

Assiut University Journal of Multidisciplinary Scientific Research (AUNJMSR)
Faculty of Science, Assiut University, Assiut, Egypt.
Printed ISSN: 2812-5029
Online ISSN: 2812-5037
Vol. 51(3): 358 – 373 (2022)
<https://aunj.journals.ekb.eg/>



GEMminer: Text Mining Tool for Genome-Scale Metabolic Models

Zainab Shaban ¹, Yasser Mohammad ² and Ibrahim E. Elsemman ^{3,*}

¹Department of Mathematics, Faculty of Science, Assiut University, Egypt.

²Department of Electrical Engineering, Faculty of Engineering, Assiut University, Egypt.

³Department of Information system, Faculty of Computers and information, Assiut University, Egypt.

*Corresponding Author: e-mail: elsemman@aun.edu.eg

ARTICLE INFO

Article History:

Received: 2022-02-16

Accepted: 2022-07-24

Online: 2022-08-30

Keywords:

Text mining, Genome-scale metabolic models, Flux balance analysis, Protein secretory machinery in yeast

ABSTRACT

Genome-Scale Metabolic Models (GEMs) contain the known chemical reactions in the studied cell or organism. GEMs can be extended to include more biological processes, such as protein translation and secretion. Therefore, GEMs should be integrated with kinetic parameters to couple the metabolic fluxes with the new biological processes. Searching for these kinetics parameters takes a long time. Now millions of scientific papers are available online in the PubMed database. This tremendous amount of current information with text mining approaches creates new opportunities for finding kinetic information that improves GEMs predictions. This paper introduces GEMminer that uses text mining approaches to find this kinetic information based on the user-defined queries and filter keywords. We have validated the accuracy of GEMminer with a manually curated list of GEM papers. Additionally, we have demonstrated that GEMminer can be used to search for kinetic parameters in the protein secretory machinery in yeast, even though there is no available database containing such parameters. These results show that the proposed GEMminer toolbox is a good search tool in PubMed with user-defined queries and filters.

INTRODUCTION

Reconstruction of genome-scale metabolic models (GEMs) plays a key role in understanding how cells use nutrients to build their biomass compositions [1]. The Flux balance analysis (FBA) method converts GEMs into mathematical models to predict the flux distributions maximizing the growth rates [2]. The enzyme turnover (kcat) parameter can reduce the search space in flux spaces to predict the metabolic shift in *E. coli* and yeast at high growth rates [3]. Including other parameters, such as membrane space can increase the power of predictions of FBA [3]. The modeler spends a lot of time and effort to find the appropriate parameters in databases (i.e., Brenda and BioNumbers) and literature (PubMed database) [3-5]. Text mining tools can accelerate the process of

finding these parameters. Many text mining tools have been proposed that extract important information from PubMed [6, 7]. According to a survey on annotation tools, which was carried out on the thirty most important tools, it was found that there is no yet a single tool that can achieve all the desired results in all cases of annotating biomedical literature [8].

The PubMed database contains about 28 million abstracts and 6.5 million full papers [9]. PubMed adopted new AI and natural language processing approaches to extract information that matches the user query. Many programming languages can communicate with PubMed using E-utilities [9]. Therefore, many text-mining tools have been developed to assist in the biocuration process, reconstruction of mathematical models, and biomarker identification. For example, Wei *et al.* [10] developed PubTator which is a web-based text mining tool for assisting biocuration. For each PubMed paper, PubTator can annotate five biomedical concepts: gene, chemical, mutation, species, and disease. Subramani *et al.* [11] developed HPIminer for reconstructing a human protein interaction network. Todorov *et al.* [12] developed INDRA that can assemble information about causal mechanisms into a standard format that can be used to build several predictive models. Chen *et al.* [13] developed a text mining tool that can identify proteomics biomarker identification in kidney diseases. However, these methods are designed for a specific information. For example, Turina *et al.* [14] developed ThermoScan which can identify PubMed papers and report thermodynamic information on protein stability. Systems biologists use BRENDA [5] and BioNumbers [15] databases to get the parameters for GEMs. However, these databases are limited to the information in them, and therefore the user cannot obtain new information related to the next generation of GEMs, such as protein secretory machinery in yeast [16].

In this paper, we developed GEMminer, a tool that uses text mining to find the kinetic information for gene names in published genome-scale metabolic models. GEMminer can facilitate the modelling process by taking the query and filter files as input to extract parameter values from the PubMed database. Firstly, the user inputs a query in an Excel sheet, a text file containing filter keywords, and a path specifying where the user wants to store the output. GEMminer searches in PubMed with a list of IDs and then divides the results into sentences and paragraphs. For each sentence, GEMminer searches for parameters and then focuses on the most relevant literature that has important parameters according to the filter file. Finally, GEMminer generates an HTML page, and the user can navigate the results by important sentences or important paragraphs. For each sentence or paragraph, the researcher can access the paper ID, the result itself, and the relevant keyword(s). Thus, GEMminer can be used to improve model prediction and update the published models with the newly collected information.

MATERIALS AND METHODS

The proposed system is called GEMminer which stands for Genome-Scale Metabolic model text miner (Figure 1). GEMminer takes a list of user-defined queries and a list of filter words. GEMminer then searches the PubMed database based on these queries and retrieves potential articles matching them. Finally, GEMminer finds relevant sentences and paragraphs using the filter keywords and displays them in HTML format. Figure 1

shows the main structure of GEMminer with three main processes: input, analysis, and output. The description of each process is shown in the following subsections.

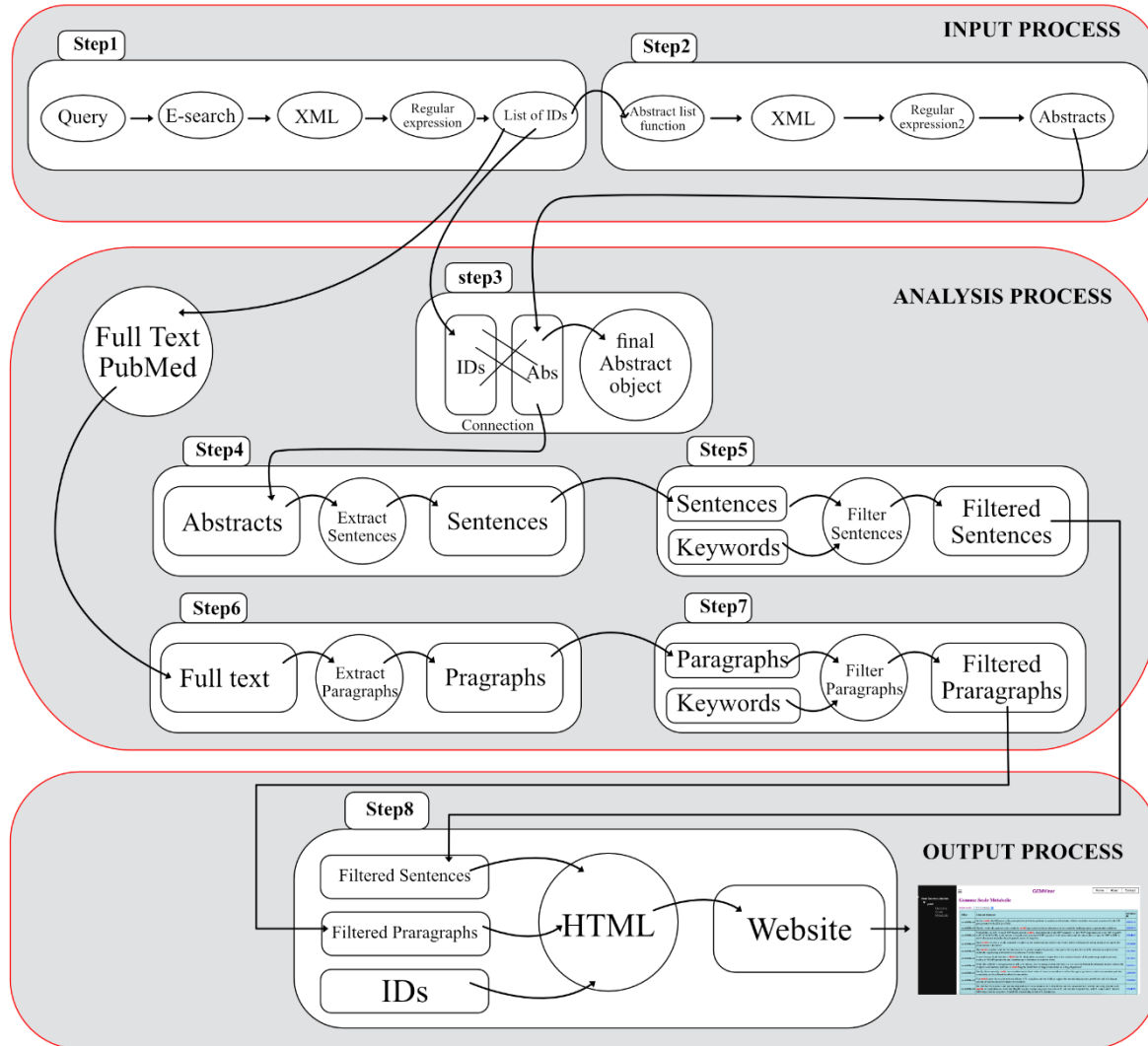


Figure 1. GEMminer structure and steps

1. Input process

In this process, the user gives the system two files: query and filter files. For each query in the file, we convert the query into a URL which is sent to PubMed to retrieve abstracts.

1.1 Step 1 (retrieving list of IDs)

GEMminer formulates URL and sends it to PubMed as [baseURL]/[utilities]/[dbParam]/[termParam]/[retmax]. The description of these parameters is given in Table 1. For example, if we want to know which paper IDs in PubMed have information about the genome scale metabolic, we send the next URL to PubMed:

[http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=PubMed&term=genome scale metabolic&retmax=100000](http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=PubMed&term=genome%20scale%20metabolic&retmax=100000).

PubMed decoded the parameter in this URL and then returned PubMed IDs that match the query as an XML file. Therefore, GEMminer parsed again the returned XML file using regular expression '**\w+\d+\w'** to extract IDs.

Table 1. Description of e-search URL parameters

URL	Name	Example
baseURL	Constant base URL	' http://eutils.ncbi.nlm.nih.gov/entrez/eutils/ '
Eutilities	e-utilities server	'efetch.fcgi?'
dbParam	Database Parameter	'db=PubMed';
termParam	Term Parameter	'term= Genome scale metabolic';
Retmax	maximum number of returned articles	100000

1.2 Step 2 (Retrieving Abstracts)

For each extracted paper ID from Step 1, GEMminer uses the e-fetch server in e-utilities to retrieve abstracts corresponding to the requested paper ID. Firstly, GEMminer formulates the URL and sends it to PubMed as [baseURL]/[utilities]/[dbParam]/[ID] the description of these parameters is given in Table 2. For example, <https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=PubMed&id=23300594>

After using the e-fetch server, PubMed returns all abstracts that match the query as an XML file. In this step, there can be many attempts to connect to the PubMed database, so this can lead to disconnection. To solve this technical problem, we decide on the "step_number" represents the number of concurrent calls followed by sleep before repeating the process. This reduces the probability of broken connections. According to our experiments, we found that the connection disconnected after 1000 calls. We then extracted abstracts using a regular expression, because the returned XML file contained unstructured data for abstract information. Each line in the XML file was parsed file with the regular expression '**abstract "\.[\s]*" PMID \d+'**'. GEMminer then extracted the abstract paragraph(s) from XML content.

Table 2. Description of e-fetch URL Parameters

URL section	Name	Example
baseURL	Constant base URL	' http://eutils.ncbi.nlm.nih.gov/entrez/eutils/ '
Eutilities	e-utilities server	'efetch.fcgi?';

dbParam	Database Parameter	'db=PubMed';
ID	Article ID in PubMed	'23300594'

2. *Analysis process*

GEMMiner identifies potential articles based on all sentences in the abstract or the full text if available that match any keyword or regular expression in the filter file.

2.1 Step 3 (Categorizing and Organizing)

GEMMiner collects abstracts and IDs from previous steps as a table for each ID with its corresponding abstract. From step 3, we get an object containing ID numbers and abstracts.

2.2 Step 4 (Segmenting abstracts into sentences)

GEMMiner segments each extracted abstract into sentences using the regular expression `\\.\\s]*[A-Z]'`. This expression splits abstract into statements.

2.3 Step 5 (Finding Potential Sentences)

GEMMiner scans each sentence in all abstracts to find the potential sentences that match the filter file. For example, if the query file contains one sentence which is 'genome scale metabolic', and the filter file contains the word model. It selects all the sentences having information about models. Filter words are determined by experts and searched for in all sentences generated in the previous step.

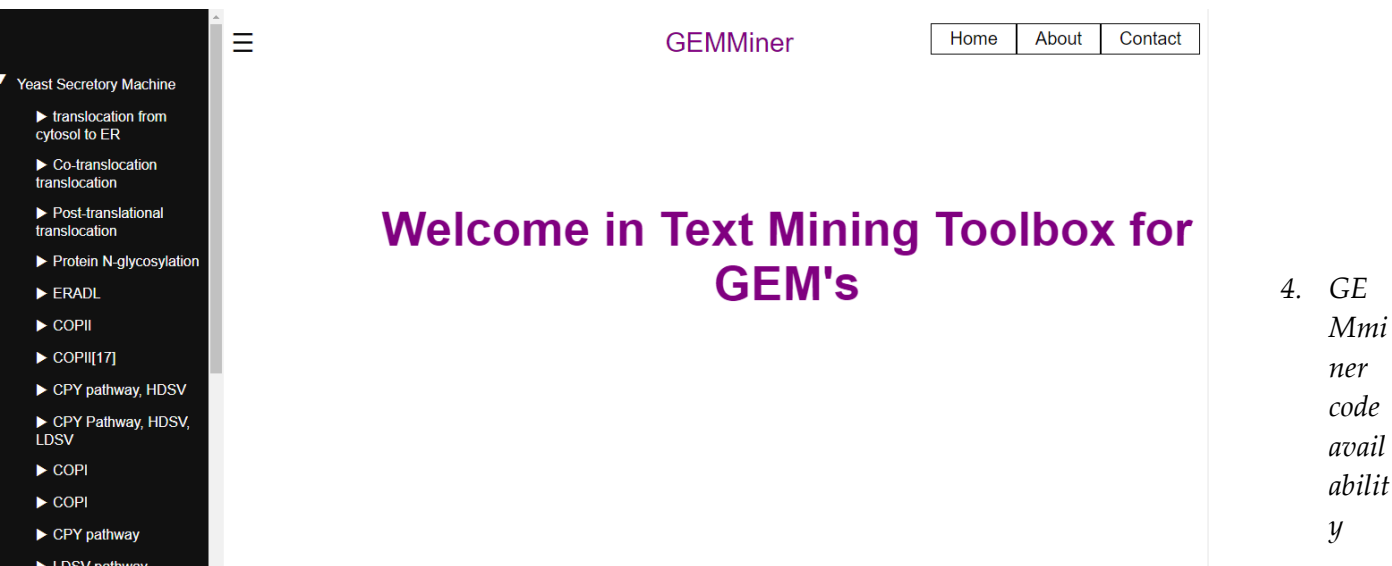
2.4 Step 6-7 (Finding potential paragraphs in full text articles)

PubMed has about 6.5 million available full-text papers that can be downloaded. GEMMiner can extract the available paper using the PMCID number that exists in the abstract XML file (From Step 4). Additionally, GEMMiner can download these papers using e-fetch. GEMMiner then segments the downloaded paper into paragraphs and finds the relevant paragraphs within the returned XML files using the filter words.

3. *Output Process*

In this process, GEMMiner displays the results in an HTML file divided into two sections with a responsive sidebar (Figure 2). The HTML format displays the results clearly and is easy to navigate and search through. GEMMiner can store these results in a path selected by the user. GEMMiner uses HTML, CSS, and JavaScript to design the sidebar and interactive elements for organizing the data. The user can select the query from the left frame and access the potential sentences and paragraphs in the right frame.

The user can also get more details about these sentences/paragraphs by simply opening the PubMed abstract in a new tab in the browser from the provided interface.



GEMminer tool is available for everyone who wants to use it as we have made its code available on GitHub at the following link <https://github.com/zainab9291/GEMminer>. The user can use GEMminer by calling only one function *finalInfoWeb* as:

```
finalInfoWeb(queryFile, filterFile, path)
```

Figure 2. GEMminer interface

Where queryFile is an Excel file containing user-defined queries, filterFile is a text file containing user-defined filter files, and the path is the folder path where GEMminer stores the generated HTML files.

RESULTS AND DISCUSSION

In this section, we validate the proposed system in a series of experiments designed to show the effectiveness of the proposed approach.

1. Validating GEMminer with known PubMed ID

A Genome-scale metabolic model can be represented as an Excel sheet with reactions, metabolites, and genes. Each GEM paper may report the reaction, metabolite, and gene numbers. Monk *et al.* [17] collected manually published GEMs until the year of 2012. They reported 110 GEMs, where each GEM is associated with a PubMed ID. In our first experiment, we used this list to validate that if we know the published IDs, GEMminer can decide that this paper is a GEM model. First, we searched in abstracts or articles for models, reactions, and metabolites. To customize GEMminer to this task, we

used the filters defined in Table 3. In this validation experiment we did not use a query but directly used the IDs in the Monk list.

We used these PubMed IDs in Monk’s dataset and compared our result with the Monk list (Table 4). The manually curated results contained 115 papers, but they represented 110 unique PubMed IDs. Our filters could mark 97 of them as GEM models leading to an accuracy of 97/110 (about 88%). The remaining articles, (13), may exist but with no full text in PubMed. These experiments validated the use of the regular expressions in Table 3 for the given task and show that it has a false-negative rate of 11.8%.

Table 3. Regular expression used in filter results of paper IDs

Target	Regular expression
Model	i\w+\d+
Reaction	((\d+)(\d+),(\d+)) reaction
Metabolite	((\d+)(\d+),(\d+)) (metabolites w+ metabolites)

Table 4. Accuracy of GEMminer results we found 75 models, 81 reactions and 80 metabolites

Filter	Result
Model	75/110
Reactions	81/110
Metabolite	80/110
Total (at least one of the three)	97/110

2. Validating GEMminer with a user-defined query

First, we searched for the query “*Genome-Scale metabolic*”. We used an extra filter for data “until 2012”, as the same date for the Monk’s dataset. The number of returned IDs from a general query was 24,683. Our system was able to reduce it to 558 PubMed IDs. We found 86 out of 110 PubMed IDs are shared with the manual curation list so GEMminer was able to achieve a success rate of up to 80% (Figure 3).

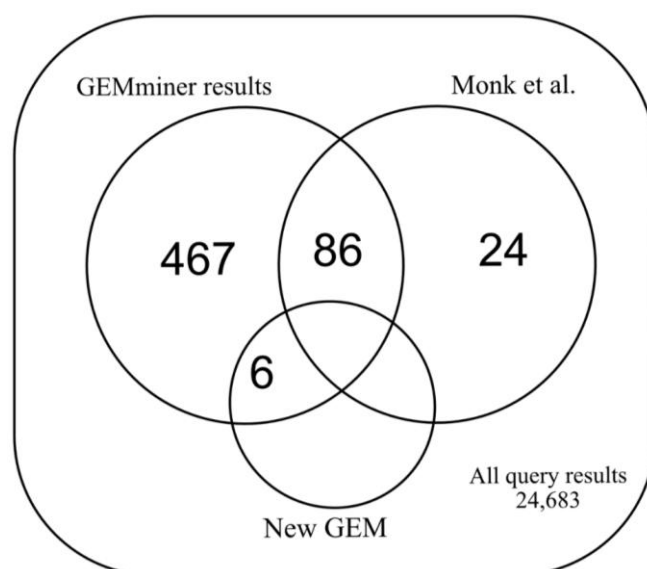


Figure 3. GEMminer results it has 86 papers shared with Monk et al. and has six new papers

GEMminer was not able to retrieve 24 PubMed IDs because 18 of them did not use the term Genome scale metabolic which was changed to another term such as metabolic network, genome-based predictions and *in silico* so they did not appear in the general query. Additionally, six of them appeared in the query but did not match any of our keyword filters. GEMminer reported 467 articles as GEMs but they were not related to the reconstructed models because these articles considered as application of GEMs. Finally, GEMminer was able to extract six new GEM names that were not reported in this list (Table 5). In conclusion, our system was able to achieve a success rate of 80 %. This suggests that it can help in facilitating the reconstruction of genome-scale metabolic models (GEMs). We believe that the accuracy can be increased when the user chooses a suitable query statement. Additionally, GEMminer found six new models. Therefore, we believe that the GEMminer can be used as an assistance tool that assists systems biologists to find the relevant information quickly.

Table 5. The new six models that GEMminer able to find them

Sentence	PubMed ID
We addressed the complexity of cyanobacterial ETC by developing a genome-scale model for the diazotrophic cyanobacterium, <i>Cyanothece</i> sp.	22529767
Furthermore, the relationships among cotton species or between cotton and other model plant systems were analyzed.	21829504
The human heart-specific model provides valuable information for the studies of cardiac activity and development of CVD	22057009
These essentiality data, the library and the Nmb_iTM560 model are powerful and widely applicable resources for the study of meningococcal metabolism and physiology.	22208880
The developed model forms a basis for rational analysis and design of <i>Scheffersomyces stipitis</i> metabolic network for the production of fuels and chemicals from lignocellulosic biomass	22356827
In addition, <i>in silico</i> simulations with the iCG238 model have enabled a set of carbon and nitrogen sources to be defined, which would also support a viable phenotype in terms of biomass production in the strain Pam, which lacks the first three steps of the tricarboxylic acid cycle.	22376077

3. GEMminer with protein secretory machinery in yeast

In this section, we applied our GEMminer to extract the kinetic parameters in the protein secretory network in yeast. To the best of our knowledge, this kinetic information is not reported in any systems biology databases, such as BioNumbers and BRENDA databases. Therefore, this example validates the ability to use GEMminer in a real application. The yeast secretory machinery pathway describes how proteins can be transmitted from the cytoplasm to the cell membrane and extracellular (Figure 4). The secretory machinery pathway starts on the surface of Endoplasmic reticulum (ER), where there are ribosomes that decode mRNA to protein molecules. These proteins are transported from one compartment to the other compartment as a vesicle, each compartment has a unique mechanism to bud and fuse a new cargo or vesicle. These vesicles can be transmitted directly across the ER membrane. After that, it is moved from the ER to the Golgi compartment and finally from the Golgi compartment back to the ER.

3.1 Preparing GEMminer

Our developed toolbox GEMminer can be run in personal computers. To use GEMminer we first should prepare two files: Query and filter files. The query file contains two columns the first column is the process name, and the second column is the gene name. Table 6 shows the used queries, and Table 7 shows the filter keywords we used. The Filter file contains all keywords that the user wants to highlight in returned abstracts.

Table 6. Queries used in GEMminer for secretory machinery

Process Name	Gene Name
Budding on ER and COPII	Sar1 Sec23 Sec24 Sec13 Sec31 Sec12 COPII
Tethering on Golgi	Ypt1 Sed5 Sec22 Bos1 Uso1 Hrr25 Bet1 TRAPPI AND Golgi
Budding on Golgi and COPI	Arf1 Ret2 Ret3 Sec26 Sec21 Gea1 Gea2 Ret1 Sec27 COPI
Tethering on ER	Sec22 Sec20 Ufe1 Use1 Tip20 Sec39 Dsl1

Table 7. Filter keyword used in GEMminer

Filter
Km km, Kinetics kinetics, Kcat kcat, Vmax vmax, (ATP) ATP, vesicle Vesicles, dissociation\skd, kd KD, turnover Turnover, structure Structure, copy copies, ((d+) (d+)).(d+))\s(s)

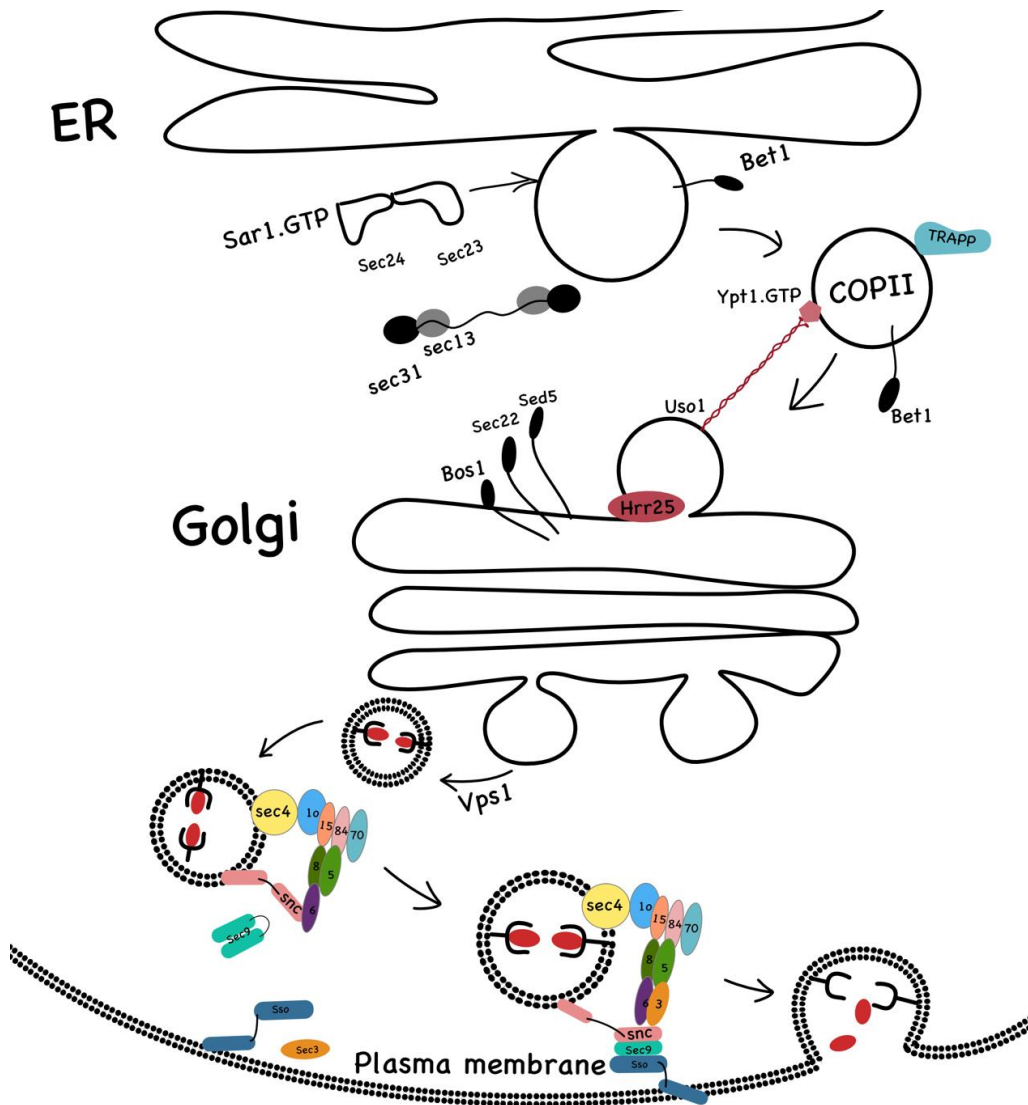


Figure 4. Secretory machinery stages in yeast. We divided this machine into four stages. This figure was adopted from Feizi et al. [18], Barlowe et al. [19], and Mei et al. [20]

3.2 Stage 1: budding from ER

In this stage, the buds on surfaces of ER membrane are activated by the protein Sar1. These buds will form the cargo (COPII) that transfers the proteins from ER to Golgi compartment. Figure 5 shows the main statements that were extracted by GEMminer. First, GEMminer reported the article PubMed: 17604721 that explained the structure of COPII. GEMminer can find the number of Sar1 in COPII as 30-48 copies, the area covered by Sar, and COPII diameter as 40 nm. Furthermore, GEMminer reported article PubMed: 15665868 which measured the reaction time for Sar1-GTP as 200 seconds. This rate is low, so the cell should make a lot of Sar1p. We found that yeast has a high number of Sar1 copies per cell. Furthermore, GEMminer reported also PubMed: 11389436 that measured the assembly and disassembly rate of Sec23/Sec24 and Sec13/Sec31 as 10 sec.

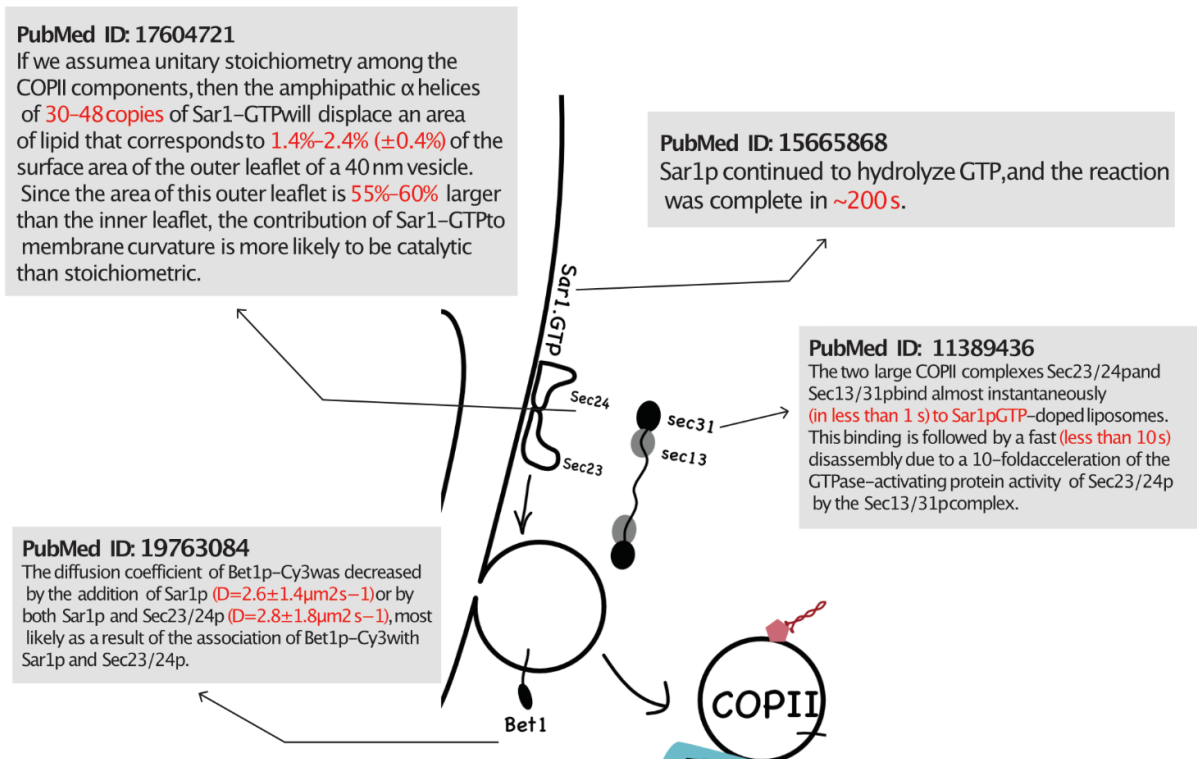


Figure 5. Stage 1 budding from ER, GEMminer found four potential papers.

3.3 Stage 2: tethering on Golgi

The secreted vesicles From ER to Golgi needs the TRAPPI complex that guides COPII to the Golgi surface (Figure 6). GEMminer reported the article PubMed: 1198998 that estimated the copy numbers of proteins Bet1, Bos1, Sec22, Sfpt1, Gos1, Ykt6, and Vti1 per one copy of COPII as 360, 510, 537, 600, 580, 220, and 280, respectively [21]. Furthermore, our GEMminer reported the article PubMed: 17336899 explained the function of TRAPPI complex that guides COPII to bind with the Golgi membrane [22]. GEMminer reported the PubMed article 11419942 that explains the role of TRAPPI complex in Ypt1 activation and the rate of this reaction is 0.01 per second. Finally, GEMminer found two articles PubMed ID: 2132587 and 2174157 that describe the role of protein Hrr25 in COPII docking on the Golgi surface.

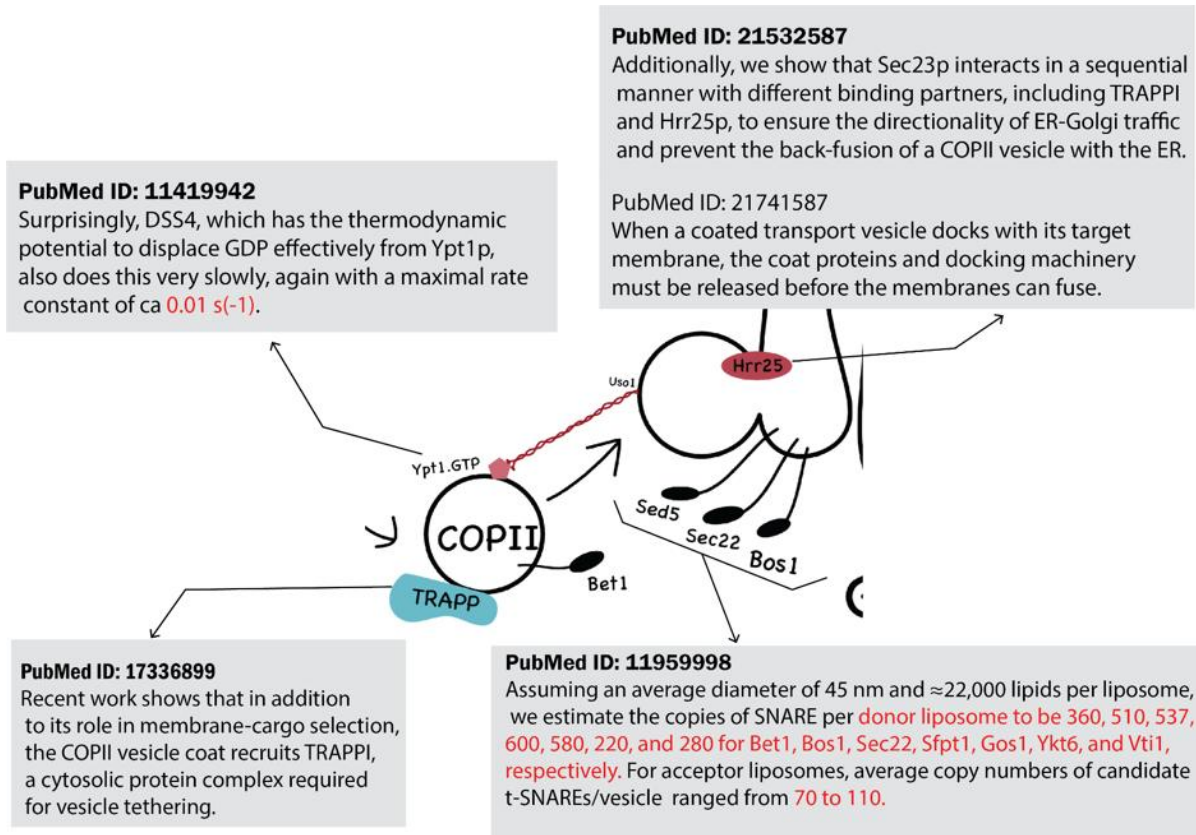


Figure 6. Stage 2 Tethering on Golgi, GEMminer found four potential papers

3.4 Stage 3: budding from TGN-network to plasma membrane

The protein Vps1 is required for the fission of secreted vesicles from TGN-network to the plasma membrane (Figure 7). Vps1 is considered as a scissor in TGN-network. Our GEMminer reported PubMed ID: 7862158 described the function of the protein Vps1. Additionally, Our GEMminer reported PubMed ID: 15550248 explained the working mechanism of VPS1 on TGN-network and the vacuole fission and fusion processes.

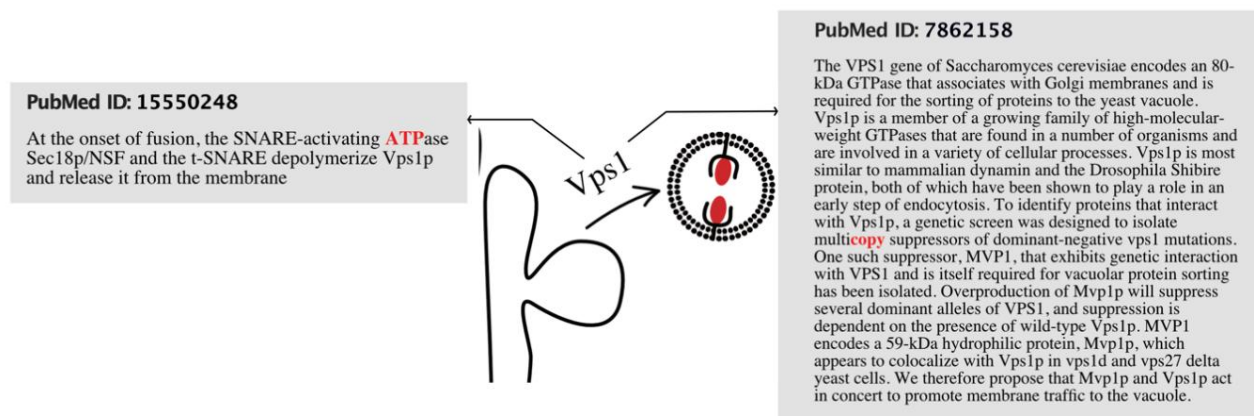


Figure 7. Stage 3 budding from Golgi to plasma membrane, GEMminer found two potential papers

3.5 Stage 4: plasma membrane fusion

The secreted vesicles from the trans-Golgi Network (TGN) bind with the exocyst complex to assist it in binding in the plasma membrane (Figure 8). Our GEMminer reported the paper PubMed ID 16826234 that discussed the requirement of exocyst complex in vesicle fusion through the plasma membrane, and the paper PubMed ID 28112172 that discussed the mechanism of use exocyst complex in vesicle fusion in the plasma membrane. Additionally, our GEMminer reported the Paper PubMed ID 15583031 that estimates the assembly rate of the exocyst complex as 57 seconds. Furthermore, our GEMminer reported the PubMed ID 10048921 that described the role of the proteins Sso1, Sso2, and Sec9 in the assembly of the exocyst complex. During using GEMminer for exocyst complex, GEMminer reported some kinetics values that can be used in yeast growth in lactate. GEMminer reported the paper PubMed ID 11964174 that measured the abundance of the lactate transporter as 1674 molecule/cell and the catalytic rate is 123 per second. We think that this information is important because yeast grows slowly in lactate. Therefore, the yeast growth rate is limited by making the Jen1 transporter and not caused by the low transporting rate of Jen1.

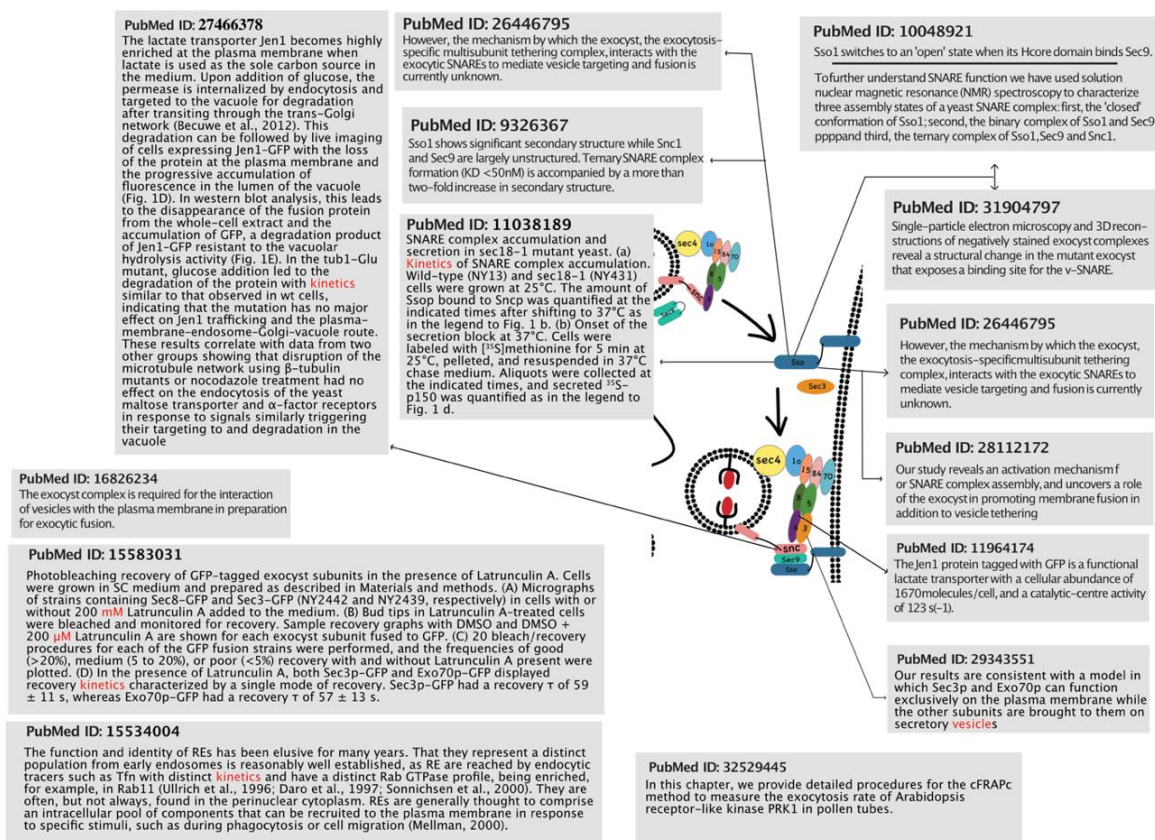


Figure 8. Stage 4 Plasma membrane fusion, GEMminer found 14 potential papers

CONCLUSION

In this paper, we have developed a text mining toolbox that assists system biologists to extract kinetic parameter values needed to improve GEM predictions by reducing the search space. In traditional methods, system biologists may spend a long time in searching manually for kinetics parameters, and biological functions in BioNumbers, and BRENDA databases to extract these parameters. However, these databases are limited, and the user cannot go beyond them. This paper developed GEMminer toolbox that can retrieve kinetic parameters from PubMed database directly. What distinguishes GEMminer from other biological text mining tools is that the user can search for any parameters with any query and filter the user's needs. The results are organized in a simple, structured, and user-friendly format.

GEMminer was validated with a manual-curated list of 113 GEMs and was able to retrieve 86 of them shared with this list and identified six new GEMs (that were already available at the time the list was compiled). Additionally, GEMminer has been validated in a real application for extracting useful kinetics parameters in a protein secretory network in yeast. The validation results suggest that our GEMminer toolbox can be used as an assistance tool for the curation of GEMs and a search tool in PubMed with user-defined queries and filters.

REFERENCES

- [1] Förster, J., I. Famili, P. Fu, B.Ø. Palsson, and J. Nielsen, Genome-Scale Reconstruction of the *Saccharomyces cerevisiae* Metabolic Network. *Genome Research*, 2003. 13(2): p. 244.
- [2] Nielsen, J., Systems biology of lipid metabolism: From yeast to human. *FEBS Letters*, 2009. 583(24): p. 3905.
- [3] Palsson, B., *Systems Biology*. 2015: Cambridge University Press.
- [4] Schomburg, I., L. Jeske, M. Ulbrich, S. Placzek, A. Chang, and D. Schomburg, The BRENDA enzyme information system-From a database to an expert system. *J Biotechnol*, 2017. 261: p. 194.
- [5] Simon, C., K. Davidsen, C. Hansen, E. Seymour, M.B. Barnkob, and L.R. Olsen, BioReader: a text mining tool for performing classification of biomedical literature. *BMC Bioinformatics*, 2019. 19(Suppl 13): p. 57.

-
- [6] Hearst, M.A., Untangling text data mining, in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. 1999, Association for Computational Linguistics: College Park, Maryland. p. 3.
- [7] Kong, X. and M.B. Gerstein, Text mining systems biology: Turning the microscope back on the observer. *Current Opinion in Systems Biology*, 2018. 11: p. 117.
- [8] Neves, M. and U. Leser, A survey on annotation tools for the biomedical literature. *Brief Bioinform*, 2014. 15(2): p. 327.
- [9] Fiorini, N., R. Leaman, D.J. Lipman, and Z. Lu, How user intelligence is improving PubMed. *Nature biotechnology*, 2018.
- [10] Wei, C.H., A. Allot, R. Leaman, and Z. Lu, PubTator central: automated concept annotation for biomedical full text articles. *Nucleic Acids Res*, 2019. 47(W1): p. W587.
- [11] Subramani, S., R. Kalpana, P.M. Monickaraj, and J. Natarajan, HPIminer: A text mining system for building and visualizing human protein interaction networks and pathways. *Journal of Biomedical Informatics*, 2015. 54: p. 121.
- [12] Todorov, P.V., B.M. Gyori, J.A. Bachman, and P.K. Sorger, INDRA-IPM: interactive pathway modeling using natural language with automated assembly. *Bioinformatics*, 2019. 35(21): p. 4501.
- [13] Chen, L., W. Su, H. Chen, D.Q. Chen, M. Wang, Y. Guo, and Y.Y. Zhao, Proteomics for Biomarker Identification and Clinical Application in Kidney Disease. *Adv Clin Chem*, 2018. 85: p. 91.
- [14] Turina, P., P. Fariselli, and E. Capriotti, ThermoScan: Semi-automatic Identification of Protein Stability Data From PubMed. *Frontiers in Molecular Biosciences*, 2021. 8(144).
- [15] Milo, R., P. Jorgensen, U. Moran, G. Weber, and M. Springer, BioNumbers--the database of key numbers in molecular and cell biology. *Nucleic Acids Res*, 2010. 38(Database issue): p. D750.
- [16] Li, F., Y. Chen, Q. Qi, Y. Wang, L. Yuan, M. Huang, I.E. Elsemman, A. Feizi, E.J. Kerkhoven, and J. Nielsen, Improving recombinant protein production by yeast through genome-scale modeling using proteome constraints. *Nature Communications*, 2022. 13(1): p. 2969.
- [17] Monk, J., J. Nogales, and B.O. Palsson, Optimizing genome-scale network reconstructions. *Nature Biotechnology*, 2014. 32(5): p. 447.

- [18] Feizi, A., T. Österlund, D. Petranovic, S. Bordel, and J. Nielsen, Genome-Scale Modeling of the Protein Secretory Machinery in Yeast. *PLOS ONE*, 2013. 8(5): p. e63284.
- [19] Barlowe, C.K. and E.A. Miller, Secretory protein biogenesis and traffic in the early secretory pathway. *Genetics*, 2013. 193(2): p. 383.
- [20] Mei, K. and W. Guo, The exocyst complex. *Current Biology*, 2018. 28(17): p. R922.
- [21] Parlati, F., O. Varlamov, K. Paz, J.A. McNew, D. Hurtado, T.H. Söllner, and J.E. Rothman, Distinct SNARE complexes mediating membrane fusion in Golgi transport based on combinatorial specificity. *Proc Natl Acad Sci U S A*, 2002. 99(8): p. 5424.
- [22] Haas, A.K. and F.A. Barr, COP sets TRAPP for vesicles. *Dev Cell*, 2007. 12(3): p. 326.