Subject: Similarity analysis using "find similar compounds..." - slow analysis of libraries
Posted by SM2020 on Tue, 24 Nov 2020 11:35:22 GMT
View Forum Message <> Reply to Message

Hi,

I'm carrying out a similarity analysis by comparing x2 cpd libraries (each ~ 16K) using the "find similar compounds in file..." option using the FragFP descriptor. I am selecting a similarity limit cut-off of ~30% (using the slide bar selector).
I want to obtain the nearest neighbour measure for a given library, so nearest neighbour values can be binned and plotted (note: I realise there are other methods for library comparison as well in DW).

The process is taking an extremely long time on my (admittedly) old computer (4-core/8 threads, i7-3615QM, 16GB Ram, MacOS) - > 24 hrs.

With that in mind, a couple of general questions...

1) What improvements (if any) would help in speeding up this process?
Would increasing the accessible RAM help, or am I generally limited by processor speeds for such analyses?

2) What general hardware upgrades and/or software upgrades would you suggest for dramatically speeding up these types of analyses (<< 24 hrs) in DW?
Faster processor/s with more cores/threads? More RAM? Both...? Other...?

Best Wishes and many thanks.

Subject: Re: Similarity analysis using "find similar compounds..." - slow analysis of libraries
Posted by SM2020 on Tue, 24 Nov 2020 20:33:00 GMT
View Forum Message <> Reply to Message

Update:
My original post is a bit detail-lite and I am aware there are many factors that could be playing a role in the slow performance experienced (not necessarily DW related).
Looking in the MacOS activity monitor, the processor is not being fully consumed (~70% idle), RAM is not fully consumed (~8 GB of 16GB being used only ~4-5 GB being used by DW (this allocation could be user increased) but the task is currently sitting at ~50 hrs. It is progressing, but slowly.
The MacOS service manager, launchd, seems very disk heavy during this task (>11GB written vs. ~500 MB read), as is kernel_task (~10 GB written/100 MB read).

Ultimately, I would like to get some general feedback to help distinguish whether the task I have set DW is indeed very resource intensive (particularly for my computer) or whether there are other

likely issues causing bottlenecks in task execution.
More generally, has any other user experienced similar >>24hr run times when running similar processes (with similar library sizes) and if so, has anyone performed any system upgrades/changes which have reduced computation times?

Best

PS. In case it is not clear, I think DW is a fantastic program!

Subject: Re: Similarity analysis using "find similar compounds..." - slow analysis of libraries
Posted by thomas on Thu, 26 Nov 2020 17:16:19 GMT
View Forum Message <> Reply to Message

I could confirm this with two files containing 16000 random structures each. Two reasons together cause the incredibly bad performance (many numbers and bad sorting):

A new column receives all individual similarity values to compounds of the second file, which are higher than the given threshold. With a 30% limit these are about 70% of the other file's molecules. Therefore, about 8000 to 12000 similarity values were put into every row of the first file. The individual similarity values are kept sorted by DataWarrior. Unfortunately, this was done in a very inefficient way by keeping the cell content as text, converting it to numbers to find the insert position for the new value. With just a few values this is not a problem, but with thousands of values, this was very expensive. I have updated the source code. The next development release early December will contain the fix. Now it takes about 2 minutes.

If your original idea was to get a distribution of all mutual similarities between any pair of molecules in a file, than there is a much faster way: Launch DataWarrior in development mode (with Java option '-Ddevelopment=true'). The you get a few undocumented additional items in the chemistry menu. 'Compare Descriptor Similarity Distribution' counts and shows a graph of the Gaussian like curve of all similarity values binned into 1-percent bins.

Subject: Re: Similarity analysis using "find similar compounds..." - slow analysis of libraries
Posted by SM2020 on Fri, 27 Nov 2020 00:35:09 GMT
View Forum Message <> Reply to Message

Thanks very much Thomas!! - that will be a really useful update. Looking forward to that rolling out soon.

Getting a distribution of molecules within a single file would be great, but comparing 2 discrete files - and plotting a resulting similarity distribution - is more what I was thinking about. In my mind they are slightly different (but I could be mistaken).

Maybe I'm misunderstanding, but using your suggestion, could I just put all the cpds into a single file, run the analysis you suggest and then (assuming there is a unique identifier for the cpds - e.g. vendor), use DW to highlight by vendor and then assess the distribution in the resulting histogram? Would this compound selection step be possible (like in other graphical displays in DW)?

I was essentially thinking to keep one library constant (reference) and compare against "other" libraries thereby generating a binned histogram of closet neighbour similarities for each "other" library compared to the reference - one measure of similarity between "other" libraries and a reference library is then gauged by the apparent shift of distributions between (near) 0 and 1.

Either way, I'd like to try both analyses. It's great that DW (will) allow it.

Is there a hard upper limit for the no. of compounds analysed in these ways (i.e. file size, > 16K cpds?? or is it driven by available computational resources to carry out the task?

Thanks again for your effort.

Subject: Re: Similarity analysis using "find similar compounds..." - slow analysis of libraries
Posted by thomas on Sat, 28 Nov 2020 23:05:43 GMT
View Forum Message <> Reply to Message

I have changed the algorithm again. Now it just writes the highest similarity and the number of compounds with similarity above threshold into the open file. This accelerates again. Now a 16k by 16k comparison takes about 10 seconds on my computer. A million by a million would probably take around 12 hours.

Putting two sets into one file and use the procedure I suggested earlier would not work for your purpose, because it just uses the complete similarity matrix of all compounds without considering sets. But I hope, the current update works for you. It can be downloaded as development patch from the download page after clicking the 'read and understood' box. The links are in the small print.

This task actually does not need much memory. It basically needs to fit the first file into the memory, which should be possible with even a few million compounds, if the -Xmx setting is adapted. The second file's size doesn't matter much, because it is processed row by row.

Please let me know, if there are problems of any kind.

Subject: Re: Similarity analysis using "find similar compounds..." - slow analysis of libraries
Posted by SM2020 on Sun, 29 Nov 2020 14:07:13 GMT
View Forum Message <> Reply to Message

That great Thomas! Thanks very much.

I've installed and it seems to be working well, including using other descriptors as the basis for the similarity comparison.
I played around with workflows in Knime to achieve similar results but now that this is incorporated into the DW development version, it so much more straightforward(!. Brilliant.

Thanks again.

Subject: Re: Similarity analysis using "find similar compounds..." - slow analysis of libraries
Posted by SM2020 on Thu, 04 Feb 2021 10:11:11 GMT
View Forum Message <> Reply to Message

Hi Thomas,

Can I ask, did you implement this updated algorithm in your development patch for the DW linux distribution?

I was comparing 2 libraries to each other (120k vs 500k) and on MacOS with development patch, the timing was sitting at ~30hrs whereas linux was sitting at > 170hrs (the linux machine is more hardware capable).

Was just wondering where the time discrepancy is coming from.

Many thanks

Subject: Re: Similarity analysis using "find similar compounds..." - slow analysis of libraries
Posted by thomas on Thu, 11 Feb 2021 09:55:58 GMT
View Forum Message <> Reply to Message

Sorry for the delay. Since the MacOS and the Linux version use exactly the same jar file, they both use the same procedure. However, it slipped my attention that your version didn't use multiple threads for the similarity calculation unless you used the flexophore descriptor. With that version I made a comparison on my Linux desktop and my MacBookPro (87.000 against 105.000 compounds), which took about 20 minutes (FragFp) and >2 hours (SkelSpheres) on both computers. I had expected the Linux machine to be faster, because it has a desktop i7 compared to the some years older MacBookPro laptop i7. I don't have an explanation for the Mac being equally fast.

I now updated the code (and dev version) to use all threads when calculating similarities independent of the descriptor type, which now increased the speed by factor 4 on my hexacore Linux. The speed gain is not higher, because not everything is multithreaded and because of the

overhead to launch threads for every molecule from the second file set. In general the needed processing time should grow more or less linearly with the number of compounds in file 1 and 2. 30 or even 170 hours seem very high to me. Which descriptor did you use? The molecule size does not play a role, because the calculation is running on descriptors only. If the second file doesn't contain the needed descriptors, they are calculated on the fly, which then adds to the time needed. I will make some more tests with larger files...

Subject: Re: Similarity analysis using "find similar compounds..." - slow analysis of libraries
Posted by SM2020 on Mon, 01 Mar 2021 00:00:30 GMT
View Forum Message <> Reply to Message

Hi Thomas - sorry for my delay.
I was using the SkelSpheres descriptor.
I will install the updated dev version and play around.

Many thanks again for your time.

Subject: Re: Similarity analysis using "find similar compounds..." - slow analysis of libraries
Posted by SM2020 on Tue, 02 Mar 2021 22:40:17 GMT
View Forum Message <> Reply to Message

Hi Thomas - FYI, 120K vs 234K was sitting at ~1hr using an 8 core/16 thread setup (skelspheres) using the newest dev version. A definite speed improvement.

Thanks again.
