Interactive Extraction of Metabolic Subnets - the Netsplitter software implementation

Wynand S Verwoerd

Correspondence: wynand.verwoerd@lincoln.ac.nz

Centre for Advanced Computational Solutions, WF & Molecular Bioscience Department, Lincoln University, Ellesmere Junction Road, Lincoln 7647, New Zealand.

Abstract

Background: Partitioning or clustering is a technique commonly applied to simplify large networks, but for metabolic networks the underlying mass balance constraints need to be taken into account too. A previously published algorithm using a random walk analysis achieves this but cannot by itself provide for user aims and priorities in the partitioning process. The Netsplitter application addresses this in a two-stage strategy. First the random walk algorithm is implemented, identifying metabolite nodes with the potential to partition the network when cut, but allowing the user full interactive control over their selection in each iterative round. In the second stage the user can choose individual subnets to be merged together again, based on information about the content and interconnections between subnets supplied by Netsplitter in graphic and tabular form.

Results: Netsplitter gives an efficient partitioning of genome scale metabolic networks with reduced fragmentation compared to commonly used connectivity based splitting, further improved by the merging step. A complete SBML specification of each subnet is produced by the program. In the case of the *M. pneumoniae*, the bacterial network of 189 reactions and 229 metabolites is reduced to 12 subnets, each of which closely corresponds with recognised biochemical functionality. Application to the network for the mouse *M. musculus*, shows that it works equally well for a network with more than 2000 reactions and metabolites which is partitioned into 162 subnets, reduced by merging to 102 subnets. Again the logical partitioning reflects biochemical reality, including cellular compartments. Subjective assessments of qualitative improvements are confirmed by calculated values of the objective partitioning efficacy index.

Conclusions: Netsplitter is a flexible tool that combines interactive user participation in partitioning with targeted assembly of subnets of a size and composition appropriate for the purpose of a particular study. Due to the efficient display of network structure in both matrix and layout forms, it is feasible to maintain the overview needed for partitioning and merging decisions even for the large and complex genome scale networks of eukaryotic organisms.

Background

In common with other areas of bioinformatics, the increasing range of species and detail covered by available data on metabolic network structure also exacerbates the problem of representing their complexity in a comprehensible way. Conventional network maps used in online databases such as KEGG [1,2] or network analysis tools such as Pathway Tools [3,4] use a pathway-oriented approach. This successfully links relatively small sets of individual chemical reactions into functional units. However, the sheer number of pathways is still overwhelming for genome-size networks of all but the simplest organisms. Moreover, when pathways are assembled into the full network as done for example by Yus *et al* [5], common nodes are usually duplicated to avoid visual tangling, but this detracts from their value as accurate network representations and obscures the complex interconnectivity between pathways.

The network complexity issue is well known from many other applications of graphs and networks. It is commonly handled by algorithms that analyse the connection structure of the network to find clusters or communities of nodes; grouping these together

allows one to interpret the network organisation at a higher level than that of single nodes. In the context of social network analysis for example, there are commonly used software packages such as Pajek [6] for this purpose. Methods for community detection in graphs were comprehensively reviewed by Fortunato [7]. Many such general methods have also found application to biological networks; for example, see a recent discussion by Nayak and De [8]. A general algorithm of particular relevance to the work reported here is Markov Clustering (MCL) [9].

In cellular biology, the Cytoscape [10] software environment is commonly used for network analysis and the community finding algorithms MCODE [11] and NeMo [12] are available for this platform. Comparisons of these with other widely used methods such as Metis and kMetis [13,14] are discussed in reference [12]. A common feature of these module prediction approaches is that they aim to identify densely connected nodes or "cliques" in a network. As found by Zhang *et al* [15] the widely held assumption that cliques identify network modularity, is not satisfied in many real-world networks. In recognition of this fact the Nemo implementation [12] extends the concept to include

dense node pairing structures, and this is successfully applied to protein-protein interaction and regulatory networks.

The Netsplitter algorithm [16] that forms the basis of the software implementation discussed in this article, is based on quite a different premise - it does not aim to identify cliques. Even though its computational approach is based on a matrix representation of random walks as in the MCL, Netsplitter specifically excludes the "inflation" matrix operation which is responsible for clique selection in MCL. Instead, the focus is put on careful selection of the network nodes that form the modular interface, so as to minimize information loss when the network is partitioned. Such boundaries define the modules or subnets, and there is no assumption that subnets are particularly dense in terms of node degree or interconnections.

The reason for this different approach is that networks representing biochemical reactions have special features that need to be taken into account. Firstly, a biochemical network is a bipartite graph, i.e. it has two distinct types of nodes (metabolites and reactions) whereas many general methods including the MCL are designed for simple graphs. More importantly, there is a stoichiometry associated with each reaction, and related mass balance constraints associated with some (but not all) metabolite nodes. Much of the information content of a biochemical network is embodied in the mass balances so these need to be treated carefully in manipulating the network. Another approach that has been proposed [17] for modularising biochemical networks, is based on Petri net t-invariants. For biochemical networks these invariants reduce to so-called elementary modes, and the modularization is achieved by using hierarchical clustering to construct groupings of elementary modes. A network module is then defined as the collection of metabolites and reactions involved in such a grouping. Such subnetworks can overlap or contain each other, which is meaningful in some contexts, but limits their usefulness for network partitioning.

In some contexts however, it is actual partitioning into sub networks that can be of most use rather than just conceptual grouping of network nodes. One of these, is network analysis based on elementary modes and extreme pathways [18] where the total number of modes can become extremely large as network size and complexity increases. A key reason for this, is the combinatorial explosion [19] caused by the fact that if two small networks are joined together sequentially, each pathway in one can be joined to each pathway in the other if the two networks are joined sequentially. It follows that starting from a large network, partitioning it into subnets reverses this and helps to keep mode numbers manageable. Flux balance analysis (FBA) [20] calculations might also benefit from suitable partitioning. There, stoichiometric and mass balance constraints are applied to calculate fluxes, but at least some flux values need to be known as boundary value input to the calculation. If it is possible to partition a network into modules with suitable interface metabolites, the FBA calculation could be limited to a "relevant" network section in a study focussed on a particular aspect of metabolism and avoid the need for boundary conditions that exclusively determine other metabolic aspects. For example, to

study the mitochondrial metabolism in a plant cell, excluding consideration of external fluxes that only impact on photosynthesis in the chloroplast would be a useful simplification. In other cases, the interface that is determined between network partitions may require unknown intracellular fluxes and hence not be useful for FBA. The flexibility allowed by Netsplitter for the user to influence which metabolites define the interface, may assist in producing a partitioning that is usable in this regard.

The Netsplitter and t-invariants methods are similar in avoiding the clique concept, and in using hierarchical clustering to determine node groupings (albeit applied to different aspects of the network representation). However Netsplitter is different in not requiring the determination of elementary modes, and in fact is designed to do network partitioning first, as a facilitating step e.g. to calculate modes afterwards in much smaller subnets.

The problem with partitioning is that some information is inevitably lost. This follows from the fact that the periphery of a metabolic network is defined by the metabolite nodes that do not have associated mass balance constraints. Such nodes are termed "external" because they represent metabolite sources and sinks, i.e. buffered exchanges with the environment. By contrast, there are mass balances constraints associated with "internal" nodes that express the fact that the concentration of the metabolite that the node represents, remains constant in a steady metabolic state, as it is produced by one or more reactions and consumed by others. Partitioning a network amounts to introducing a new periphery, formed by some internal nodes becoming external. More exactly, an internal node is split into two, typically becoming a sink of one subnet and a source in the other. In the process the mass balance constraint associated with the internal node is lost.

This perspective of network partitioning by splitting nodes is another fundamental difference between Netsplitter and most graph oriented modularisation techniques referred to above, where instead, partitioning involves deleting links.

Within this perspective any particular subdivision of the network into separated subnets can be uniquely specified by listing the nodes that were cut, i.e. the internal metabolites that were made external. The partitioning algorithm can be seen as a prescription of how to choose this list. The structure of the network clearly creates the framework for this choice. But more subjective considerations can also play a part. Firstly, the information loss when different nodes are cut may not be considered equally significant. For example, mass balance of commodity and currency molecules such as H_2O , PO_4 , ATP or NADH may be considered sufficiently well represented by simple buffering (so they can all be made external), while that of metabolites on the backbone of a pathway of interest needs to be preserved. In a different study, perhaps focussing on energy transport, mass conservation of ATP may be vital so it would not be an acceptable external. Also more generally, it is desirable to tailor the extent of a subnet to the area of interest in a particular investigation.

Netsplitter can be seen as a further development of strategy used by Schuster *et al* [21] to identify metabolites that are added to the list

of externals. They proposed that metabolite nodes that are connected to many reaction nodes can be considered uninformative. Such metabolites can be identified computationally as those connected to more reaction nodes than an adjustable connectivity threshold, and made external. This was successfully applied [21] to some small networks such as *Mycoplasma pneumoniae* and is implemented as an option in the Yanasquare [22] network analyzer program (which also offers the opposite option of making all metabolites with connectivities lower than the threshold external). Several authors [23-25] have pointed out that this approach relies on a local property (node connectivity) and have proposed alternative network partitioning schemes based on the longer range global properties of a network.

Moreover, it has been found [16,26] that in large networks, the simple connectivity based method tends to either leave a large part of the network unresolved or fragment the network completely, depending on the threshold value chosen.

This was addressed in the previous paper [16] that details the Netsplitter algorithm. It uses a matrix representation of random walks on the network (which also forms the basis of the MCL) to explore long range network structure. It applies hierarchical clustering to the matrix to identify potential subnetworks, as well as candidate metabolites that can be made external to separate them. It also incorporates a connectivity threshold as an essential preliminary filtering step to reclassify the most common currency metabolites. The Netsplitter algorithm succeeds in separating eukaryotic genome-scale metabolic networks with thousands of metabolites and reactions, into a moderate number of subnets of reasonable size. Nevertheless, it still leaves a remnant of small network fragments. Also, while there is a gratifying and somewhat surprising correspondence between the subnets produced purely on the basis of network structure, and the conventional biochemical interpretation of biochemical functionality, the correspondence is not perfect. For example, a functional area such as flavonoid metabolism may be split over several automatically generated subnetworks.

The problem to be faced is that because of the subjective considerations pointed out above, no automated network partitioning can by itself achieve the objective of decomposing a metabolic network into subnets optimal for a particular purpose. User input is needed as well, both to supply domain knowledge not directly present in the network specification and to give guidance about the goal being pursued.

For this reason the central proposition of this article is that a good network splitting algorithm is only half of the solution. In addition to the automated splitting, implementation of the decomposition algorithm should allow for user interaction to guide the selection of external metabolites. A priori setting of algorithmic parameters and preparation of input files can provide some user direction. But to allow full expression of user preferences and biochemical knowledge, the implementation has to be interactive so the user can adapt his control to the situation as it develops.

A major challenge in such an interactive implementation is to present the progress of the network partitioning even for large

and complex networks clearly, so that intelligent decisions can be made by the user. The discussion in the next section sets out the strategies adopted in Netsplitter to make this possible.

A key idea, not previously exploited, is to provide the means to selectively merge subnets created by the automated decomposition. In this way, not only can fragmentation be reduced by reincorporating small fragments into larger subnets, but also targeted assembly of a subnet that represents a metabolic functional area of specific interest becomes possible. This is a substantial advance on previous approaches that stop at structurally based network decomposition.

Implementation:

Netsplitter is available [27] as open source software in the form of *Mathematica* [28] source code. The controlling notebook is simply opened and executed in the *Mathematica* front end, from where a test dataset describing flavonoid metabolism as stipulated below and supplied as a SBML file with the software, can be loaded and analysed.

Selection of externals and visual presentation of network partitioning. The Netsplitter implementation allows the user to specify some settings beforehand, in particular to optionally supply (in an auxiliary input file) a list of metabolites that are/are not to be taken as externals. But the main emphasis is on real-time interactive control.

As illustrated by the more detailed discussion in [16], some aspects of the Netsplitter algorithm design by nature facilitates user involvement. Firstly, the fact that candidate externals are selected progressively over several selection rounds, allows the user to judge individually how appropriate or effective it would be to reclassify a particular metabolite. Secondly, the matrix formulation of the algorithm allows network structure to be visualised and latent subnets to be recognised even in large networks. This is further enhanced by the blocking transformation that is an inherent part of the computational algorithm, but which also gives a striking visual display.

The principle is demonstrated in Figure 1(a) for a relatively small network that describes flavonoid metabolism in *Arabidopsis Thaliana*. Each row and column in the matrix represents an internal metabolite, and matrix element values are shown in colour or grey shading rather than numerically. As shown, the original matrix calculated from the probabilities of random walks connecting metabolite nodes on the network, has been subjected to hierarchical grouping and the subsequent blocking transformation. As detailed in the algorithm exposition [16], isolated subnets form non-overlapping blocks in the matrix. As a visual aid, blocks are displayed in the graphical interface with a blue background. Figure 1(a), for example, shows one large block and two small ones at the lower right. The interior of the large block shows further internal structure. Grey shades are used to indicate the extent to which each particular element can be uniquely assigned to a hierarchical group. The large block in Figure 1(a) shows two distinct but overlapping groupings; within one of those, two more dark grey centres can be distinguished. The dark centres can be interpreted as the cores of partially separated

subnets, while the lighter grey shades identify the metabolites that link them. Using some built-in criteria, Netsplitter identifies the most promising candidates out of these to be made external in order to separate the embryonic blocks, and highlights these in shades of orange. For example, in Figure 1(a) one row and two columns are highlighted as candidates and the corresponding metabolite names are displayed on the right. Furthermore, tick boxes are provided for the user to accept or decline the nominations, with concomitant updating of the matrix display.

The display clearly gives at a glance a detailed overview of the state of separation of the entire network produced by the current set of external metabolites, and Figure 1(b) demonstrates that this is equally effective in a full-scale metabolic network two orders of magnitude bigger.

However, to make intelligent decisions the user may require more detailed information. This is available from the tabbed panels visible at the top of Figure 1(a) where metabolite names are listed, indexed by the sequence number assigned in the matrix display. Apart from looking up individual metabolites, the user can also select one or more blocks by clicking on the display or by entering block numbers in a dialog box, in which case the metabolites belonging to the selected blocks are highlighted in colour on the tabbed listings. This facilitates interpretation of particular blocks in terms of biochemical functionality. Finally, selection of blocks can also be used to restrict further processing to those blocks and prevent further fragmentation of any block where this is undesirable. These opportunities for user inspection and direction are available after each progressive selection round, and the process can also be terminated at any stage where sufficiently fine-grained partitioning has been achieved.

Locating all subnets connected with a particular area of interest by searching through metabolite lists, can be tedious and error-prone for large networks. To assist with that, the Netsplitter implementation provides an additional option to read a previously prepared text file with a list of "target" metabolites of specific interest to the user, and will then highlight these by a coloured background in block listings and also graphically on the matrix display. Quite often, all such functional target metabolites end up in the same structural block; but if not, the subset of blocks containing them is visible at a glance.

In cases where metabolites are explicitly allocated to cellular compartments as part of the input network specifications, this is also incorporated into the display to facilitate recognition of associations between subnets and compartments. Firstly, in the matrix display a background colour is chosen for each compartment and each matrix block is shown on a background shade that reflects the compartment mix of its metabolites. The same colouring is used to display subnet nodes in network layout diagrams. Secondly, for merging subnets the user interface provides buttons for simultaneous selection of all subnets that predominantly contain metabolites in each compartment.

Merging of subnets:

Figure 2 shows a typical distribution of subnet sizes (as reflected in the number of internal metabolites) once the partitioning process

has run its course, for the genome-scale mouse (*M. musculus*) network. Excessive splitting of a network into fragments can be as uninformative as a failure to achieve partitioning; the most useful granularity lies somewhere between the extremes. It is shown in [16] that (disregarding detailed network structure) the optimal partitioning of a network with N internal nodes that minimises effort in interpretation has a uniform distribution of $\sqrt{N}$ subnets each of size $\sqrt{N}$. Clearly biological reality does not reflect such a rigid structure, but the idealized value can serve as a reference point to judge whether a given partitioning has achieved significant simplification.

In this case with $N = 773$ the optimal subnet size and count would be 28. Clearly the partitioned distribution is far from this ideal; it is more like a power law distribution (note the logarithmic vertical scale).

Figure 1 Netsplitter interface for selecting external metabolites

Matrix display of network connectivity structure, where disjoint blocks (indicated by a blue background) correspond to subnets already separated by the current set of external metabolites. Each row or column represents an internal metabolite. Within each block, dark grey shades indicate the cores of further partially separated blocks, and lighter grey shades identify metabolites that link them. Candidate rows and columns proposed by the program for reclassifying as externals, are highlighted in orange and listed explicitly on the right for approval by the user. Tab labels at the top give access to detailed blockwise listings of metabolite categories.

(a) Demonstration example for a small network (flavonoid metabolism, 117 reactions x 137 metabolites) (b) Genome-scale network for *Mus musculus* [29] (2037 reactions x 2179 metabolites).

More than half of the subnets have only one internal metabolite (termed an “orphan” subnet).

However, the distribution can be improved considerably by targeted merging of subnets based on clues supplied by the Netsplitter analysis. An example of this is seen in the second dataset shown in Figure 2, where the fraction of all internal metabolites contained in subnets of size 3 or less, dropped from 24% to 12.5% and the mean subnet size was increased from 8.9 to 16.7.

As a quantitative guideline, Netsplitter displays the calculated efficacy value achieved after each iteration. The concept of efficacy as a quality measure for splitting was introduced in [16] and signifies how closely the $\sqrt{N} \times \sqrt{N}$ partitioning is approached, as a percentage of the theoretical range of variation. For the example in Figure 2, efficacy was improved from 48% to 63% by merging. This improvement indicates a more informative size distribution of subnets, but interpreting whether it is more biological meaningful would require further scrutiny.

Sensible merge decisions are based on recognising significant relationships between subnets – in principle any two subnets can be formally merged, but may remain disjoint if they do not share metabolites. The Netsplitter implementation facilitates this recognition in a number of ways. The highlighting of target metabolites described before allows the user to identify blocks related to a function of interest and merge them if so desired. The colour based display of compartment associations and options for block selection by compartment, also give indications of blocks that are functionally associated and so prone to be merged.

Other clues are based on the structure; for example, it occasionally happens that the node splitting is incomplete, i.e. a metabolite becomes external for one subnet but remains internal in another. The reason for this phenomenon was explained in [16] and it signifies a particularly close relationship between the subnets that is a good basis for merging them. To identify such cases, the printout that Netsplitter produces as a permanent record of a network splitting session, contains a section listing all cases of such internal-external overlaps.

More typically, subnets are linked by sharing external metabolites, and the most significant such links are those where the sharing is exclusive to a pair or small number of subnets, i.e. metabolites that are not highly connected. Information about such sharing is best expressed in a network layout diagram and Netsplitter provides a facility for that as explained in the next section.

A final consideration of more subjective nature is the size of each individual subnet. A small subnet that is merged into a large one reduces fragmentation without adding much to the existing complexity, while merging two large subnets may defeat the general purpose of reducing overall complexity. Observing the change in efficacy before and after a merge operation provides some guidance in this.

When focussing on a particular subnet the user may also prefer not to incorporate specific metabolites deemed as irrelevant, even though the subnet containing them is structurally linked. Such decisions are also easily made by referring to the printout where a blockwise metabolite listing gives an overview of both the size and content of each individual block.

To communicate merging decisions to Netsplitter, the user can select blocks from a graphic matrix display similar to that in Figure 1 by point-and-click or by entering explicit block numbers in an input field.

When implementing a merge decision Netsplitter first combines the internal metabolite sets of the subnets, and then determines all external metabolites that are not connected to any other subnets so that they can be reincorporated into the merged subnet. Performing this process iteratively to convergence, the merge operation becomes the complete inverse of the splitting and hence gives the user complete control to selectively and in a targeted way reverse parts of the splitting process. The overall effect is that network partitioning can be tailored in a very detailed way while still working within the constraints dictated by the inherent network structure.

It is finally noted that merging also makes it possible to relax the strict non-overlapping of subnets produced by the automated splitting, in the sense that different merge decisions can be made

in separate runs of Netsplitter to tailor the partitioning to the functionality of interest in each run.

Subnet and metanet layout diagrams:

Even though the primary purpose of this work is to produce a collection of subnets, connections between subnets are also conceptually important and in addition useful to make decisions about subnet merging. Most commonly this linking is by shared externals.

To give an overview of such linkages, it is useful to introduce the concept of a metanetwork. The idea is that every subnet is contracted into a single “meta reaction” and represented by a single node. Included in this node are all the subnet reactions and internal metabolites as well as all external metabolites that are exclusive to the subnet. The remaining externals are classified as inflows and outflows that are shared between subnets, as well as the crossflows produced by one subnet and feeding into the next.

To present this information in a succinct way, Netsplitter has a facility to produce a metanet layout showing the subnets as well as the shared externals and the reactions that connect them. The metanet obtained from the test dataset is shown in Figure 3. The underlying network represents flavonoid metabolism in a plant species, and has 117 reactions and 137 metabolites. This is quite small compared to the full genome scale network, but even so is hard to represent intelligibly as a layout.

The metanet is in effect a condensed view of the network, and

more detail is seen by zooming in to a subnet such as shown in Figure 4 for subnetwork 1 of the test dataset.

To reflect the bipartite nature of a metabolic network, the layout algorithm chosen by default to construct network diagrams is as a vertically layered directed graph. Consecutive layers alternate between metabolite nodes and reaction nodes, and colour coding is used to distinguish internal metabolites and the three kinds of external metabolites. Subnets appear in reaction layers of the metanet graph, mostly identified by the numbers allocated to their corresponding blocks in the matrix representation. However the smallest fragment subnets containing just a single internal metabolite are labelled by the internal metabolite name and shown in a separate colour on the metanet.

As pointed out in the previous section, sharing of ubiquitous metabolites is not as significant as when subnets are linked by a uniquely shared external metabolite. Hence Netsplitter allows an adjustable ceiling value to be chosen, and metabolite nodes with higher connectivity than the ceiling are not displayed on the network diagram. This filtering greatly reduces complexity in large networks. The implementation also allows an optional threshold subnet size to be set in order to filter out small subnet fragments and only display major components on the metanet.

To demonstrate the use of the metanet diagram for merging, Figure 3 suggests that it would make sense to merge the three orphan subnets labelled by their internal metabolites coumaraldehyde, 4-coumaroylquininate and CPD-3041 with subnet 1 because of the

Figure.3 An example metanet

Metanet for the demonstration flavonoid network. Nodes alternate between subnets (numbered, **black** or orphans labelled in **purple**) and metabolites (coloured) in horizontal layers. Inflow metabolites are labelled in **green**, outflows in **blue** and crossflows that are exchanged between subnets in **red**. The figure was generated from the demonstration model of Figure 1, using the default algorithmic options, finishing at the second selection round and choosing a connectivity cutoff of 12 for the layout display.

shared externals. The position of the shikimate-5P orphan is more ambiguous and a decision whether to merge it with subnet 2 or 3 might depend on the functional roles or even on the fact that with 28 internal metabolites subnet 2 is already quite large, so merging with the smaller subnet 3 might be preferable. Merging subnets 3 and 4 also seem an attractive option.

The same vertically layered layout strategy is also used by Netsplitter to produce subnet diagrams, such as the example shown in Figure 4. By construction, the subnet diagram shows all connections (i.e. reactions) present in the full network for all internal metabolites, but for external metabolites only those connecting to internal nodes in the subnet. External metabolites are once more shown with colour coding to indicate their status as inflows, outflows or exchanges with other subnets or the environment. Variable filtering of high connectivity metabolites is also implemented in the subnetwork diagrams.

The combination of the metanet and subnet layout diagrams allows a detailed but still intelligible overview of the metabolic network structure.

In Figures 3 and 4 there are cases where labelled network nodes partially overlap and visually obscure each other. This is not a problem in the software implementation, as Netsplitter allows the user to interactively adjust the layout size and aspect ratio, font sizes, the relative size of node boxes and the display magnification, all dynamically in real time. In addition, non-layered network layouts can also be selected by mouseclick to disentangle overly dense layouts.

Reversible reactions and the incorporation of flux balance data As mentioned in the background section, preliminary filtering of high connectivity metabolites is found essential in order to reveal the internal structure of the network and allow the automated splitting to proceed. This issue is discussed more fully in [16] where it is interpreted as a percolation phenomenon, i.e. there is a threshold beyond which high connectivity nodes create long range paths that penetrate the entire network, to the detriment of the random walk analysis underlying the Netsplitter algorithm.

For similar reasons the analysis is compromised when the metabolic network contains a substantial percentage of reversible reactions. To avoid that, the default strategy applied by Netsplitter is to ignore reaction reversibility and take all reactions in the direction

implied by the arithmetic sign of elements of the stoichiometry matrix. The user has the alternative choice to let the program augment the stoichiometry matrix by adding columns that represent the opposite direction for all reversible reactions. In practice, this slows down the calculation without having a major effect on the final partitioning for networks with roughly 10% or fewer reversible reactions. Beyond that level, performance of the partitioning can deteriorate and leave large blocks unresolved.

Generally, experience shows that the overall results of the splitting procedure is not very sensitive to the direction of individual reactions, although cases can certainly be constructed where a block will split in two if a single reaction is reversed or changed from reversible to unidirectional. In principle that is no different from any other lacking information; adding or removing a reaction can also produce different blocks, and subnet splitting is only as good as the network it is applied to.

However, reactions are often taken as reversible by default in biochemical databases resulting in much larger percentages of reversible reactions. Realistically, under given metabolic conditions a reaction can only run in one particular direction. So if the majority of reactions are given as reversible in the network specification, that is more likely to be an expression of incomplete knowledge than of physical reality.

For that reason, it is recommended that reaction specifications should not be taken from general listings in biochemical databases, but from complete and specific network model specifications created by human curation, such as available from the Biocompare database. This approach is promoted by Netsplitter taking its primary input from SBML or BioOpt model specification files rather than databases.

As a further measure to alleviate concerns about treating reaction directions realistically, the Netsplitter implementation is also able to read a file of actual flux values calculated in a flux balance (FBA) calculation. As such a calculation reflects a particular metabolic state, Netsplitter takes all non-zero flux values as positive information about reaction directions that overrides the general information in the network specification. Sometimes flux balance calculations for different metabolic states may be available, and it might be of interest how the subnet partitioning differs for the different states. In this situation the sensitivity of the partitioning algorithm to directionality would be an asset rather than a liability.

Finally, it should be pointed out that even where reactions are treated as unidirectional for the purposes of the partitioning, the subnetwork specification files that are produced as the final output of Netsplitter specify such reactions as reversible exactly in accordance with the original specification of the full model. Only if the optional FBA output was used, are any refinements of directions gleaned from that file incorporated in subnet files as well. Generally, special care is taken that all reaction specifications (including e.g. kinetic data that is not relevant for its own calculations) are nevertheless faithfully transmitted to subnet specification files.

Availability and requirements:

The Netsplitter project is hosted at the Bioinformatics Organisation and the software is available as Netsplitter version 1.3 for free download [27] in the form of a *Mathematica* [28] notebook and accompanying User's Guide and a test data set. It is subject to the GNU General Public License version 3, and more details are available on the project website.

Use of the Netsplitter software requires installation of *Mathematica* version 6 or higher or a corresponding version of *Mathematica* Player Pro. Both programs are available commercially from Wolfram Research Inc. Use of Netsplitter is platform independent to the extent that a suitable *Mathematica* version is available for the relevant operating system.

Results and Discussion

Results are presented here for two metabolic networks at opposite ends of the scale in terms of network size and complexity: the bacterium *Mycoplasma pneumoniae* and the eukaryote *Mus musculus*.

The genome-scale *M. pneumoniae* metabolic network:

The metabolic network for this well-studied model organism was thoroughly curated and experimentally validated in a recent study by Yus *et al.* [5]. It consists of 189 reactions and 229 metabolites and it is stated [...] that "it is more linear than that of larger bacteria", based on a small average connectivity degree and a small proportion of branching nodes compared to other model bacteria.

In the network specification, 9 metabolites (H⁺, glycerol, glycerol-3-phosphate, ascorbate, glucose, fructose, ribose, mannose and mannitol) were assigned as extracellular and these were taken as external by Netsplitter. In addition, the connectivity threshold was set to 6 resulting in a further 13 metabolites (H₂O, H⁺, ammonia, ATP, ADP, AMP, phosphate, diphosphate, NAD⁺, NADH, pyruvate, phosphoenolpyruvate and glyceraldehyde 3-phosphate) being taken as external and these were supplemented by a further 7 (coenzyme A, CTP, CMP, UDP, fructose 6-phosphate, glycerol 3-phosphate and phospho-D-ribose_1-diphosphate) specified explicitly in a supplementary input file. The selection of further externals based on the interactive matrix display terminated after 5 rounds in which a further 7 metabolites (CO₂, dCTP, dCDP, ribose 1-phosphate, deoxyribose 1-phosphate, methenyltetrahydrofolate and formyltetrahydrofolate) were nominated by Netsplitter as block separators and reclassified as externals. The combined list of 27 externals leads to a network separation into 15 subnets (including 4 orphans). Finally, the merge steps outlined in the previous section were performed reducing the final number of externals to 14 and the subnet count to 12 including 2 orphans. Correspondingly, the efficacy is improved from 85% to 89% by merging. The result is summarised by the metanet diagram shown in Figure 5. Details of the twelve subnets are listed in Table 1 together with a biochemical function assignment arrived at by recognition of pathways involving their respective internal metabolites.

Figure .5 Metanetwork for *M. pneumoniae*

The metabolic metanetwork after merging, showing all 12 subnetworks and their most significant interconnections by shared metabolites, based on suppressing metabolite nodes with more than 8 connections. Refer to Table 1 to associate numbered subnets with the labelling used by Yus *et al* [5].

Metabolite name abbreviations and the functional descriptions are chosen to conform to those used by Yus *et al* [5] and comparison with their Figure 2, shows a striking correspondence with the partitioning of their metabolic map. They list a total of 18 named sections; 17 of those are uniquely contained in one of the constructed subnetworks as listed in Table 1. The discrepancies are (i) that the single reaction they ascribe to Vitamin B6 metabolism is missing from the table because it only involves external metabolites and so is not a proper subnet, and (ii) the Lipoamide reactions in block 6 appear to be missing from their figure.

While the overall agreement is quite remarkable between the purely computational procedure based on network structure

embodied by Netsplitter, and biochemical wisdom expressed in pathway interpretations, there are some discrepancies in the details. For example, the occurrence of NADP+ and NADPH in block 7 that contains purine metabolism, appears anomalous as these metabolites are not involved in the purine pathways as described e.g. in [5] Figure 2. In fact, NADP+ and NADPH appears there in 3 different functional partitions: first in a redox reaction with thioredoxin in pyrimidine metabolism, second in reducing methylenetetrahydrofolate in folate metabolism, and lastly in phosphorylation reactions with ATP in nicotinate metabolism. Also, thioredoxin plays a role in both purine and pyrimidine metabolism.

As both thioredoxin and NADP⁺ and NADPH act as internal metabolites, the constraints on the framework under which Netsplitter operates dictates that first thioredoxin is allocated uniquely to the purine subnet, and then as a consequence NADP⁺ and NADPH are incorporated there as well. If this is considered unacceptable, a straightforward remedy would be to force NADP⁺ and NADPH as externals by explicitly specifying them as such in the stoichiometry input file. The fact that they were not recognised as such automatically by the program is because firstly there are e.g. reactions that both produce and consume NADPH, and also because in this small network their connectivities remain low. Biochemical knowledge however implies that these molecules are carriers of reduction potential just as NAD⁺ and NADH which are automatically made external, and so it makes sense that explicit use of this knowledge improves the partitioning.

This example illustrates that while the computational algorithm generally produces excellent overall results, some direction from the user may be useful to improve the details. It is one of the goals of the Netsplitter implementation to provide the means by which such fine tuning can be done.

Sub-net	Internal metabolites	Function
1	Ade, Ado, SAH, SAM	Methionine
2	NACD, dNAD	Nicotinate
3	CBP, Cir	Arginine fermentation
4	RIB, R5P, dRu5P, E4P, S7P	Pentose, phosphate
5	AC-P, AcAld, AcCoA, dR5P, Lia, DHLA, SAcDHLiA	Pyruvate, Lipoamide
6	Guo, Gua, dATP, dGTP, 2GP, 3GP, dAMP, dADP, dGMP, dGDP, NADP ⁺ , NADPH, redThio, oxThio, CDP, DGP, GDR, GMP, GTP	Purine nucleotides
7	GLY, DHA, DHAP, Mt1P, G6P, M6P, 3dG6P, FBP, A6P, GA, F1P, proHis, pro-NpPHis	Glycolysis; Carbohydrates; Ascorbate
8	Ado3,5P, dPCoA, PC, pGLY, UDP-GAL, 1aG3P, ACP, ACP-R, apoACP, CDP-DAG, CHO, DAG, FA, G3PC, PAC, PAN4P, pG3P, snG3P, UDP-GLC, UTP	Lipids; Choline; CoA,
9	Asp, Glu	Amino acid
10	THF, 10fTHF, dTTP, Ser, UMP, Ura, Urd, Cyd, dCMP, dCyd, DHE, dTDP, dTMP, dUMP, dUrd, FOR, Thd	Pyrimidine nucleotides; Folate
FMN	FMN	Riboflavin
TPS	TPS	Thiamine

Table. 1 Functional associations of *M. pneumoniae* subnets

Subnets produced by Netsplitter partitioning and subsequent merging, compared to pathway assignments made by Yus [5] *et al.* Abbreviations for metabolite names are taken from [5] and a listing with full names is available as supplementary material to that work.

The genome-scale *M. musculus* metabolic network:

While the previous example shows excellent agreement of the network partitioning with established knowledge in the case of a small and relatively simple bacterial network, it may leave some doubt whether the same analysis can practically be applied to metabolic networks of complex organisms. This is addressed by applying Netsplitter to a metabolic network for the mouse, *Mus musculus*. The network was reconstructed and validated by Quek and Nielsen [29] using flux balance analysis (FBA) calculations to ensure that it reproduces 17 specific biomass components associated with the core metabolism of the cell. The network contains 2016 reactions and 2158 metabolites, giving a stoichiometry matrix two orders of magnitude larger than the *M. Pneumonia* case.

The Netsplitter calculation was done with the default connectivity threshold of 8, which produced automatic classification of 92 external metabolites in addition to 3 that were specified as such as part of the network specification. The facility to specify additional externals by a supplementary input file was not used in this case, but the flux values calculated in the FBA core metabolism simulation [29] were supplied to Netsplitter to refine the allocation of reaction directions. During the externals selection rounds of the calculation all candidates proposed by Netsplitter were accepted and the partitioning allowed to proceed to completion, requiring a total of 36 selection rounds. The result is a total of 167 subnets, 87 of which are orphans, and the detailed size distribution is shown in Figure 2.

Next, the merging procedure was executed according to the structural guidelines outlined in a previous section. A total of 25 merge steps were identified in this way, reducing the subnet count to 102 of which 58 remained as orphans, and efficacy is raised from 48% to 63%. This by no means exhausts the merging options; further merging could be done based on goal oriented user choices or biochemical expertise. A general impression of the complexity of the metanetwork at this point can be gained from Figure 6. While not as simple as the previous examples, it is clearly still feasible to recognise relationships between subnets. The diagram also shows that even in this complex network, it is still possible to ascribe individual functional areas to subnets by recognition of their internal metabolites. Based on such recognition and the layout diagram, additional merge decisions can be made and implemented to produce further simplification.

A notable feature on inspection of internal metabolite lists for the subnets, is that they closely reflect cellular compartmentalisation. Of the 44 subnets with more than one internal metabolite, 35 contain only cytoplasmic metabolites and 6 are exclusively mitochondrial. The remaining 3 subnets contain predominantly metabolites from one compartment, but with a few from the other as well. Bearing in mind that the partitioning algorithm has no access to compartmental allocation of either reactions or metabolites, this illustrates that the logical structure of the network as extracted by Netsplitter largely coincides with its physical structure, although they are not identical. The same observation has also been made in Netsplitter calculations on plant cells such as *Arabidopsis Thaliana* and *Chlamydomonas Reinhardtii* where both mitochondria and chloroplasts are represented

and subnets clearly separate into the 3 compartments although with a few minor overlaps.

Comparison of network splitting and merging:

Figure 7 compares results for three genome scale networks: the small *M. Pneumoniae* and large *M. Musculus* networks discussed in more detail above, as well as an intermediate size network for the plant species *Arabidopsis Thaliana* extracted from the Aracyc 4.5 database [30] and containing 1468 reactions and 1348 metabolites, for which more detailed results were presented in [16]. In each case, results for three calculations are represented: simple connectivity splitting, the Netsplitter splitting algorithm, and a sample application of its merge facility.

In the figure, vertical bar segments each correspond to a subnet with its height proportional to the internal metabolite count. So for each partitioning, the bar height denotes the total number of internal metabolites. Comparing the heights with the reference bar on the right, shows how many internal metabolite mass balance constraints have been sacrificed to achieve each split. Also, the reference bar serves to show the $\sqrt{N}$ -size partitioning of the original network that achieves the maximal theoretical efficacy (ignoring detailed topology). Subnets have been sorted by size, putting small bar segments at the bottom and the large ones on top. The efficacy values are also shown for comparison with the more detailed appraisal afforded by the bar charts.

In each case, the Netsplitter results are shown as two magenta bars (before and after merging) while the yellow bar on the left shows the best partitioning achievable by the connectivity cutoff method and the cyan reference bar on the right is the theoretically ideal partitioning. A common pattern that emerges is that splitting by Netsplitter performs better than the connectivity approach by capturing most of the network in medium-sized subnets that bracket

the ideal size. Also, Netsplitter produces fewer fragmentary subnets (visible as black or grey areas at the bottom) and achieves these improvements mostly while sacrificing fewer mass balance constraints. All of these trends are further enhanced by judicious merging. However, as the network size increases stretching the available range of subnet sizes, it becomes harder to achieve satisfactory partitioning largely because of increasing fragmentation which is improved by the Netsplitter approach, but not eliminated in large networks even after the merging step.

Discussion

In the wider context of module detection in graphs, the online supplement to a recent article by Kovács *et al* [31] lists 18 different definitions and 132 different methods and algorithms for finding communities or modules. The Netsplitter method, however, makes no claim to such general applicability. The way that a subnet is defined by splitting nodes that form the interface between modules, as well as the combination of automated and subjective criteria to choose these nodes, are only relevant to the biochemical networks for which the method was designed. This is in line with the remark made in the concluding paragraphs of the general review by Fortunato [7], that "Therefore in the future we envision the development of domain-specific clustering techniques. The challenge here is to identify the peculiar features of classes of graphs, which are bound to become crucial ingredients in the design of suitable algorithms." Also the provision by Netsplitter of a merging facility to perform targeted reassembly of subnets is particularly aimed at the applications such as elementary mode analysis and FBA for which the software implementation is intended.

When comparing the results from Netsplitter with conventional metabolic maps such as that of Yus *et al* [5] or online databases, it should be borne in mind that those maps are pathway oriented.

Figure. 7 Stacked bar chart display of network splitting

Different partitioning schemes applied to metabolic networks of three organisms. Connectivity partitioning (with optimal cutoff values denoted by C) are shown as **yellow** bars on the left, the Netsplitter partitioning is shown as a pair of **magenta** bars without and with merging, and the theoretical maximal efficacy partitioning of the original network as **cyan** reference bars on the right. Vertical bar segments represent subnets, with height proportional to subnet size. Segments were sorted in order of increasing height.

Consequently they duplicate metabolite nodes when they occur in different pathways and only reactions relevant to a particular pathway are shown in each instance. That is different to the approach followed here: each internal metabolite is allocated uniquely to a particular subnet, and all reactions in which it participates are by definition included in the subnet. Only external metabolite nodes are split between subnets, and even for them the full connectivity is shown in the metanet. This gives a much more direct overview of the interconnection structure, albeit at the price of a larger visual complexity. To compensate, Netsplitter allows suppression of “uninteresting” metabolites and subnets by means of display thresholds.

It may be questioned whether there is any need for network partitioning, especially for FBA where genome-scale calculations are routinely done, albeit mostly for smaller microbial networks. For larger networks such as human metabolism, recent FBA studies do however focus on subnets such as work on mitochondrial metabolism[32]. The need is more pronounced for methods based on elementary mode [EM] analysis such as multiple cut sets [MCS]. It has been

established [33,34] that calculating either EMs or MCS is an NP-hard problem, making it impractical for large networks. However, a subnet modelling flavonoid metabolism has been successfully used [35] to study observed loss-of-function mutations in a plant species by the use of these methods.

As shown by the results section of this article the interface implemented in the program allows flexibility and an intelligible display for both small and large networks. There is some price to pay for this in terms of computational speed. For the small bacterial network, the calculation time is less than 2 seconds on a standard desktop PC with an Intel Core2 Duo processor and 4 Gb memory running at 2.66 GHz, while the mouse network runs for 270 seconds. That suggests a roughly quadratic dependence on the dimension of the stoichiometry matrix. While the response time for the large network is still acceptable for an interactive application, this might become an issue if advances in experimental metabolomics should increase the number of metabolites in a single network by a substantial factor from the current level of around 2000.

A somewhat more detailed account of the Netsplitter program and its practical use, including the compilation of input files and a stepwise report on the analysis of the *M. Pneumoniae* results reported above has been published elsewhere [26].

Conclusions

The combination of an efficient algorithm to partition a metabolic network into structurally determined component subnets, and selective merging of these subnets, sets the scene for constructing a network description at two levels.

First, there is the metanet level where subnets act as meta-reaction nodes and the focus is on how they are connected by shared metabolites.

Secondly, each subnet is composed of internal metabolites that are represented in the full context of all reactions in which they participate in the full network, and a set of external metabolites that define the interface between the subnet and the rest of the network. Moreover, subnets can be allocated to particular biochemical functions and/or cellular compartments by inspection of the metabolites they involve.

In order to realise this two-level description, and tailor it to the particular purpose for which the network partitioning is intended, the user needs to participate in selecting external metabolites and in making decisions about which subnets to merge. A crucial role in this is played by the visual matrix representation of the network structure, as well as interactively configurable network layout maps. Using these tools, it was demonstrated that network partitioning can be applied to genome-scale networks ranging from bacteria to eukaryotes and in the simpler case where comparison is feasible, the results are in agreement with experimentally validated network analysis and its pathway-oriented interpretation.

The overall success of the complementary strategies of splitting and merging, as represented by the objective efficacy measure, shows that partitioning of the small bacterial network achieves 89% of the idealised reference value. Even for the large eukaryotic network a value of 63% or more can be reached, although a certain degree of fragmentation remains.

Authors' contributions

WSV conceived this project, designed the algorithm, programmed the software implementation, performed all calculations and drafted the final manuscript.

Acknowledgements

My appreciation is extended to Dr. Lake-Ee Quek for supplying his data files for the *M. Musculus* model and useful discussions, and Dr. Judith Wodke for the data files used in their *M. Pneumoniae* model.

Competing interests

Authors declare that they have no competing interests.

Article history

Received: 31-Jan-2012 Revised: 27-Feb-2012

Accepted: 10-Apr-2012 Published: 16-Apr-2012

References

1. Kanehisa M, Goto S, Hattori M, Aoki-Kinoshita KF, Itoh M, Kawashima S, et al.: From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res* 2006; 34;(Database issue.);D354-7. | [Article](#) | [PubMed Abstract](#) | [PubMed Fulltext](#)
2. Ogata H, Goto S, Sato K, Fujibuchi W, Bono H, Kanehisa M: KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res* 1999; 27;(1.);29-34. | [Article](#) | [PubMed Abstract](#) | [PubMed Fulltext](#)
3. Karp PD, Paley S, Romero P: The Pathway Tools software. *Bioinformatics* 2002, 18:S225-232. | [Article](#) | [PubMed](#)
4. Paley SM, Karp PD: The Pathway Tools cellular overview diagram and Omics Viewer. *Nucl Acids Res* 2006, 34:3771-3778. | [Article](#) | [PubMed Abstract](#) | [PubMed Fulltext](#)
5. Yus E, Maier T, Michalodimitrakis K, van Noort V, Yamada T, Chen W-H, Wodke JAH, Guell M, Martinez S, Bourgeois R, et al: Impact of Genome Reduction on Bacterial Metabolism and Its Regulation. *Science* 2009, 326:1263-1268. | [Article](#) | [PubMed](#)
6. De Nooy W, Mrvar A, Batagelj V: *Exploratory Social Networking Analysis with Pajek*. Cambridge University Press; 2005. | [Book](#)
7. Lancichinetti A, Fortunato S: Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys Rev E Stat Nonlin Soft Matter Phys* 2009; 80;(1 Pt 2.);016118. | [Article](#) | [PubMed Abstract](#)
8. Nayak L, De RK: An algorithm for modularization of MAPK and calcium signaling pathways: Comparative analysis among different species. *Journal of Biomedical Informatics* 2007, 40:726-749. | [Article](#) | [PubMed Abstract](#)
9. Enright AJ, Van Dongen S, Ouzounis CA: An efficient algorithm for large-scale detection of protein families. *Nucl Acids Res* 2002, 30:1575-1584. | [Article](#) | [PubMed Abstract](#) | [PubMed Fulltext](#)
10. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T: Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research* 2003, 13:2498 - 2504. | [Article](#) | [PubMed Abstract](#) | [PubMed Fulltext](#)
11. Bader GD, Hogue CW: An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics* 2003, 4:2. | [Article](#) | [PubMed Abstract](#) | [PubMed Fulltext](#)
12. Rivera C, Vakili R, Bader J: NeMo: Network Module identification in Cytoscape. *BMC Bioinformatics* 2010, 11:S61. | [Article](#) | [PubMed Abstract](#) | [PubMed Fulltext](#)
13. Abou-Rjeili A, Karypis G: Multilevel algorithms for partitioning power-law graphs. *Parallel and Distributed Processing Symposium, 2006 IPDPS 2006 20th International: 2006* 2006:10. | [Article](#)
14. Karypis G, Kumar V: Multilevel k-way Partitioning Scheme for Irregular Graphs. *Journal of Parallel and Distributed Computing* 1998, 48:96 - 129. | [Article](#)
15. Zhang XS, Wang RS, Wang Y, Wang J, Qiu Y, Wang L, Chen L: Modularity optimization in community detection of complex networks. *Europhysics Letters* 2009, 87:38002. | [Article](#)
16. Verwoerd WS: A new computational method to split large biochemical networks into coherent subnets. *BMC Systems Biology* 2011, 5:25. | [Article](#) | [PubMed Abstract](#) | [PubMed Fulltext](#)
17. Grafahrend-Belau E, Schreiber F, Heiner M, Sackmann A, Junker B, Grunwald S, Speer A, Winder K, Koch I: Modularization of biochemical networks based on classification of Petri net t-invariants. *BMC Bioinformatics* 2008, 9:90. | [Article](#) | [PubMed Abstract](#) | [PubMed Fulltext](#)
18. Papin JA, Stelling J, Price ND, Klamt S, Schuster S, Palsson BO: Comparison of network-based pathway analysis methods. *Trends Biotechnol* 2004, 22:400 - 405. | [Article](#) | [PubMed](#)
19. Klamt S, Stelling J: Combinatorial complexity of pathway analysis in metabolic networks. *Mol Biol Rep* 2002, 29:233-236. | [Article](#) | [PubMed](#)
20. Kauffman KJ, Prakash P, Edwards JS: Advances in flux balance analysis. *Current Opinion in Biotechnology* 2003, 14:491-496. | [Article](#) | [PubMed](#)

21. Schuster S, Pfeiffer T, Moldenhauer F, Koch I, Dandekar T: Exploring the pathway structure of metabolism: decomposition into subnetworks and application to *Mycoplasma pneumoniae*. *Bioinformatics* 2002, 18:351-361. | [Article](#) | [Pubmed](#)
22. Schwarz R, Liang C, Kaleta C, Kuehnel M, Hoffmann E, Kuznetsov S, Hecker M, Griffiths G, Schuster S, Dandekar T: Integrated network reconstruction, visualization and analysis using YANASquare. *BMC Bioinformatics* 2007, 8:313. | [Article](#) | [Pubmed Abstract](#) | [Pubmed Fulltext](#)
23. Gagneur J, Jackson D, Casari G: Hierarchical analysis of dependency in metabolic networks. *Bioinformatics* 2003, 19:1027 - 1034. | [Article](#) | [Pubmed](#)
24. Holme P, Huss M, Jeong HW: Subnetwork hierarchies of biochemical pathways. *Bioinformatics* 2003, 19:532-538. | [Article](#) | [Pubmed](#)
25. Ma HW, Zhao XM, Yuan YJ, Zeng AP: Decomposition of metabolic network into functional modules based on the global connectivity structure of reaction graph. *Bioinformatics* 2004, 20:1870-1876. | [Article](#) | [Pubmed](#)
26. Verwoerd WS: Dissecting metabolic networks into functional subnets. In *A Practical Guide to Bioinformatics Analysis*. Edited by Fung GPC. Brisbane: iConceptPress; 2010: 79-98.
27. Verwoerd WS. (2010). Netsplitter (Version 1.3) [Mathematica Notebook]: Bioinformatics Organisation. | [Book](#)
28. Wolfram S. (2011). *Mathematica* (Version 8.0). Champaign, Illinois Wolfram Research, Inc. Available from | [Book](#)
29. Quek L-E, Nielsen LK: On the Reconstruction of the *Mus musculus* Genome-Scale Metabolic Network Model. *Genome Informatics* 2008, 21:89 - 100. | [Article](#) | [Pubmed](#)
30. Rhee SY, Beavis W, Berardini TZ, Chen G, Dixon D, Doyle A, Garcia-Hernandez M, Huala E, Lander G, Montoya M, et al: The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community. *Nucleic Acids Research* 2003, 31:224-228. | [Article](#) | [Pubmed Abstract](#) | [Pubmed Fulltext](#)
31. Kovács IA, Palotai R, Szalay MS, Csermely P: Community Landscapes: An Integrative Approach to Determine Overlapping Network Module Hierarchy, Identify Key Nodes and Predict Network Dynamics. *PLoS ONE* 2010, 5:e12528. | [Article](#) | [Pubmed Abstract](#) | [Pubmed Fulltext](#)
32. Smith A, Robinson A: A metabolic model of the mitochondrion and its use in modelling diseases of the tricarboxylic acid cycle. *BMC Systems Biology* 2011, 5:102. | [Article](#) | [Pubmed Abstract](#) | [Pubmed Fulltext](#)
33. Acuña V, Chierichetti F, Lacroix V, Marchetti-Spaccamela A, Sagot M-F, Stougie L: Modes and cuts in metabolic networks: Complexity and algorithms. *Biosystems* 2009, 95:51-60. | [Article](#) | [Pubmed](#)
34. Acuña V, Marchetti-Spaccamela A, Sagot M-F, Stougie L: A note on the complexity of finding and enumerating elementary modes. *Biosystems* 2010, 99:210-214. | [Article](#) | [Pubmed](#)
35. Clark S, Verwoerd W: A systems approach to identifying correlated gene targets for the loss of colour pigmentation in plants. *BMC Bioinformatics* 2011, 12:343. | [Article](#) | [Pubmed Abstract](#) | [Pubmed Fulltext](#)

Citation:

Verwoerd WS: **Interactive Extraction of Metabolic Subnets – the Netsplitter software implementation.** *Journal of Molecular Engineering and Systems Biology* 2012, 1:2. <http://dx.doi.org/10.7243/2050-1412-1-2>