Guney E**, Nussinov R, Keskin O. Architectures and functional coverage of protein-protein interfaces. *Journal of Molecular Biology*, 2008, 381(3), 785802.

Abstracts / Conference Proceedings

- **Guney E**, Tuncbag N, Keskin O, Gursoy A. Large Scale Identification of Spatial Motifs in Protein Interfaces. *Proc. of HIBIT'08*, Istanbul.
- Tuncbag N, Gursoy A, **Guney E**, Nussinov R, Tsai CJ, Keskin O. A New Dataset of Protein-Protein Interfaces. *Proc. of the 51st Biophy. Soc. Ann. Meet.*, 2007.