

Systems Biology – FS 2017

The structure or form of a protein determines its function and use within an organism. Throughout evolution, the base sequence and the function have been conserved for the most part.

Ex.: The genetic code for ubiquitin in yeast and humans is to 96% identical. The function remains the same.

Def. Genetic determinism: Gene → Protein → Function

The reality shows that the current dominant theory of genetic determinism is incomplete. Sending a drug into a cell that targets a certain protein should work in theory, but in more than 90% this is not the case.

Classically, a drug takes effect on its target, which results in a therapeutic effect. If it also takes effect on other off-targets, one considers these side effects.

In a more contemporary view, a drug takes effect on many different targets and the resulting effects affect each other in turn and so on. One will also gain a therapeutic effect plus side effects. (**network view of a drug**).

Sometimes, more than just one model can describe the data. What does it mean? They are all correct. A next step would be to see if the models can predict future data (they have to be measured in reality) or disturb the system and see if the models still describe the data correctly.

Definitive Summary of Systems Biology

10.5.2017 – 12.5.2017

Conventions:

$X := [X]$, where X is a molecule and $[X]$ is its concentrations, for convenience's sake.

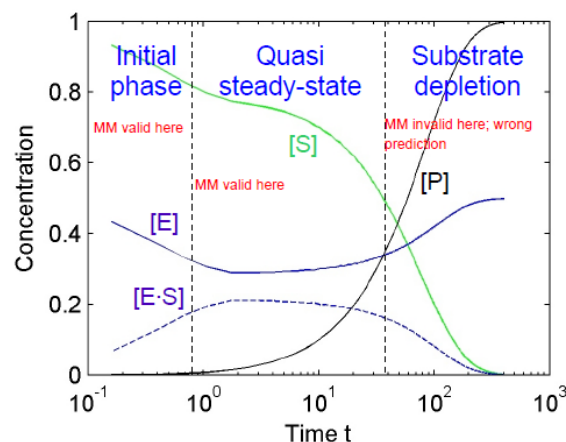
The Michaelis-Menten model:

Assumptions: 1) Steady state: $dX/dt = 0$, especially $dES/dt = 0$.

2) There is an infinite amount of substrate S (not true for all problems).

3) Total concentration of all enzymes E_i is conserved: $E_{\text{tot}} = \sum_{i=1}^n E_i$.

4) No feedback by product P – $[P]$ only depends on $[ES]$.



(calculate all the specific cases as in Ex. 3 – that is inhibition with competition etc. => see Sysbio-2.5)

Structurally similar problem and generalization of reaction kinetics: Gene transcription:

Gene G is bound by transcription factor T:

Without repression: $[GT] = [G]^T * [T] / ([T] + K)$

Competitive repressor R: $[GT] = [G]^T * [T] / ([T] + K(1 + [R]/K_I))$

Cooperative binding: $[GT] = [G]^T * [T]**n / ([T]**n + K**n)$

General assumptions for a thermodynamic ODE system:

- 1) The system is well-mixed (no heterogeneity).
- 2) There is a very large number of molecules n.
- 3) It is a closed system (no external forces acting on the system and no exchange of mass m or energy E with its surroundings).

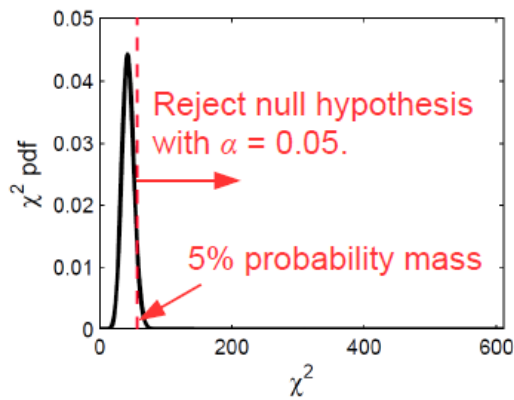
The complexity of a network increases extremely fast with the number of components. Thus, it becomes increasingly a challenge to handle vaster amounts of data.

	Components	Interactions
- enzyme reactions	3-10	>>50
- signaling pathway	3-30	>100
- metabolic network	~1 000	~2 000
- protein network	>1 000	~50 000
- cell	>10 000	?
- organs	>100 000	?
- human	?	?

Approximation for an unknown parameter with given data: Approximate K_m given the experimentally measured data x_e and the simulated data x_s with inherent measurement error σ_e :

$$\Phi(K_M) = \sum_{i=1}^N \left[\frac{x(t_i, K_M) - x^E(t_i)}{\sigma^E(t_i)} \right]^2$$

Use the chi-square test to ensure that data is not disturbed too much by background noise. If data is statistically significant ($d(x_e, x_s)$ is small enough), then the model can be used to approximate K_m .



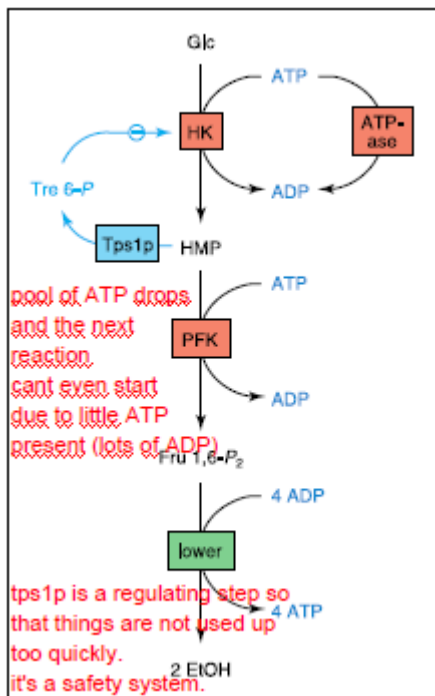
Metabolite dynamics for the network:

A cell faces several problems: Unregulated pathways lead to an overshoot of initial metabolites (Ex. Carbon sources) leading to **toxicity** within the cell (self-destruction). Overly abundant metabolites may start participating in other reactions of the cell, leading to competition and perturbation of the system's equilibrium (Ex. Cell might be drained of ATP, phosphates, proteins, other metabolites). Also, an unregulated pathway takes a very long time until it reaches steady state or it might never reach it.

A cell counters these problems by implementing regulatory mechanisms.

Control problems for a general biological system: avoid extreme responses; quickly achieve a new steady state; flexible responses during a time scale that matters (system has to be solid enough to remain at a constant state, but it has to have a certain range of flexibility to respond adequately to a sudden change in its environmental disposition, although not too much flexibility will lead to chaos (chaotic behaviour) and it might not be able to recover to its optimal original state); remain robust to different types, frequencies and intensities of perturbation.

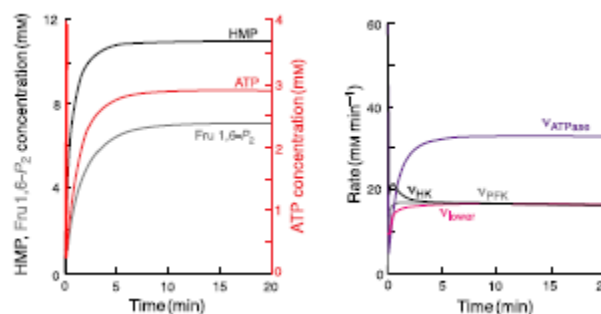
Ex.: Turbo Design (a positive feedback system): Regenerated ATP fuels the beginning of glycolysis.



Sudden excess of substrate leads in some species such as bacteria or yeast to substrate-accelerated cell death. Tps1p plays no role in glycolysis, but it is a control point that inhibits the first step of glycolysis to reduce the initial flux that is to guard the cell from glycolytic flux.

ATP drops extremely fast when unguarded and it can only recover a small amount after some time. No ATP leads to numerous problems for the cell, one of them cell death (there is no steady state at all).

Guarded glycolysis enables the cell to reach a steady state a lot quicker and easier (flux is balanced with regulation):



Feedback systems: Let X, Y be species (Ex. Metabolites, proteins) that interact with each other.

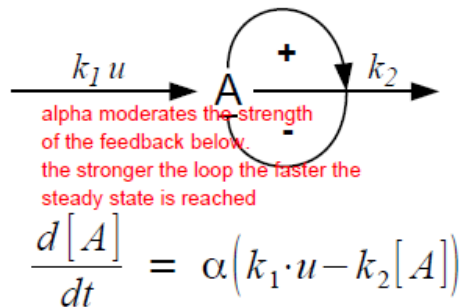
Def. positive feedback: "X activates Y, Y activates X" is called positive feedback.

Def. mutual antagonism: "X inactivates/inhibits Y, Y inactivates/inhibits X" is called mutual antagonism

Def. negative feedback: "X activates Y, Y inactivates/inhibits X" is called negative feedback.

Negative feedback will lead the system to steady state eventually (product-degradation feedback). In biological systems, negative feedback conducive for reaching homeostasis (existence of a steady state), which is often a desired state to be in.

Hypothesis: Existence of negative feedback in systems \Leftrightarrow existence of steady state (homeostasis).



The bigger alpha the faster the system reaches steady state. Also, this leads to a faster response to perturbation in biological systems. (Ex.: glycolysis)

Positive feedback enables a system to have several steady states, where it might be desirable to be in a certain steady state under certain external conditions. In a nonlinear system, the phenomenon of memory occurs where the system “remembers” its internal state. In a biological system, at some point, it might enter an irreversible state. We call such a process **development**.

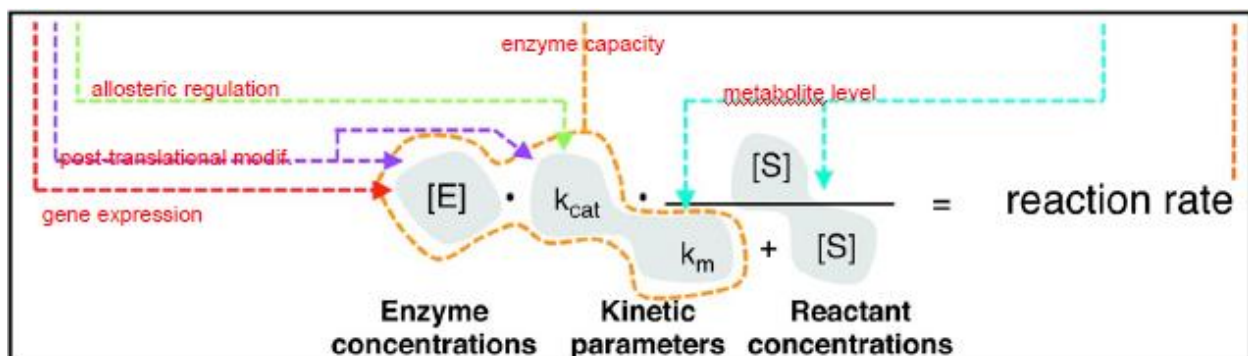
Challenges of a pathway operating in two ways:

Direction is determined by metabolite concentration. Genetic regulation is needed to reduce protein costs. When external conditions change, new reactions (pathways) will be needed to occur, but genetic regulation takes time. Simultaneous presence/activity of enzymes catalysing opposing directions lead to futile cycling.

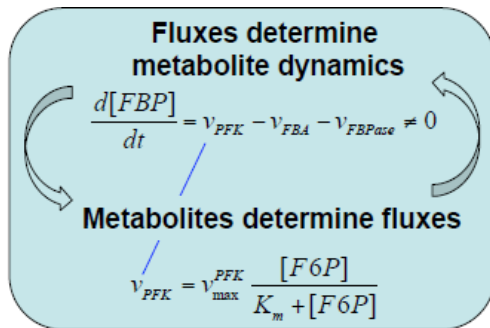
Mechanisms a cell uses to control reaction rates:

Allosteric effects (regulation by metabolite-protein interaction); post-translational modification; transcriptional regulation.

All these effects change enzyme activity/abundance.



Relationship between metabolite concentration and fluxes:



The flux can be seen as the flow of matter (metabolites) using only this one single characteristic to describe a whole metabolite network. Also, a change of flux in a certain metabolic pathway should also be communicated to the rest of the cell so that it can remain in a steady state.

Flux and phenotype: The flux in a reaction can be defined based on one of three things:

The activity of the enzyme catalysing the reaction.

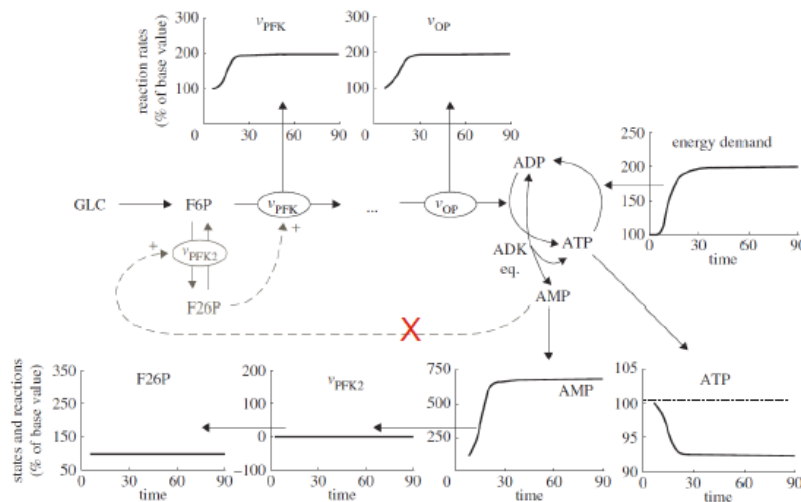
The properties of the enzyme.

The metabolite concentration affecting enzyme activity.

In this light, the flux can be seen as the ultimate representation of the cellular phenotype expressed under certain conditions.

Integral negative feedback: Example on glycolysis, ATP and AMP: In order to ensure a constant production of ATP and no disruption of the glycolytic pathway, one can integrate a secondary pathway that is activated when ATP levels lower and AMP levels increases: Since AMP activates another pathway (v_{PFK2}) it leads to the production of another metabolite X' (F6P becomes F26P = X') which can activate the next step of the glycolytic pathway too.

$$\frac{d[F26P]}{dt} = v_{PFK2} \propto error \Rightarrow v_{PFK} \propto [F6P] \propto \int_t error$$



Kor.: Steady state \Leftrightarrow there is no net flux.

Several allosteric regulators are useful to reduce instabilities (no oscillating behaviours of the system). (Keep the logic of biological systems in mind.)

Constraint based method: Flux Balance Analysis (=: FBA):

Stoichiometric representation in matrix form is needed: Problems to take care of:

Reaction stoichiometry and directionality; Cofactor specificity (ATP, NADPH, NADH etc.); Localizations of reactions (physical compartments, that means, some reactions will occur in mitochondria for example, while others will only occur in the cell plasm); Special reactions (respiratory chains etc.).

Where to gather the data for a reaction of an organism: Textbook, internet (KEGG, Brenda, MetaCyc), scientific literatures, genome sequences.

The biomass flux: Examples: cells, tissues, any biological structures with a function that was given rise to by its components, which is not present in the components themselves.

Definition of biomass flux depends on:

Anabolic biochemistry; Macromolecular synthesis; Composition of macromolecules within a cell; Cellular content of macromolecules.

Side note: When there are ample concentrations of ATP, a reaction can be assumed to be irreversible, since its reversed reactions is very unlikely to occur.

After reconstructing a stoichiometric model, that is, we have matrix S with coefficients of the reaction as its entries and vector v : Solve $Sv = 0$ (this is a quasi-steady state assumption).

Not all solutions are feasible. Some are mathematically correct, but make no biological sense (oftentimes, the signature will be wrong +/-). Only choose the vectors, that are biological sound. “-x” stands for a backward reaction, “+x” for a forward reaction and “0” for no reaction.

Optimal cell states: maximal growth rate; minimal energy consumption (short pathways are preferred over longer pathways when the final product can be reached in less steps); minimal production of toxins and maximal degradation or removal of toxins; maximal use of current energy sources, most efficient production of ATP embedded in many different pathways (not sure if that is correct), minimization of metabolic adaption etc.

All these can be used for the optimization function ϕ and for the weights vector w .

Def. weights vector w : w encodes the optimal feasible state for a cell such that $w_{\text{cellular_objective_function}} = 1$ and $w_{\text{all_other_fluxes}} = 0$. The cellular objective function can be maximal growth for example.

FBA optimization

$$\Phi(v) = w^T \cdot v \rightarrow \max!$$

problem is linearized

s.t.

$$S \cdot v = 0$$

constraints:

$$\alpha_i \leq v_i \leq \beta_i$$

Main assumption in FBA: The system is in steady state for all metabolites: sum of all input fluxes = sum of all output fluxes.

3 inputs are needed for FBA to be performed on a metabolic network: Stoichiometric matrix S , reactions constraints α_i and β_i , optimization function $\phi(v)$.

Kor.: Define $\alpha_i = \beta_i = 0$ for an i in $\{1, \dots, n\}$ in order to create a mutant: no flux is possible then. Apply this definition to essential genes, so essential genes are KO'ed in the mutant case.

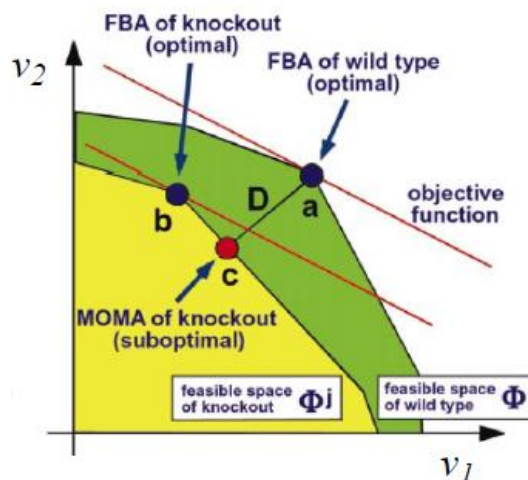
The optimization function should be biologically sound; it cannot be used for genetically engineered strains, since they did not evolve naturally and sometimes it does not make sense, for example, excessive growth means cancer.

Kor.: Pathways are preferred with ATP production (not always the case).

How to observe a pathway P : Use fluorescents on enzymes to determine how active a pathway is and what kind of pathway. Alternatively, apply genetic engineering methods to turn off certain pathways and observe changes (genetic perturbation).

For kinases, understand that they work in networks and less pathways.

Response to perturbation: MOMA: MOMA uses a quadratic objective function.



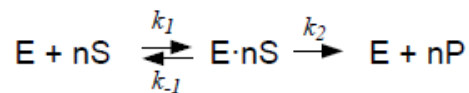
$$D(a, c) = \sum_{i=1}^n (a_i - c_i)^2$$

Def. ultrasensitivity: Response of a biological system that is more sensitive than it is expected from classical hyperbolic Michaelis-Menten kinetics. (The larger the Hill coefficient, the more ultrasensitive is max amplification.)

Kor.: Cooperativity: The binding of a ligand to a site of multimeric enzyme or receptor leads to conformational changes in other subunits of the enzyme or receptor resulting in either an increase of binding affinity or decrease of binding affinity.

Kor.: Multistep ultrasensitivity: The allosteric effector affects more than just one enzyme in a pathway (e.g. activation of forward reaction and inhibition of backward reaction). Higher Hill coefficients can be achieved in cascades of such reactions as seen in the integral negative feedback example of F26P and PFK2 (I think, I am not entirely sure about this claim of mine).

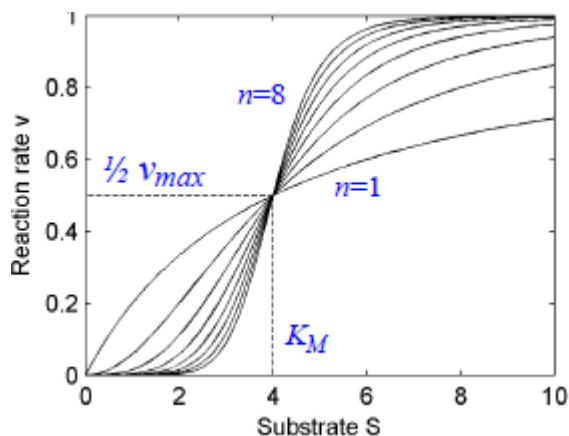
Def. Hill coefficient: A quantification for cooperative binding. It puts the fraction of saturated ligand binding sites in relation with the ligand concentration:



The Hill coefficient is n:

$$v = \frac{v_{max} [S]_0^n}{[S]_0^n + K_M^n}$$

Cooperativity with a Hill coefficient redefines K_M .



Increasing the Hill coefficient leads from a hyperbolic (graded) to a ultrasensitive reaction rate (sigmoidal) signal response characteristic. $n = 1$ is a square root graph.

A higher Hill coefficient models the quickness of a cell's response to small changes (n high \Rightarrow quick response). This can be achieved through cascades for example.

Signaling networks and pathways: A response to a signal must be reversible. Often, a target is phosphorylated and it leads to further cascades. Such a modification has to be reversible such that it can be used again for the same signal transduction or for another signal transduction if it is subject to several targeters. **Phosphatases and autodephosphorylation** are two of these removal mechanisms.

Factors that influence **Signal processing**:

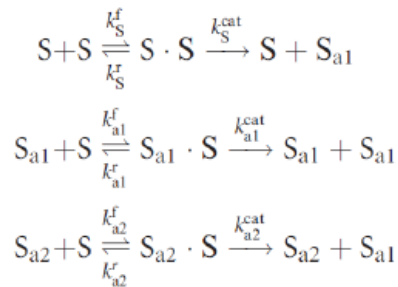
Parameters: Ultrasensitivity, amplification and dampening (e.g. MAPK cascades), dynamics (homeostasis vs memory), dynamic filtering

Regulation structure: feedback/ feed forward, autoregulating mechanisms (dephosphorylation etc.), inhibitory mechanisms, mutual antagonism, adaptive mechanisms

Pathway interaction: Multiple inputs (e.g. carbon sources), competition of targets for a kinase, a target can be influenced by more than one molecule (leads to amplification of activity, allosteric inhibition, complete inhibition)

An autocatalytic network exemplified by Src kinase: S is the substrate, S_{a1} and S_{a2} are the active forms of the substrate.

Autocatalytic network:



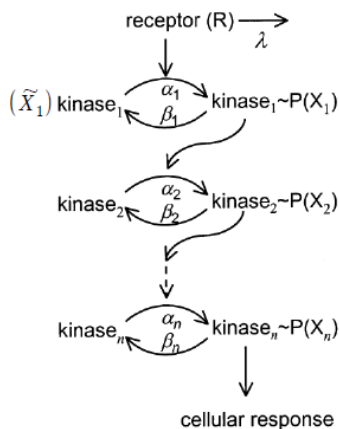
Since the states are not independent, Michaelis-Menten kinetics cannot be applied. Mass-action law has to be used instead:

$$v_3 = \left(\frac{k_S^{\text{cat}}}{K_S} [S] + \frac{k_{a1}^{\text{cat}}}{K_{a1}} [S_{a1}] + \frac{k_{a2}^{\text{cat}}}{K_{a2}} [S_{a2}] \right) [S]$$

Since the autocatalytic network implies positive feedback, a dynamic memory becomes possible depending on the context and the parameters.

Example: Abstract model of MAPK singaling:

An abstraction of the signaling pathway looks like this:



With α_i and β_i being parameters for kinase (alpha) and for phosphates (beta). X_i is an active phosphorylated kinase and X'_i is the inactive dephosphorylated form of the kinase.

Assume mass-action law to derive the following formulas ($R(t)$ cannot be derived from mass-action law):

Kinase activity ($i > 1$):

$$\frac{dX_i}{dt} = v_{p,i} - v_{d,i} = \alpha_i X_{i-1} \tilde{X}_i - \beta_i X_i$$

With total concentration C_i :

$$\frac{dX_i}{dt} = \alpha_i X_{i-1} \left(1 - \frac{X_i}{C_i}\right) - \beta_i X_i$$

Activation by $R(t) = \exp(-\lambda t)$:

$$\frac{dX_1}{dt} = \alpha_1 R(t) \left(1 - \frac{X_1}{C_1}\right) - \beta_1 X_1$$

Def. expected signaling time of arrival τ :

$$\tau = \frac{1}{\lambda} + \sum_{j=1}^n \frac{1}{\beta_j}$$

Def. signal duration θ (variance of expected time):

$$\theta = \sqrt{\frac{1}{\lambda^2} + \sum_{j=1}^n \frac{1}{\beta_j^2}}$$

Def. Average signal amplitude S :

$$S = \frac{S_0 \prod_{k=1}^n \frac{\alpha_k}{\beta_k}}{\sqrt{1 + \lambda^2 \sum_{j=1}^n \frac{1}{\beta_j^2}}}$$

A signal can be a lot faster propagated to the final output, when the cascade has several levels (n high). A cell might still prefer the quickest pathway with fewest intermediate steps (keep in mind that reality is not linear signaling, it is a complex network that seeks efficiency to many different factors such as flexibility, fast adaption to small perturbations, robustness, energetic efficiency. A network is better, when a single molecule (kinase) can be implemented in several pathways. This saves place in the genetic code and it allows for fewer proteins to be synthesized (less energy consumption)).

Systems Biology FS 2017 – PART 2 (Zamboni/Borgwardt)

Def. top-down approach: Given the data one obtains from experiment, one tries to derive interactions from the data.

Def. bottom-up approach: One tries to create a system or a model that can successfully predict the system's behavior.

3 common problems in OMICS data: Overfitting: data is **noisy** and one might identify wrong patterns with the noisy data, leading to very **weak reproducibility (#of detected features >> #of samples)**. In experiments, keep in mind to carry out positive and/or negative controls too.

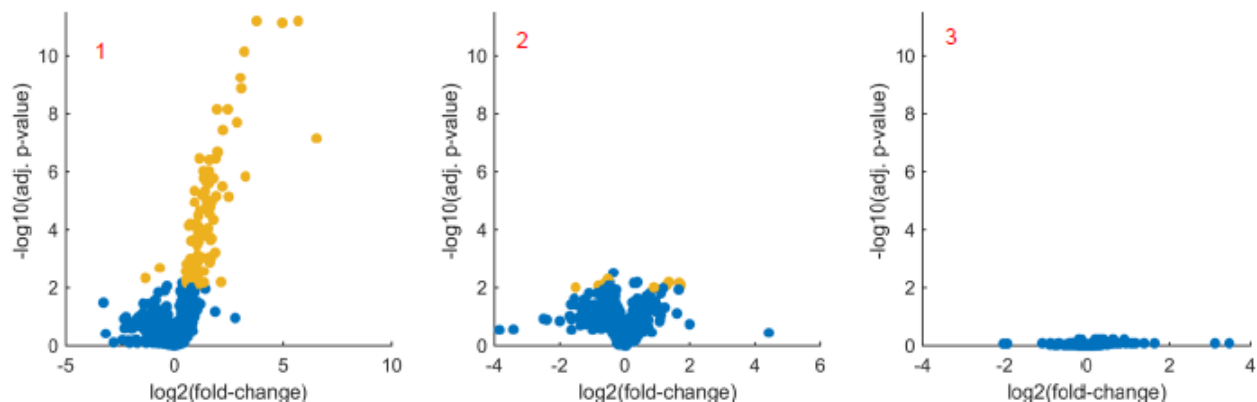
Univariate analysis: Consider only one feature at a time and compare it in the two groups (such as healthy vs sick or wildtype vs mutant etc.).

In order to derive a quantization of the difference of a feature in two groups, calculate the fold change:

$FC = \text{mean}(\text{group}_1) / \text{mean}(\text{group}_2)$ (you can also use $\log_2(FC)$ to have a more symmetric visualization of the fold change).

Naturally, proceed with t-tests and the like (use $\alpha = 5\%$ to get statistically acceptable answers. $\alpha = 1\%$ is nice to have.).

Fold change situations:



Zero significant changes 3	Check data quality Low p-values, high FC > increase number of replicates
Very few significant changes or very few “strong” changes 2	Simplest case > What is the identity of the markers?
A lot of significant changes No clear ranking 1	Quite common, but hard to generate hypotheses > Too many hits > Combination of primary and/or secondary effects > Enrichment analysis!

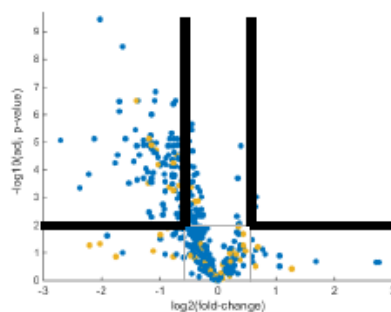
Note: One single feature can be long into multiple groups or processes.

In order to analyze the first scenario, where there are a lot of significant changes one can proceed with an enrichment analysis (gene set enrichment analysis in the context of genes). One can find out whether some features were overrepresented or not.

With a Fisher’s exact test, one can calculate the probability of obtaining the same results again assuming the null hypothesis is correct.

In MATLAB: `[~,p] = fishertest([A B; C D], ‘tail’, ‘right ’)`

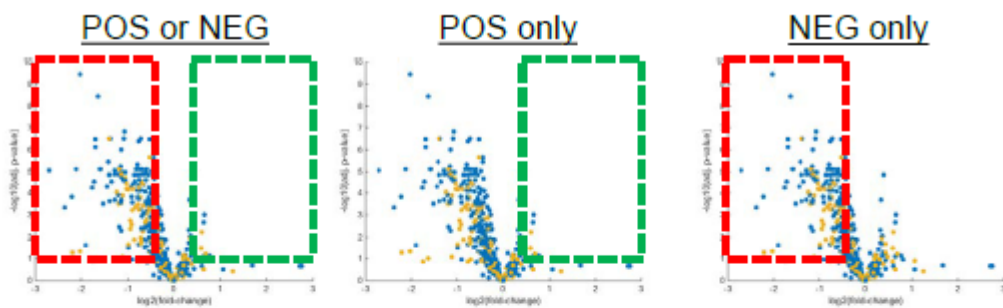
To find the right numbers A, B ,C ,D I have two define the properly according a contingency table:



	In group (G+) ●	Not in group (G-) ●
Significant (S+)	A	B
Not significant (S-)	C	D

GSEA procedure:

1. Decide on what signs do we want to include:



2. Procedure for 1 pathway/group/process:

1. Choose very permissive thresholds (e.g. $\log_2FC > 0.2$ and $p\text{-value} < 0.1$)
2. Rank all hits on either p-value (preferred) or \log_2FC
3. Build contingency tables with either 2, 3, 4, ..., ALL top hits
4. For each table, calculate p-value using Fisher's exact test
5. Keep lowest p-value = "best" enrichment

3. Repeat for all pathways

4. FDR-correction (Benjamini-Hochberg or Storey)

Feature selection

In systems biology (and biology in general), we want to find the major component that influences the behaviour or outcome of a biological system (can be phenotype for example that is strongly influenced by one single gene/transcription factor, a group of genes or maybe miRNA/siRNA or small molecules).

Feature selection is used for: finding correlations or causalities of a particular phenotype regarding its underlying molecular mechanisms; finding lower dimensional models that are easier and cheaper to study; removing noisy features.

Mathematically, evaluate:

$$\arg \max_j (r(\mathbf{x}(:, j), \mathbf{y}))$$

With $r: \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\mathbf{x}(:, j)$ is a vector with the expression levels of the object in question of object j across all n elements (e.g. object = gene and elements = patients, so \mathbf{x} is the vector of the expression levels of gene j across all patients n) and \mathbf{y} is the vector of phenotypes of the patients.

General framework: 1) compute score $r(j)$ for all j (for-loop or while-loop for example).

2) sort features according to score $r(j)$ (highest to lowest for example) and return a sorted list.

There are two approaches to determine the number of most relevant genes:

- 1) Generate a random feature selection z and then chose: $r(j) > r(z)$. (**Probe method**)
- 2) Carry out statistical tests on all scores $r(j)$ and chose those that are statistically significant (value must be lower than significance value α). (**Significance method**)

Due to lots of statistical tests carried out, false positives are prone to occur. Therefore, use corrections such as **bonferroni correction** (divide alpha by number of total tests to be carried out; is rather conservative though which is why fewer significant genes) or **false discovery rate** (less conservative).

Keep in mind that such lists can be rather unstable that is that when you only analyze the subset of the data, it can give a completely different ranking.

To counter the problem, one can either calculate the average rank of all genes in all experiments or calculate the probability to be in the top-k genes.

Limits of univariate selection:

It can only capture the effect of one single gene, it ignores the additivity of multiple genes, it ignores the correlation of genes, it ignores the interaction of genes.

Multivariate selection: solve

$$\arg \min_{\beta} ||\mathbf{y} - \mathbf{X}\beta||_2^2$$

Lasso model:

$$\arg \min_{\beta} ||\mathbf{y} - \mathbf{X}\beta||_2^2 + \lambda_1 ||\beta||_1$$

Idea: Reward solutions (β) in which few entries of β are non-zero.

Concept: This is achieved by minimizing the L1-norm of β :

$$||\beta||_1 = \sum_{i=1}^d |\beta_i|$$

Disadvantage: If there are groups of correlated features, the Lasso often picks just one feature from a group.

Ridge regression:

$$\arg \min_{\beta} ||\mathbf{y} - \mathbf{X}\beta||_2^2 + \lambda_2 ||\beta||_2^2$$

Idea: Reward solutions (β) in which correlated features get similar weights.

Concept: This is achieved by minimizing the L2-norm of β :

$$||\beta||_2 = \sqrt{\sum_{i=1}^d \beta_i^2}$$

Disadvantage: The solution is often not sparse.

The L2-norm rewards reducing larger values more than smaller values, whereas the L1-norm rewards both cases identically.

Elastic net:

$$\arg \min_{\beta} ||\mathbf{y} - \mathbf{X}\beta||_2^2 + \lambda_1 ||\beta||_1 + \lambda_2 ||\beta||_2^2$$

Idea: Reward solutions (β) in which groups of correlated features get similar weights and few weights are non-zero.

Concept: This is achieved by simultaneous minimization of the L1-norm and the L2-norm of β .

Disadvantage: Two parameters have to be set.

Avoid selection bias (overfitting) by only using feature selection on the **training datasets** and making predictions from the previously selected features only on the **separate test dataset**.

Clustering

k-means: most common and popular clustering method: Lloyd's algorithm:

1. Randomly pick k points as initial cluster means.
2. Assign each points to its nearest cluster mean.
3. Recompute the mean of each cluster.
4. Repeat steps 2 and 3 until cluster assignment does not change any more.

Limitations of k-mean:

Number of clusters k has to be specified by the user (how many clusters should I choose?).

Since the first centroids are initialized randomly, there can be different clusters depending on the initial point of initialization (idea for a solution: repeat clustering with the same number of clusters several times until you can identify the most common cluster).

k-means works very well with spherical data distributions but fails with non-spherical ones such as concentric circles – it misses clusters/it cannot identify them.

Solutions of Lloyd's algorithm are local optimum – better solutions might exist.

Graph based clustering:

Principally, one starts with a network consisting of nodes and weighted paths:

- 1) Remove all weights, such that $\text{weights} > \text{user_defined_weight_theta}$
=> all paths removed that have a smaller weight than θ .
- 2) Find all remaining connected components in the resulting graph.
- 3) Each component (a subgraph basically) is a cluster.

Since this also captures noise data (graph clusters are NOT robust to noisy data), we end up with incorrect graphs, which is why use DBSCAN (:= density based spatial clustering of applications of noise).

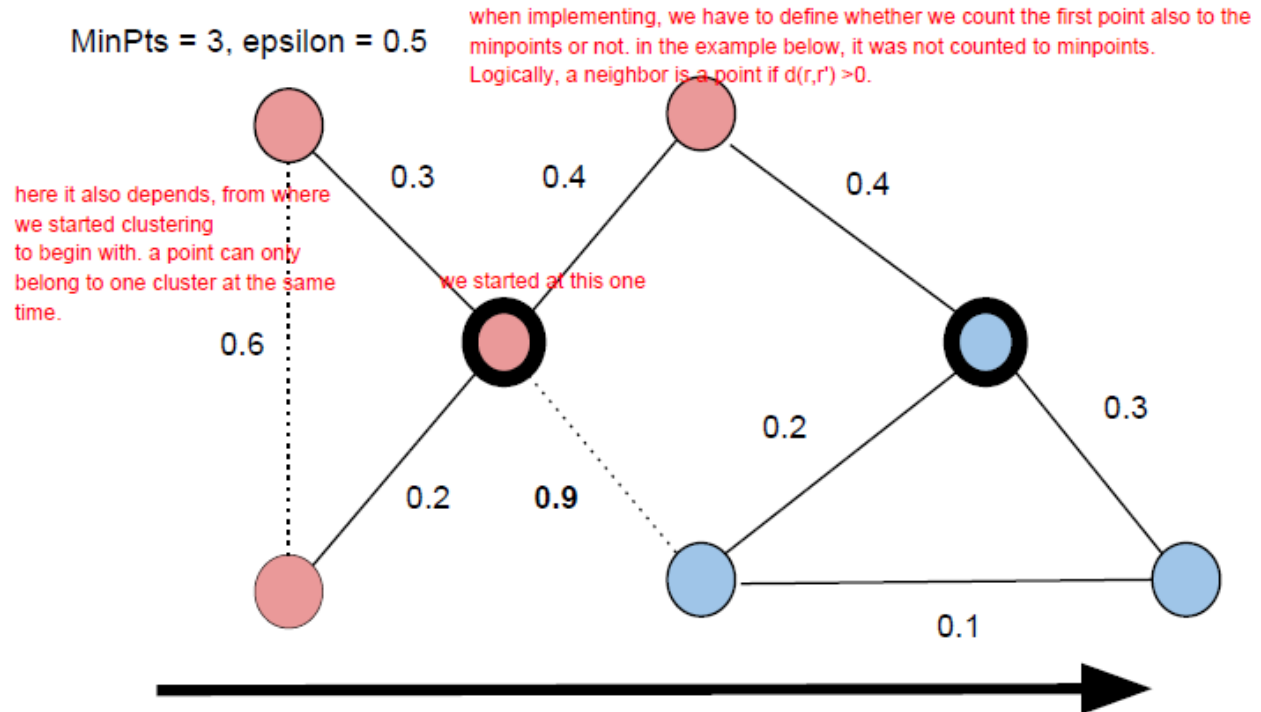
Definitions:

X is core object \Leftrightarrow there are $y > \text{minpoints}$ in $r_epsilon$

- **Core object:** a point is a core object, if there are (MinPts) points within a distance of (epsilon) from this point. Both (MinPts) and (epsilon) are user-defined parameters.
- **Border point:** a point that is not a core object, but in the epsilon-neighborhood of a core object
- **Noise:** All points that are neither a core object nor a border point.

DBSCAN algorithm:

1. Pick a point that has not been assigned to a cluster yet.
2. If it is a core object, add it to a new cluster C (if not, label it as "noise").
3. Add its neighbors to the same cluster C .
4. Check for each of the neighbors whether it is a core object.
 - a. If yes, assign its neighbors to C and perform Step 4 for these neighbors.
 - b. If no, do not further extend the cluster from this point.
5. Return to Step 1 until all points have been clustered.



Advantages of DBSCAN:

No need to set the number of clusters as in k-means.

It is able to find clusters that k-means misses.

It is more robust to noise data than is the naïve graph-based clustering approach.

(Successful in many clustering applications.)

Disadvantages of DBSCAN:

Two parameters have to be set. If density varies as is the case in high-dimensional data it will often miss clusters. The results are initialization dependent.

Hierarchical clustering: This type of clustering can identify clusters in clusters. It can give a sort of 3D insight into the clustering of data (hierarchy).

Principally, every data point is defined as its own cluster, then the two most similar will always form a cluster together. Then all these resulting clusters will be clustered together with their most similar one and so on until there is only one cluster left (trivial case basically).

Distance types in hierarchical clustering: single link, average link, complete link:

$$d_{single} = \min\{d(\mathbf{x}, \mathbf{x}') | \mathbf{x} \in C, \mathbf{x}' \in C'\}$$

$$d_{average} = \text{mean}\{d(\mathbf{x}, \mathbf{x}') | \mathbf{x} \in C, \mathbf{x}' \in C'\}$$

$$d_{complete} = \max\{d(\mathbf{x}, \mathbf{x}') | \mathbf{x} \in C, \mathbf{x}' \in C'\}$$

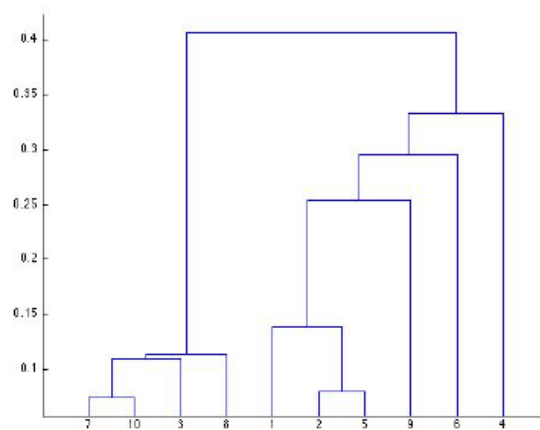
MATLAB

```
z=rand(10,2)
```

```
d = pdist(z)
```

```
l =linkage(d)
```

```
dendrogram(l)
```



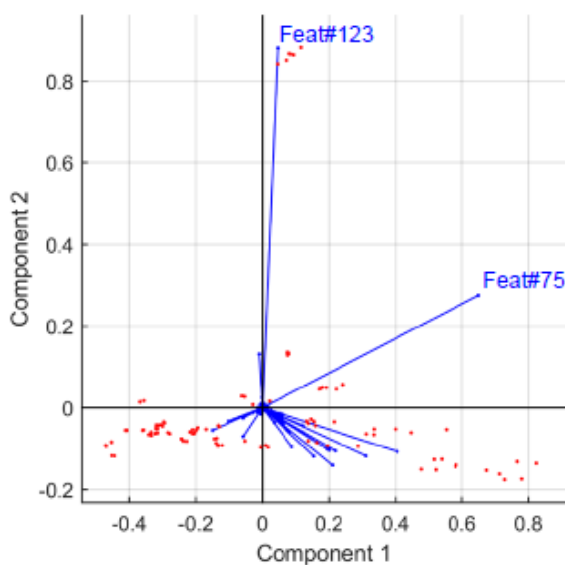
Dimension reduction

We want to analyze multidimensional, multifeature data. The idea is to transform such huge matrices into another form, which is easier to computationally analyze and it still preserves the underlying information (e.g. by removing redundancies and noise data etc.).

Principal component analysis: Method used to reduce dimensions (variables) of a multidimensional dataset for a better visualization, for quality control (it can spot systematic errors) and for preprocessing before other data mining techniques are used. (I have to be able to draw the PC's in a given dataset).

Bi-plot:

Sample scores and Feature loadings are scaled and overlaid to emphasize Associations.



- **Samples** that are close have similar profile
- **Features** that are close are correlating
- **Samples** that are close to a **feature** have high levels of the same feature.
- **Samples** that are opposite to a **feature**, have low levels of the same feature.

- Groups unknown
 - **Clustering**
 - **Dimension reduction** > visual inspection

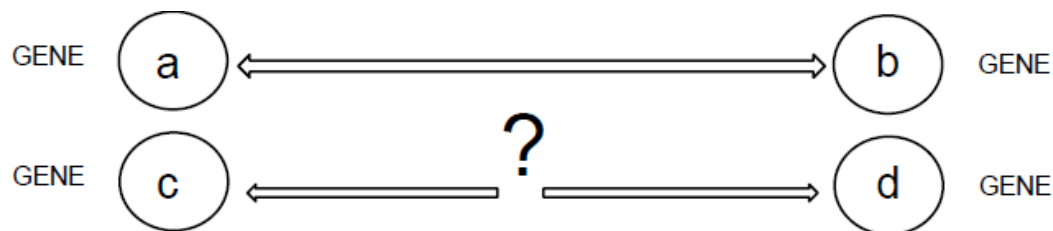
- Groups known
 - **Feature selection**
 - **Univariate**
 - **Multivariate**

Enrichment analysis
Classification

Link prediction

In link prediction, we try to predict missing connections in networks that cannot or are hard to be found by experimental means in the lab (it is also used outside of biology, e.g. in predicting the preference of movies in netflix).

General problem:



Def. supervised link prediction: We are given an example of pre-existing links and learn a prediction model based on these examples.

Def. unsupervised link prediction: The prediction model is based on a rule and predicts according to a rule instead of learning from examples.

Approaches to link prediction

An example for unsupervised link prediction:

Similarity based approach: If $s(a,b) > \theta$, θ is user defined, then infer a link between a and b.

Problem: How to set θ ? Is similarity necessary for an interaction?

Similarity measures that can be used: Pearson's correlation coefficient, mutual information, number of shared neighbours, string kernels that count common subsequences in two proteins sequences (k-mers).

2nd example for unsupervised link prediction:

Similarity based approach (kernel approach): 1) **Tensor pairwise approach**

Given two pairs of nodes (a, b) and (c, d) .

$$k_{\text{tensor}}((a, b), (c, d)) = k_{\text{nodes}}(a, c) k_{\text{nodes}}(b, d) + k_{\text{nodes}}(a, d) k_{\text{nodes}}(b, c);$$

kernel-function is big, when the k_{nodes} are similar:
 $k_{\text{tensor}} = N$, N big \Rightarrow similarity big

This kernel quantifies the similarity of the source and target nodes in both edges, for both directions.

k_{nodes} is a kernel that measures the similarity of two nodes.

2) Metric learning pairwise approach:

Given two pairs of nodes (a, b) and (c, d) . maybe genes react, when gene A has feature A and gene B has feature B and then a reaction between them occurs

$$k_m((a, b), (c, d)) = [(\varphi(a) - \varphi(b))^\top (\varphi(c) - \varphi(d))]$$

$\varphi(g)$ is a vector that describes features of gene/protein g .

A pair (a, b) is similar to a pair (c, d) if $a - b$ is similar to $c - d$, or if $a - b$ is similar to $d - c$.

this function is about differences, instead of the similarity of the starting node and end node as in the previous slide

An example for supervised link prediction:

Cluster based learning: A predictor can learn from already known interactions and non-interactions:

Predict a link between a and b if a and b belong to the same cluster C .

Advantage: More general than pairwise similarity.

Problem: Biologically unrealistic maybe, since interaction only depends of membership of a cluster.

2nd example of supervised link prediction:

Inferring interaction probability between genes: latent group models.

Division into two phases: training phase and prediction phase.

Training phase:

1. Pick a subset S of genes.
2. Cluster genes from S into k different groups based on their gene expression profiles.

3. For each pair of clusters i and j , determine the empirical interaction probability p_{ij} .

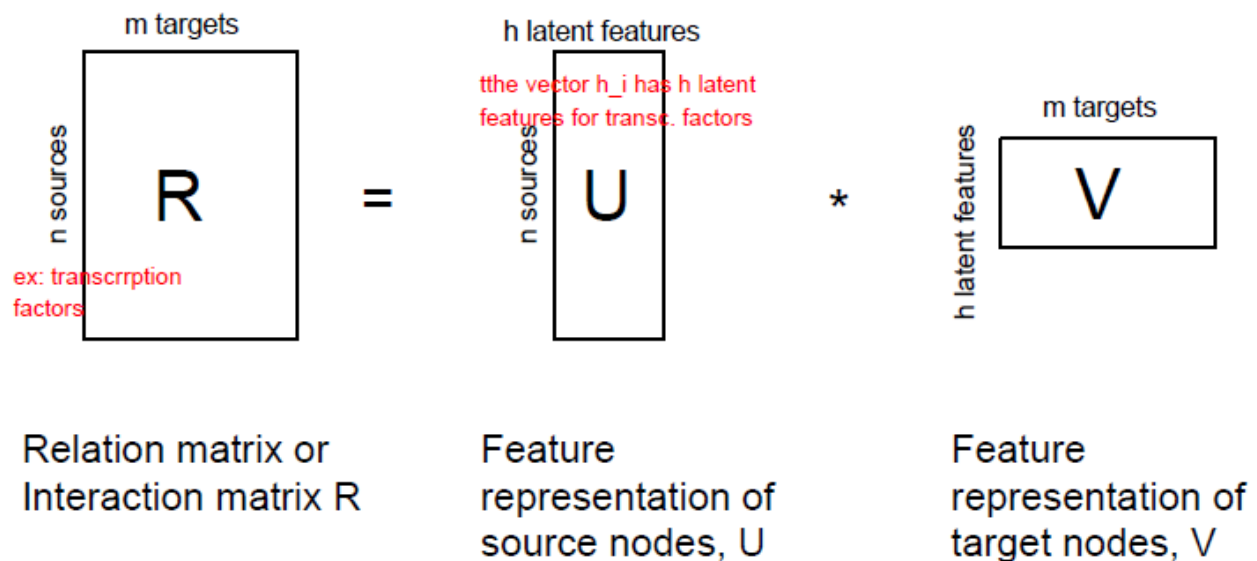
Prediction phase:

1. Given a new pair of genes a and b .
2. Assign a and b to the most similar Clusters C_a and C_b .
3. Report interaction probability $p_{a,b}$ learned in the training phase.

In cluster based link prediction, the interaction of two genes only depends on **one latent variable**, which is cluster membership. Naturally, one can predict links between two genes with several latent or hidden variables. A latent variable can be thought of an imaginary, virtual or theoretical characteristic which has no biological origin, but is computationally derived.

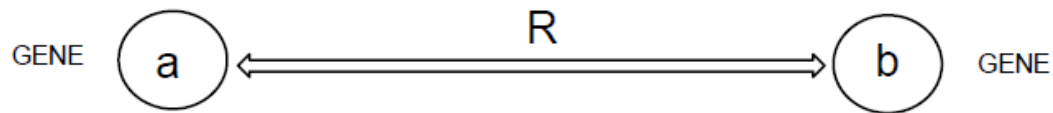
Latent feature based link prediction:

Given a matrix R , which has numbers (data, measurements of expression, activity etc.) as its entries. Then, $R = U \cdot V$, where U has the columns as its latent features and V has its rows as latent features:



R has missing entries (ideally). Matrix completion is equivalent to link prediction in the systems biology interpretation.

Evaluation of link predictions



True Label (R) vs. Predicted Label (R*)	R = 1	R = -1	
R* = 1	TP	FP	
R* = -1	FN	TN	
	P	N	

P = positive
TP = true pos.
FN = false neg.

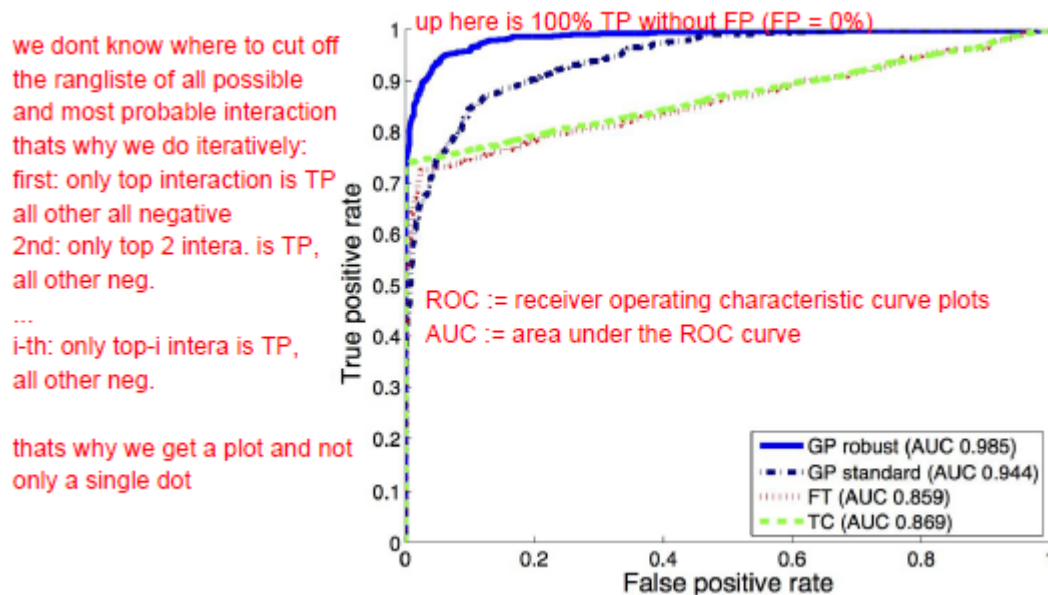
N = negative
FP = false pos.
TN = true neg.

TP+TN is true predictions

Formulas:

- Accuracy = $(TP + TN) / (TP + TN + FP + FN)$
- Sensitivity = TP / P , also called recall or true positive rate
- Specificity = TN / N
- False positive rate = $FP / N = 1 - \text{specificity}$
- Precision = $TP / (TP + FP)$

ROC & AUC



The AUC is the probability that the classifier will correctly discover which one is the positive example.

Problem: Even when AUC is high, a classifier can be rather useless when one class is a lot larger than the other class.

Biological networks

For exemplification's sake, let's proceed with protein-protein interactions (=: PPI) and with predicting PPI (prePPI). One might as well proceed with transcription factor-gene interactions for example.

Databases:

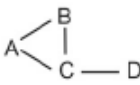
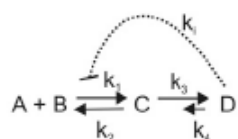
STRING: A database for known and predicted PPI.

Reactome: ...

TRED: ...

(Are these all databases?)

Modelling approaches and requirements

Model class	Level of abstraction	Required information	Example applications
Topological (steady state)	Interaction 	Components and unspecified connections must be known	- Genetic networks - Protein-protein interaction - Metabolite-protein interaction
Stoichiometric (steady state) Steady state	Reaction stoichiometry $A + B \rightleftharpoons C \rightarrow D$	Mass and energy balances thermodynamics (directionality)	Metabolic networks - flux balance analysis - elementary flux modes -
Mechanistic (dynamic) Dynamic	Enzyme mechanism and regulation 	Kinetic parameters	Kinetic models (including regulation)

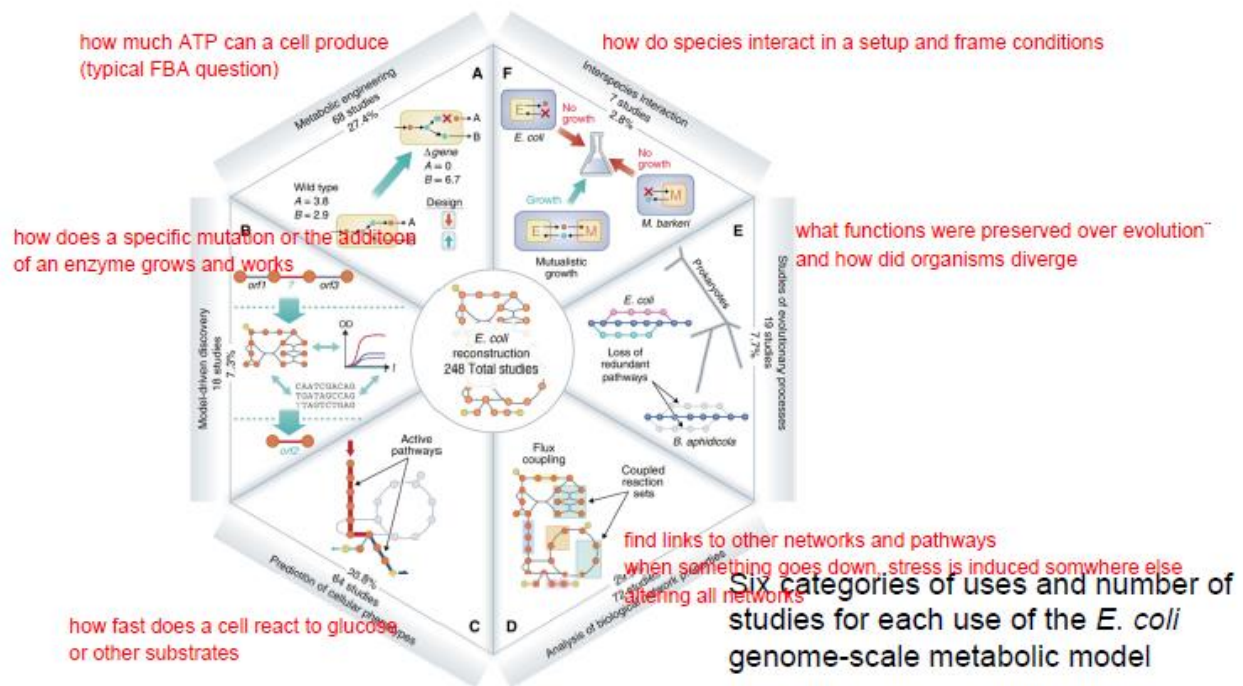
PPI prediction issues: PPI are bad in predicting transient interactions and interactions with lesser studied proteins, which typically have no tertiary structure data, few known interactions, few orthologs in different species or no gene ontology or domain annotations.

Also, PPI are bad in predicting transient interactions based only on correlating gene expression profiles or protein sequences and structures.

Also, PPI can't predict interactions with gene ontology annotations are missing or no known interaction partners have been identified previously.

If training is required, they might acquire an inherent bias of their training sets.

Use of metabolic networks - examples



(FOR THE SECOND PART OF THE LECTURE, CAREFULLY ANALYZE THE EXERCISES SO THAT I KNOW THE RESPECTIVE USAGES, STRENGTHS, APPLICATIONS AND LIMITATIONS OF THE STRATEGIES – SEE PDF “Sysbio-another_exam” FOR MORE CONCRETE INFO.)

PSEUDOCODES:

Score: Calculates the goodness of fit (page 2 in my summary – it is the $\phi(K_m)$ function with unknown parameter K_m)

```

1      score = 0                                %initialize a variable score
2      for i in length(metabolites)
3          for j in 1:n                          %1 to n are the discrete time points
4              temp = ((simulation_data(j,i) – experimental_data(j,i))/stand_deviation(i))**2
5              score = score + temp
6          end
7      end
    
```

k-means (Lloyd’s algorithm): naïve clustering algorithm

```

1      init_ind = randperm(size(data), number_of_clusters)
2      centroids = (data(init_ind), :)
    
```



```

3     previous_ind = zeros(size(data), 1)      %vector that will contain previous indices to check
                                              convergence
4     while true
5         distances = euclidean_dist(data, centroids)
6         new_ind = min(distances, [], 2)      %returns min value of all columns as row vector
7         if all(previous_ind == new_ind), then break%end here
8         else: previous_ind = new_ind
9         for j in 1 to number_of_clusters: centroids(j,:) = mean(data(new_ind==j,:),1); end

```

Add “iter = 1” before the while-loop and “iter += 1” after the for-loop but still inside the while-loop to get the number of iterations till convergence.

Sorting algorithm: Sort a vector (I think something is missing, therefore search in internet or in python)

```

1     Let vec be the vector with values (data)
2?    ind_vec = indices(vector)                %ind_vec contains the indices of vec
3     for k in length(vec)
4         for j in length(vec)-1
5             if vec(k) > vec(j+1), then temp = vec(j+1)
6                 vec(k) = vec(j+1)            %value in vec(k) is saved into vec(j+1)
7                 vec(j+1) = temp              %the smaller value is restored into the spot vec(k)
8         end; end; end

```

simplex algorithm: ??? (what’s a simplex algorithm – Sysbio-5.5 and internet, because it’s not really explained in the pdf)

Flux Variability Analysis (= FVA):

Inputs: S is a m x n stoichiometric matrix (has the reaction rates as its entries)

alpha and beta are two n x 1 vectors with lower and upper boundary conditions for reactions (optimization function)

```

1     F = zeros(n,2)
2     for d in 1:2
3         for z in 1:n
4             w = zeros(n,1)
5             w(z) = (-1)**d
6             Run FBA, that is, solve objective_func(v0,w,S,alpha,beta) = v_z
              %maximize w^T * v, s.t. Sv = 0 under constraints alpha_i and beta_i ( ⇔ FBA)
7             F(z,d) = v_z
8         end
9     end

```

OptKnock: ??? (where to find and what is that)

Link prediction: ... (Sysbio-12)

DSCAN algorithm (Sysbio-10): Noise-robust graph based clustering

Inputs: S is a $m \times m$ matrix with the weighted distances between two nodes. Note that it has 0 on its diagonal and is symmetric.

minpoints and r_eps are user inputs, with minpoints being the minimal points of a point in its neighbourhood with radius r_eps . Cluster is a list that saves the core objects and its neighbours and temp is a list.

```
1      Pick a random point and check if it is a core object  $\Leftrightarrow$  point x has  $\geq$  minpoints in  $r\_eps$ 
1a     count = 0; m = length(S(x,:))
1b     for j in 1:m
1c         if  $S(x,m) > r\_eps$ , then count += 1; add neighbour_index to temp;
1d     if count  $\geq$  minpoints, add all neighbours and x to Cluster      %then x is a core object
1e     else: define x as noise and pick another random point
2      for all neighbours  $n\_i$ , repeat 1a-e: while temp_next or temp  $\neq$  empty
2a     for n in length(temp): set count to 0 again
2b         if  $S(temp(n),m) > r\_eps$ , then count += 1; add neighbour_index to temp_next;
2c     if count  $\geq$  minpoints, add all neighbours and x to Cluster      %then x is a core object
```