Metabolomics (2012) 8:719–726
DOI 10.1007/s11306-011-0369-1

SHORT COMMUNICATION

MetAlign 3.0: performance enhancement by efficient use of advances in computer hardware

Arjen Lommen · Harrie J. Kools

Received: 19 August 2011 / Accepted: 25 September 2011 / Published online: 8 October 2011

Abstract A new, multi-threaded version of the GC–MS and LC–MS data processing software, metAlign, has been developed which is able to utilize multiple cores on one PC. This new version was tested using three different multi-core PCs with different operating systems. The performance of noise reduction, baseline correction and peak-picking was 8–19 fold faster compared to the previous version on a single core machine from 2008. The alignment was 5–10 fold faster. Factors influencing the performance enhancement are discussed. Our observations show that performance scales with the increase in processor core numbers we currently see in consumer PC hardware development.

Keywords metAlign · LC–MS · GC–MS · Multi-threading · Data processing · Metabolomics

Abbreviations

LC	Liquid Chromatography
GC	Gas Chromatography
MS	Mass Spectrometry
TOF	Time of Flight
PC	Personal computer
I/O	Input/output
HD	Hard drive
SSD	Solid-state drive
RAM	Random access memory
RAMD	Random access memory drive
WINE	Wine Is Not an Emulator
OS	Operating system

1 Introduction

MetAlign is a software program for the pre-processing and comparison of full scan nominal or accurate mass LC–MS and GC–MS data (Lommen 2009). This program has been successfully used in many metabolomics studies (America et al. 2006; Ballester et al. 2010; Beekwilder et al. 2008; Berendsen et al. 2009; de Bok et al. 2011; de Vos et al. 2007; Ducruix et al. 2008; Keurentjes et al. 2006; Kuzina et al. 2009, 2011; Lommen 2009; Lommen et al. 2007, 2011; Matsuda et al. 2009, 2011; Morant et al. 2010; Pino Del Carpio et al. 2011; Rijk et al. 2009; Ruiz-Aracama et al. 2011; Stracke et al. 2009; Tikunov et al. 2005, 2010; Yang et al. 2011; Tolstikov et al. 2003; Tsugawa et al. 2011; Vorst et al. 2005; Wegkamp et al. 2010). It was initially developed on a 32 bits operating system on a single core PC.

Since the publication of metAlign in 2009 the standard consumer desktop PC configuration has changed. PCs have changed from single core to multi-core processors. Large random access memory (RAM) is common and now also accessible, due to the appearance of mainstream 64-bit operating systems like for instance Windows 7. A 64 bit OS can address more than the effective maximum of 2.3 Gb RAM in a 32 bit OS. Concerning storage, a new generation of PCIe based solid-state drives are now available, which have extremely small access times, and a very high transfer rate compared to conventional hard drives.

The ability of multi-core processors to increase application performance depends on the use of multiple threads

Arjen Lommen and Harrie J. Kools contributed equally.

A. Lommen (✉) · H. J. Kools
RIKILT—Institute of Food Safety, Wageningen UR, P.O. Box 230, 6700 AE Wageningen, The Netherlands
e-mail: arjen.lommen@wur.nl

within applications. A thread of execution is the smallest unit of processing that can be scheduled by an OS. In principle many threads can run on one core, but although a thread can be moved from core to core it can not utilize more than one core simultaneously. Most current software has been developed for one core and can not exploit multiple cores. This is also the case for the previous version of metAlign, which consists of multiple separate 32 bits programs which are run in a batch through an interface using one thread on one core.

The current version of metAlign (available as a free download at www.metalign.nl) now involves running multiple threads simultaneously so that a one thread per core situation is established in which memory between threads is not shared. This allows the program to efficiently use all available processors and cores. MetAlign acts as if it is running on multiple single core PCs at the same time but within one hardware system. Since metAlign is still compiled as 32 bits executables, memory requirements never can exceed 2 Gb per thread. The new version of metAlign can run on 32 bit as well as 64 bit operating systems. For a 64 bit OS, which can address far more memory than a 32 bit OS, 2 Gb RAM per thread is an easily met requirement.

In this study the factors influencing the performance enhancement of the multi-threaded version of metAlign will be discussed for 3 different multi-core PCs with 3 different operating systems in relation to a single core PC from 2008. It will be shown that affordable PCs can give performance enhancements of an order of a magnitude using the latest version of metAlign. This will be useful in speeding up analysis of metabolomics data. The test data chosen were LC-TOF accurate mass data of resolution 8000, since this type of data is very common.

2 Materials and methods

2.1 Test data

The 16 accurate mass data sets used to test the performance of the new metAlign version were acquired on an UPLC system which was directly interfaced to a Bruker Daltonics microTOF mass spectrometer. The mass spectrometer was equipped with an orthogonal electrospray ionization source and was operated in positive ionization mode. Instrument calibration was performed externally prior to each sequence with a sodium formate/acetate solution consisting of 3.3 mM sodium hydroxide in a mixture of water/isopropanol/formic acid/acetic acid (1:1:1:3, v/v). Automated post-run internal mass scale calibration of the individual samples was performed by automated injection of the calibrant at the beginning of each run. Data were acquired between *m/z* 80–1,000 with a mass resolution of 8000. Data were exported as netCDF and were ca. 83 Mb a piece. Typically a region of 1,800 scans was used for processing with metAlign.

2.2 Software development and modification

The source code of the previous metAlign version was reused and recompiled in Visual C++ 2010. The current version accepts Masslynx format (if the correct version of Masslynx is installed), Thermo format (if the correct version of Xcalibur is installed), netCDF, mzData, mzXML, Agilent GCMS format and Agilent csv output. In the metAlign configuration interface the number of threads to use can now be defined to efficiently utilize all the available processor cores. The new version of metAlign is available as free download at www.metalign.nl. Instructions for installing and use are found in the documentation folder within the download.

2.3 PC platforms

Four PCs were used for testing and development of the software.

PC1: (year 2008: 1 core) This is the machine used in the metAlign reference (Lommen 2009) and it is used to establish the benchmark: operating system = Windows XP (32 bits), processor = Intel Pentium 4 530 3 GHz, memory = 1.5 Gb 200 MHz DDR, storage = SATA II 1 TB/7,200 rpm/32 Mb cache, no SSD.

PC4 (year 2011:4 cores) Operating system = Windows 7 (64 bits), processors = Single quad core Intel Core i5-2300 2.8 GHz, memory = 8 Gb 333 MHz DDR3, storage = SATA II 1 TB/7,200 rpm/32 Mb cache, 256 Gb OCZ Revodrive 2 PCI-Express (2× SSD).

PC8: (year 2009: 8 cores) Operating system = Windows XP (32 bits), processors = Dual quad core AMD opteron 2376 2.3 GHz, memory = 4 Gb (effective 2.3 Gb) 333 MHz DDR2, storage = SATA 640 Gb/7200 rpm/16 Mb cache, 256 Gb OCZ Z-Drive R2 m84 PCI-Express (4× SSD).

PC16: (year 2010: 16 cores) Operating system = Linux (64 bits) using WINE (http://www.winehq.org) (a Linux installation guide is given in the documentation folder of the download zip), processors=Dual octacore AMD opteron 6128 2.0 GHz, memory=64 Gb 333 MHz DDR3, storage=3× SATA II 1.5 TB/5400/32 Mb in Linux software raid5, 8 × 120 Gb OCZ Vertex 2 SATA II 2.5″ in raid0 (8× SSD).

Calculations were done on standard conventional hard drives as well as SSD's to establish the effect of storage media. Swap files were always set to the same storage medium as where the calculations were done unless mentioned otherwise. PC16 was also equipped with a variable size ramdisk to make complete runs in the memory possible.

2.4 MetAlign test runs

All test runs were done with identical settings optimized for these data (raw data in netCDF format and settings are available on request). The number of threads, the number of data files and the storage medium used (i.e. conventional hard drive, SSD, ramdisk) were varied. The time needed to process the test data files on PC1 with the previously published version of metAlign (Lommen 2009) was used as the benchmark for all other calculations.

3 Results and discussion

3.1 Difference in architecture

Compared to the old version of metAlign, the new version has a different architecture. In the old version conversion of data sets is integrated into the sequential processing of the data sets. In the new version, however, conversion and processing is separated. The program starts with data conversion of all data sets to netCDF, continues with processing of all netCDF files and then ends with a conversion back to a desired format. The new module "raw-data_conversion.exe" is executed by the metAlign interface and runs conversion in-line. This conversion tool, however, can also be used off-line as well; the allowed input formats (seven in total) now also include mzXML and mzData.

In essence "PART A" of the metAlign interface, which controls noise reduction, baseline correction, peak-picking and mass assignment has not changed. All previously validated code remains unmodified. In the new version the interface now distributes the processing of the data sets over the available cores by running independent batches (one thread per batch) in parallel.

In "PART B" of the metAlign interface, which controls the alignment of the pre-processed data sets, not all instructions are performed on multiple cores. For example, creating one file from all pre-processed data sets requires the use of one thread, since the resulting overall file is not shared in the memory. This in effect is now the speed bottleneck in the alignment. Considering the fact that a few years back the available memory was limiting, this part of the program used to require pointers in multiple opened files to be continuously moved. This in turn makes a conventional hard drive slow and therefore also the program. Since memory is no longer a limiting factor this part of the alignment has been changed to read all pre-processed data sets into memory sequentially; the effect is that the hard drive performance penalty is circumvented.

3.2 Effect of new code versus old code

Table 1 shows the effect of the new metAlign versus the old version for "PART A" running noise reduction, baseline correction, peak-picking and mass assignment. The same test calculations were done on one core on different PCs using different drives. PC1 was used as benchmark. For "PART A" the new version is on average 20% faster on one core. Since no basic changes were done on the algorithms of metAlign in this part of the program, it is concluded that this is purely due to the difference in compiler. The new version is compiled with Visual C++ 2010 while the older version was compiled using Visual C++ 6.0; the more recent compiler gives faster executables.

Using a conventional hard drive or SSD makes a difference for PC8, which has a 32 bits XP OS with limited access to RAM (2.3 Gb effective of which part is taken by the OS, the virus scanner etc.). Due to the limited RAM PC8 must make use of the swap file (paging file) and read and write more often. Therefore - for PC8 - the faster disk I/O of the SSD reduces the overall processing time. For PC4 and PC16 no difference is found for conventional hard drive versus solid state drive. Both systems (64 bits OS) have at least 2 Gb of RAM available per used thread and therefore far more addressable memory than PC8. These systems have enough memory to delay disk I/O until it can be done fast and efficient at convenient and optimal times (caching) during processing; therefore disk I/O only adds a few seconds. In effect no substantial time seems to be lost by disk I/O or swapping. This conclusion is further confirmed by the observation that running PART A totally in RAM (using a RAM drive in PC16) also shows no significant difference in processing time.

Table 2 ("PART B" = alignment) can be explained along the same lines as Table 1. In Sect. 3.1 it is noted that

Table 1 Average time in seconds needed per test data set by PART A using one core on different PCs

	Old metAlign version			New metAlign version		
	HD	SSD	RAM	HD	SSD	RAM
PC1	576	nd	nd	521	nd	nd
PC4	208	206	nd	170	167	nd
PC8	478	424	nd	393	340	nd
PC16	473	475	469	371	371	368

PC codes as described in Sect. 2

HD conventional hard drive; *SSD* solid state drive; *RAMD* drive in random access memory

Table 2 Average time in seconds needed for the alignment of the 16 test data sets by PART B using one core on different PCs

	Old metAlign version			New metAlign version		
	HD	SSD	RAM	HD	SSD	RAM
PC1	3645	nd	nd	1794	nd	nd
PC4	1592	1202	nd	493	467	nd
PC8	3338	3171	nd	1295	1146	nd
PC16	nd[a]	nd[a]	nd[a]	805	804	800

PC codes as described in Sect. 2

HD conventional hard drive; *SSD* solid state drive; *RAMD* drive in random access memory

[a] Alignment using the old metAlign does not work under Linux and WINE

part of the program is now performed by reading all data sets resulting from "PART A" into memory instead of going back and forth on the hard drive. For this example of 16 data sets this change together with a better compiler increases the speed by a factor of ca. 3. PC4 and PC16 again show the advantage of a 64 bit OS having the capability of addressing more memory and being able to cache efficiently and therefore minimize time loss due to disk I/O. For the older version of metAlign, in particular because of a higher need of disk I/O, a solid state drive helps to speed up processing.

Clearly PC4 as the most recently purchased PC is the fastest on one core. This is for the most part due to the faster processor, but also to the increased and faster memory and overall hardware configuration.

3.3 Evaluation of the effect of RAM, disk I/O and multi-threading on data reduction ("PART A")

In Fig. 1 the same settings were used to preprocess the 16 test data sets on the PCs. The type of storage drives and the number of threads were varied. Figure 1c shows that for PC16 - even with 16 data sets in parallel using 16 threads for 16 cores - the choice between a conventional hard drive, SSD or RAM drive is irrelevant. Furthermore the paging file (although present) isn't used for any of the calculations on this system. Disk I/O adds approximately 1% to the processing time. The dependence on the number of threads is skewed. This is more or less the case for all thread numbers in which the number of data sets (i.e. 16) divided by the number of threads is not a whole number. For example: 16 data sets using 12 threads will need a cycle of 12 followed by a cycle of 4 data sets, while 16 data sets using 16 threads will only depend on the slowest data

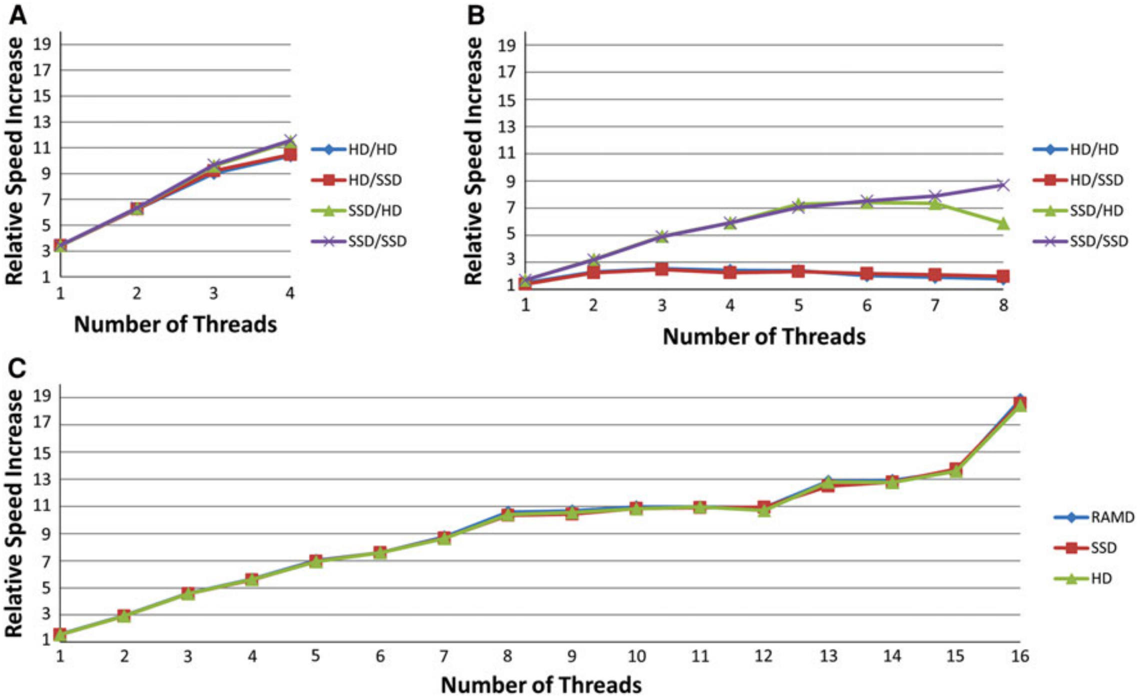

Fig. 1 Relative speed increase of the new metAlign as a function of the number of cores used (16 test data sets). The benchmark is the old version of metAlign running on PC1 (=9216 s for 16 test data sets). **a** PC4: *HD/HD* processing and paging file on conventional hard drive; *HD/SSD* processing on conventional hard drive and paging file on solid state drive; *SSD/HD* processing on solid state drive and paging file on conventional hard drive, *SSD/SSD* processing and paging file on solid state drive. **b** PC8: HD/HD, SSD/HD, HD/SSD and SSD/SSD as for (**a**). **c** PC16: paging file is never accessed; *HD* processing on conventional hard drive; *SSD* processing on solid state drive; *RAMD* processing on RAM drive

set. In practice the second cycle of 4 data sets can start as soon as threads from data sets needing the least processing time have finished. Therefore a leveling off occurs which is for example easily observed between 9 and 15 threads in Fig. 1c.

PC4 in Fig. 1a shows a minor effect on speed when using 3 or 4 cores and different combinations of storage drives. In general the performance increases with the number of threads used. The location of the paging file is less relevant, but running the program and data on the SSD gives a 13% increase in performance. This configuration has 2 Gb of RAM available per thread, which is the maximum a 32 bits module can address. However, the Windows 7 OS, virus scanner etc. also need some RAM and CPU resources. Therefore pushing this configuration to the limit with 4 threads might influence caching and therefore disk I/O and performance.

PC8 in Fig. 1b clearly shows the limits of a 32 bit OS. Although 4 Gb is installed, the effectively available RAM does not exceed 2.3 Gb. Since each thread may occasionally need up to 2 Gb of RAM during processing this is a severely limiting factor. Threads will be waiting for each other to free up RAM and the OS will start using the swap space on disk; this evidently decreases the performance. A lack of RAM will also mean that nearly no memory is available for disk caching, which will further increase the disk-I/O. In such a situation a conventional hard drive will not be able to cope with the I/O requirements and this directly results in a speed decrease as shown in traces HD/HD and HD/SSD in Fig. 1b. Exchanging the conventional hard drive with a SSD facilitates I/O greatly and therefore the performance; for the location of the paging file (swapping) this is only apparent when more than six threads are used.

3.4 Scalability of performance using multiple cores for data reduction ("PART A")

Figure 2 shows the relative speed increase compared to PC1 when increasing the number of data sets and keeping the maximum number of threads constant. All PCs show a dip at number of data sets = (maximum number of threads + 1). This is obvious since a second cycle of data sets has to be started. Because not all data sets have identical processing times asynchronicity in the data processing occurs when the number of data sets is much larger than the number of threads. This shows up as a constant relative speed increase in for instance the four thread maximum examples in Fig. 2a and c. Figure 2c clearly shows that the relative speed increase is linearly correlated to the number of available cores (with the condition of one thread per core). Therefore the new metAlign is shown to be able to scale with the number of cores for a 64 bit OS having at least 2 Gb RAM available per thread. Even PC8 with a 32 bits OS seems to scale up to about 8 threads—although not linearly—when using the PCIe SSD.

3.5 Effect of the number of cores on the speed of alignment ("PART B")

The same 16 data sets were aligned using the different PCs and different numbers of threads. As mentioned in Sect. 3.1 only part of the alignment procedure can benefit from multi-threading. A large part of the alignment is dependant on one thread. A rate determining step is now performed in memory circumventing a disk I/O bottleneck. Therefore CPU speed is a highly critical factor in the alignment.

In Fig. 3 it is shown that PC4 with the fastest CPU clearly outperforms the other PCs. The traces of PC16 show that a slight decrease in speed can be expected when using more than eight threads. In this particular alignment (>8 threads) the overhead time needed to perform multi-threading cancels out the theoretical speed increase. For PC4 (with only four cores) using four threads therefore is obviously still advantageous.

Although it seems here that more than eight threads should not be used, this will not hold for alignments with many more data sets. Alignment times increase more than linear with more data sets. Therefore the part of the alignment which is multi-threaded will have a smaller overhead time compared to the processing time when aligning far more than 16 data sets.

PC4 and PC16 have enough RAM to allow for efficient caching in the alignment process. The use of a ram drive or SSD does not influence the calculation time more than a few percent. PC8 is clearly influenced by disk I/O by a constant factor. This indicates that disk I/O does not vary with the number of threads, which is of course the case since the total data I/O is the same. Therefore most probably the limitation for PC8 is again in the amount of RAM. Dividing parts of the alignment into smaller pieces may make it possible to use the available RAM more efficiently for PC8 and therefore speed up alignment.

3.6 Accurate versus nominal mass processing

Although the above account is on accurate mass processing, it is also possible to process accurate mass data as nominal mass data using metAlign. For the purpose of comparing accurate mass processing with nominal mass processing the same data were also processed as nominal mass data.

Accurate mass processing requires more processing steps than nominal mass processing due to a second baseline correction, many more mass traces, accurate mass

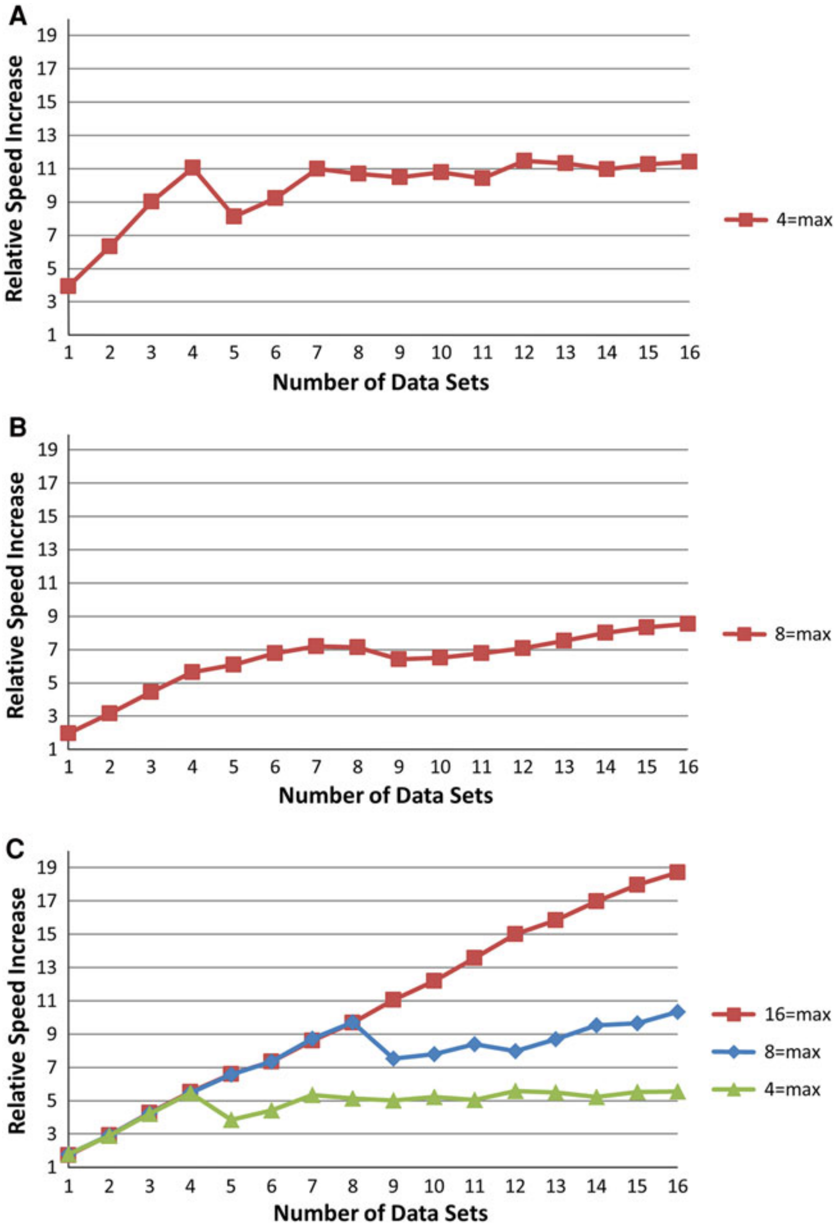

Fig. 2 Relative speed increase of the new metAlign as a function of the number of data sets. The benchmark is the old version of metAlign running on PC1 (576 s = average time per dataset). **a** PC4: calculations on SSD; using a maximum of four threads. **b** PC8: calculations on SSD; using a maximum of eight threads. **c** PC16: calculations on SSD, using a maximum of resp. 4, 8 and 16 threads

calculation over the peaks and filtering of mass artifacts. Furthermore, in the alignment—besides many more mass traces—a mass alignment must also take place. Using the old version of metAlign and PC1 (16 data sets) the difference between the average processing time per data set for "PART A" is 576 s (accurate mass) versus 27 s (nominal mass); For "PART B" this is 3,644 s versus 120 s.

Using the new metAlign and PC16 this is down to 4 s per data set in "PART A" and 73 s for "PART B". PC4 shows 3 s per data set in "PART A" and 49 s for "PART B". At these short processing times it becomes apparent that the overhead time needed for multi-threading becomes significant with regard to processing time. Especially when using high numbers of threads the relative speed increase is not linearly correlated to the number of available threads any more. On the other hand the processing time is so short that this becomes irrelevant.

4 Concluding remarks

The new version of metAlign is shown to increase the speed of processing considerably. The speed is now dependant on the available cores and CPU power. Having 2–4 Gb RAM per core available seems to make caching so efficient that disk I/O (even on a conventional hard drive) only takes up a few percent of the processing time. If the amount of RAM is limiting as is the case for a 32 bits OS,

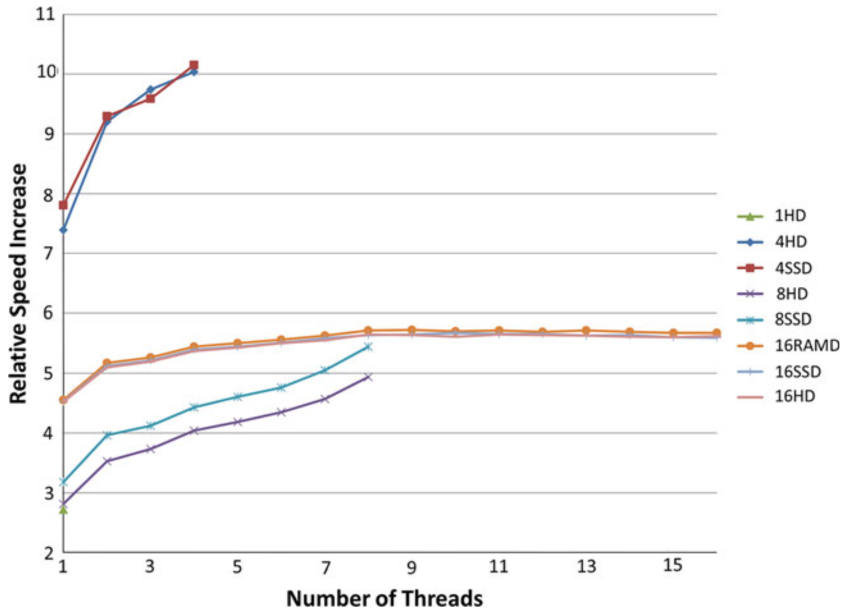

Fig. 3 Effect of the number of threads on the speed of alignment. The benchmark is the old version of metAlign running on PC1 (3645 s = per 16 data sets). 1, 4, 8, 16 indicate PC1, PC4, PC8 and PC16; *HD* conventional hard drive; *SSD* solid state drive; *RAMD* RAM drive

then it is necessary to have a fast SSD present to take care of the increased amount of inefficient disk I/O. In general, this new version is scalable with respect to the number of processor cores present assuming one thread per core. For "PART A", which performs noise reduction, baseline correction, peak-picking and mass assignment, data sets are sent to separate threads allowing full use of all cores. Alignment ("PART B") is by nature less scalable, but also benefits from having more processor cores.

Although CPU speed and the number of cores play a significant role in processing time, it should be noted that setting parameters correctly for baseline correction will also greatly influence performance. For instance in "PART A", not using mass artifact filters for accurate mass data (1B in the interface) or the absence of a threshold (parameter 8B) will greatly (perhaps 10 fold or more) increase the number of mass traces to correct; this is linearly related to processing time. As a direct consequence the processing time for alignment "PART B" is also similarly affected.

PC4 seems to be a good choice considering price and performance at the current time. This is now a standard desktop PC configuration for consumers. Using the new multi-threaded version of metAlign it already decreases processing time by a factor of 10 compared to a single core PC from 2008 using the previous metAlign version. In practice, 16 accurate mass test data sets previously required ca. 213 min for PART A and PART B together; this is now reduced to 20 min by introducing the new version of metAlign as well as a modern PC equipped with Core i5 technology, 8 Gb memory and a 64 bit OS. In addition, current mainstream processors include faster models like Core i7 processors with up to six cores, which are expected to increase speed another twofold compared to PC4 if the required 12 Gb of memory (2 Gb per processor core) is met.

Acknowledgments This work was supported by the Dutch Ministry of Agriculture, Nature and Food Quality, Strategic Research Funds RIKILT-WUR (project 1217269301) and The Netherlands Toxicogenomics Centre (NTC) (project 05060510)—Netherlands Genomics Initiative/Netherlands Organisation for Scientific Research (NWO).

References

America, A. H. P., Cordewener, J. H. G., Van Geffen, H. A., Lommen, A., Vissers, J. P. C., Bino, R. J., et al. (2006). Alignment and statistical difference analysis of complex peptide data sets generated by multidimensional LC-MS. *Proteomics, 6*, 641–653.

Ballester, A. R., Molthoff, J., de Vos, R. C. H., Hekkert, B. T. L., Orzaez, D., Fernandez-Moreno, J. P., et al. (2010). Biochemical and molecular analysis of pink tomatoes: Deregulated expression of the gene encoding transcription factor S1MYB12 leads to pink tomato fruit color. *Plant Physiology, 152*, 71–84.

Beekwilder, J., Van Leeuwen, W., Van Dam, N. M., Bertossi, M., Grandi, V., Mizzi, L., et al. (2008). The impact of the absence of aliphatic glucosinolates on insect herbivory in *Arabidopsis*. *PLoS One, 3*, e2068.

Berendsen, B. J., Essers, M. L., Mulder, P. P., van Bruchem, G. D., Lommen, A., van Overbeek, W. M., et al. (2009). Newly identified degradation products of ceftiofur and cephapirin impact the analytical approach for quantitative analysis of kidney. *J Chromatography A, 1216*, 8177–8186.

de Bok, F. A., Janssen. P. W., Bayjanov, J. R., Sieuwerts, S., Lommen, A., van Hylckama Vlieg, J. E., & Molenaar, D. (2011)

Volatile compound fingerprinting of mixed culture fermentations. *Applied and Environmental Microbiology* (Epub ahead of print).

de Vos, C. H. R., Moco, S., Lommen, A., Keurentjes, J. J. B., Bino, R. J., & Hall, R. D. (2007). Untargeted large-scale plant metabolomics using liquid chromatography coupled to mass spectrometry. *Nature Protocols, 2*, 778–791.

Ducruix, C., Vailhen, D., Werner, E., Fievet, J. B., Bourguignon, J., Tabet, J.-C., et al. (2008). Metabolomic investigation of the response of the model plant *Arabidopsis thaliana* to cadmium exposure: Evaluation of data pretreatment methods for further statistical analyses. *Chemometrics and Intelligent Laboratory Systems, 91*, 67–77.

Keurentjes, J. J. B., Jingyuan, F., de Vos, C. H. R., Lommen, A., Hall, R. D., Bino, R. J., et al. (2006). The genetics of plant metabolism. *Nature Genetics (Technical Report), 38*, 842–849.

Kuzina, V., Ekstrøm, C. T., Andersen, S. B., Nielsen, J. K., Olsen, C. E., & Bak, S. (2009). Identification of defense compounds in *Barbarea vulgaris* against the herbivore *Phyllotreta nemorum* by an ecometabolomic approach. *Plant Physiology, 151*, 1977–1990.

Kuzina, V., Nielsen, J. K., Augustin, J. M., Torpe, A. M., Bak, S., & Andersen, S. B. (2011). *Barbarea vulgaris* linkage map and quantitative trait loci for saponins, glucosinolates, hairiness and resistance to the herbivore *Phyllotreta nemorum*. *Phytochemistry, 72*, 188–198.

Lommen, A. (2009). MetAlign: Interface-driven, versatile metabolomics tool for hyphenated full-scan mass spectrometry data preprocessing. *Analytical Chemistry, 81*, 3079–3086.

Lommen, A., Gerssen, A., Oosterink, J. E., Kools, H. J., Ruiz-Aracama, A., Peters, R. J. B., et al. (2011). Ultra-fast searching assists in evaluating sub-ppm mass accuracy enhancement in U-HPLC/Orbitrap MS data. *Metabolomics, 7*, 15–24.

Lommen, A., van der Weg, G., van Engelen, M. C., Bor, G., Hoogenboom, L. A. P., & Nielen, M. W. F. (2007). An untargeted metabolomics approach to contaminant analysis: Pinpointing potential unknown compounds. *Analytica Chimica Acta, 584*, 43–49.

Matsuda, F., Nakabayashi, R., Sawada, Y., Suzuki, M., Hirai, M. Y., Kanaya, S., et al. (2011). Mass spectra-based framework for automated structural elucidation of metabolome data to explore phytochemical diversity. *Frontiers in Plant Science, 2*, 40. doi: 10.3389/fpls.2011.00040.

Matsuda, F., Yonekura-Sakakibara, K., Niida, R., Kuromori, T., Shinozaki, K., & Saito, K. (2009). MS/MS spectral tag-based annotation of non-targeted profile of plant secondary metabolites. *The Plant Journal, 57*, 555–577.

Morant, M., Ekstrøm, C., Ulvskov, P., Kristensen, C., Rudemo, M., Olsen, C. E., et al. (2010). Metabolomic, transcriptional, hormonal, and signaling cross-talk in Superroot2. *Molecular Plant, 3*, 192–211.

Pino Del Carpio, D., Kumar Basnet, R., de Vos, C. H. R., Maliepaard, C., Visser, R., & Bonnema, G. (2011). The patterns of population differentiation in a *Brassica rapa* core collection. *Theoretical and Applied Genetics, 122*, 1105–1118.

Rijk, J. C. W., Lommen, A., Essers, M. L., Groot, M. J., van Hende, J. M., Doeswijk, T. G., et al. (2009). Metabolomics approach to anabolic steroid urine profiling of bovines treated with prohormones. *Analytical Chemistry, 81*, 6879–6888.

Ruiz-Aracama, A., Peijnenburg, A., Kleinjans, J., Jennen, D., van Delft, J., Hellfritsch, C., et al. (2011). An untargeted multi-technique metabolomics approach to studying intracellular metabolites of HepG2 cells exposed to 2,3,7,8-tetrachlorodibenzop-dioxin. *BMC Genomics, 12*, 251–270.

Stracke, R., de Vos, C. H. R., Bartelniewoehner, L., Ishihara, H., Sagasser, M., Martens, S., et al. (2009). Metabolomic and genetic analyses of flavonol synthesis in *Arabidopsis thaliana* support the in vivo involvement of leucoanthocyanidin dioxygenase. *Planta, 229*, 427–445.

Tikunov, Y. M., de Vos, C. H. R., Paramas, A. M. G., Hall, R. D., & Bovy, A. G. (2010). A role for differential glycoconjugation in the emission of phenylpropanoid volatiles from tomato fruit discovered using a metabolic data fusion approach. *Plant Physiology, 152*, 55–70.

Tikunov, Y. M., Lommen, A., de Vos, C. H. R., Verhoeven, H. A., Bino, R. J., Hall, R. D., et al. (2005). A novel approach for nontargeted data analysis for metabolomics. Large-scale profiling of tomato fruit volatiles. *Plant Physiology (Break Through Technologies Section), 139*, 1125–1137.

Tolstikov, V. V., Lommen, A., Nakanishi, K., Tanaka, N., & Fiehn, O. (2003). Monolithic silica-based capillary reversed-phase liquid chromatography/electrospray mass spectrometry for plant metabolomics. *Analytical Chemistry, 75*, 6737–6740.

Tsugawa, H., Tsujimoto, Y., Arita, M., Bamba, T., & Fukusaki, E. (2011). GC/MS based metabolomics: Development of a data mining system for metabolite identification by using soft independent modeling of class analogy (SIMCA). *BMC Bioinformatics, 12*, 131–143.

Vorst, O., de Vos, C. H. R., Lommen, A., Staps, R. V., Visser, R. G. F., Bino, R. J., et al. (2005). A non-directed approach to the differential analysis of multiple LC MS-derived metabolic profiles. *Metabolomics, 1*, 169–180.

Wegkamp, A., Mars, A. E., Faijes, M., Molenaar, D., de Vos, C. H. R., Klaus, S. M. J., et al. (2010). Physiological responses to folate overproduction in *Lactobacillus plantarum* WCFS1. *Microbial Cell Factories, 9*, 100–113.

Yang, T., Stoopen, G., Yalpani, N., Vervoort, J., de Vos, C. H. R., Voster, A., et al. (2011). Metabolic engineering of geranic acid in maize to achieve fungal resistance is compromised by novel glycosylation patterns. *Metabolic Engineering, 13*, 414–425.