J Math Chem (2012) 50:2446–2470
DOI 10.1007/s10910-012-0042-6

ORIGINAL PAPER

Counting chemical compositions using Ehrhart quasi-polynomials

Shane L. Hubler · Gheorghe Craciun

Received: 29 March 2012 / Accepted: 1 June 2012 / Published online: 13 July 2012

Abstract To count the number of chemical compositions of a particular mass, we consider an alphabet $\mathcal{A}$ with a mass function which assigns a mass to each letter in $\mathcal{A}$. We then compute the mass of a word (an ordered sequence of letters) by adding the masses of the constituent letters. Our main interest is to count the number of words that have a particular mass, where we *ignore the order* of the letters within the word. We show first that counting the number of words of a given mass has a geometric interpretation, whose solutions are called Ehrhart quasi-polynomials, a class of functions defined on integers. These special functions are "periodic" in the sense that they use the same polynomial every λ steps. In addition to discovering the connection between counting compositions and Ehrhart quasi-polynomials, we also find number theoretic results that greatly reduce the number of candidates for the period, λ. Finally, we illustrate the usefulness of these results and the use of a software library named `barvinok` (by Verdoolaege et al.) by applying them to eight different classes of chemical compositions, including organic molecules, peptides, DNA, and RNA.

Keywords Chemical compositions · Peptide masses · Ehrhart quasi-polynomials · `Barvinok` software library

S. L. Hubler (✉)
Department of Mathematics and Department of Chemistry,
University of Wisconsin, Madison, WI 53706, USA
e-mail: slhubler@wisc.edu

G. Craciun (✉)
Department of Mathematics and Department of Biomolecular Chemistry,
University of Wisconsin, Madison, WI 53706, USA
e-mail: craciun@math.wisc.edu

1 Introduction

In this paper we consider an alphabet $\mathcal{A}$ with a mass function which assigns a mass to each letter in $\mathcal{A}$. We compute the mass of a word (an ordered sequence of letters) by adding the masses of the constituent letters. Our main interest is to count the number of words that have a particular mass, where we *ignore the order* of the letters within the word; "ABC" is considered the same as "CBA" but not the same as "ABCC". For results on *ordered* sequences, see [1–3]. We refer to the class of all words with the same letter composition as simply a "composition". Thus, we are interested in knowing how many compositions have a particular mass.

We show that the problem of counting compositions has a geometric interpretation involving counting the number of lattice points contained in a d-dimensional simplex, where d is the number of letters in the alphabet $\mathcal{A}$. If the masses are rational numbers we are able to use results by Eugene Ehrhart that say that the function that counts the number of lattice points found within a polytope $\mathcal{P}$ dilated by the integer factor t, $\mathcal{L}_{\mathcal{P}}(t) = \#t\mathcal{P} \cap \mathbb{Z}^d$, is a member of a class of functions now called Ehrhart quasi-polynomials.

A function $Q(t)$ is an Ehrhart quasi-polynomial if there exists a positive integer λ and polynomials $p_0, \ldots, p_{\lambda-1}$ such that

$$Q(t) = \begin{cases} p_0(t) & \text{if } t \equiv 0 \bmod \lambda \\ p_1(t) & \text{if } t \equiv 1 \bmod \lambda \\ & \vdots \\ p_{\lambda-1}(t) & \text{if } t \equiv \lambda - 1 \bmod \lambda. \end{cases}$$

These special functions are defined on integers and are "periodic" in the sense that they use the same polynomial every λ steps.

One way to characterize a quasi-polynomial of degree d and period λ is to list all $(d + 1) \times \lambda$ coefficients of the underlying polynomials. In this case it is useful to know the size of λ if we want to store the complete characterization of a particular quasi-polynomial; if λ is too large then it can be impossible or impractical to store or use the explicit solution. In Sect. 2 (Theorem 11) we determine lower and upper bounds on the period of the quasi-polynomial associated with counting compositions.

While we show that the number of compositions for *rational* masses is an Ehrhart quasi-polynomial, we also show in Sect. 3 that the current methodologies for calculating Ehrhart quasi-polynomials are extremely impractical for counting amino acid compositions of peptides, even when applied to *integer* masses. Furthermore, we show that the quasi-polynomial that counts amino acid compositions is so large that it would be largely useless, projecting that it would require 300 TB and 14 billion calculations every time it was applied. We should point out, however, that an improvement in the storage of the quasi-polynomial could make a difference in its usefulness. The software library which we use, `barvinok` [4], had already improved the representation of the quasi-polynomial through the use of remainder functions, decreasing required space and time by several orders of magnitude.

In addition we show empirically that improving accuracy appears to increase complexity in a roughly linear fashion. For example, if we wanted to improve our accuracy to 0.01 Daltons from 1 Dalton, we would increase the time and space requirements 100-fold.

Finally, we note that computing the general form of the solution involving irrational masses is a very interesting open problem.

2 Mathematical results

We show that the problem of counting the number of compositions of a given mass belongs to the class of problems that count lattice points inside a multi-dimensional solid and, as such, has a solution called an Ehrhart quasi-polynomial. In addition, we significantly restrict the number of possible periods of the Ehrhart quasi-polynomials (defined later) that count chemical compositions.

2.1 Chemical compositions and Ehrhart quasi-polynomials

Before we define the mathematical problem of counting compositions, we need to define a composition and its constituent parts.

First, we define an *alphabet* $\boldsymbol{a}$ as a finite ordered list of distinct objects $\{a_1, \ldots, a_d\}$.

A *composition* x from alphabet $\boldsymbol{a} = \{a_1, \ldots, a_d\}$ is an unordered multi-set consisting of zero or more copies of each element in $\boldsymbol{a}$. More intuitively, we think of a composition as a collection of letters where we alphabetize the letters (to remove the order information from the original sequence). Or, even simpler, we can call a composition a vector in $\mathbb{N}^d$. We denote the infinite list of compositions of $\boldsymbol{a}$ by Comp($\boldsymbol{a}$).

Next we define a *mass function*, mass : $\boldsymbol{a} \rightarrow \mathbb{R}^+$. In addition, we extend the mass function for $\mathbf{x} \in \text{Comp}(\boldsymbol{a})$, $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, by defining $\text{mass}(\mathbf{x}) = \sum_{i=1}^{d} x_i \text{mass}(a_i)$. We will use the notation $m_i = \text{mass}(a_i)$.

We define the *weighted alphabet* $\widetilde{\boldsymbol{a}}$ as the ordered pair ($\boldsymbol{a}$, mass()).

We denote the list of compositions of mass M or less, composed of elements of $\boldsymbol{a}$, as Comp $\left(\widetilde{\boldsymbol{a}}, M\right)$.

The size of set Comp $\left(\widetilde{\boldsymbol{a}}, M\right)$ is denoted $C(M)$. The function C is called the *cumulative composition counting function* for the weighted alphabet $\widetilde{\boldsymbol{a}}$. In other words,

$$\begin{aligned} C(M) &= \#\left\{\mathbf{x} \in \text{Comp}(\boldsymbol{a}) | mass(\mathbf{x}) \leq M\right\} \\ &= \#\text{Comp}\left(\widetilde{\boldsymbol{a}}, M\right). \end{aligned}$$

The general question we are trying to answer is the following *Composition Counting Problem: Suppose C is the cumulative composition counting function for* $\widetilde{\boldsymbol{a}}$. *How can we compute the function C in terms of mass M?*

Now that we have defined the algebraic aspects of our problem, we need to define its geometric aspects. In this paper the term *lattice points* refers to the points in a multi-dimensional space $\mathbb{R}^n$ that have integer coordinates ($\mathbb{Z}^n$). Of particular interest to us is the number of lattice points contained in a polytope $\mathcal{P}$ when it is dilated by a positive integer t, i.e. $\#\left(t\mathcal{P} \cap \mathbb{Z}^d\right)$. This number is denoted $\mathcal{L}_{\mathcal{P}}(t)$ [5].

We say that s is a mass simplex of dimension k if s is a k-dimensional simplex in $\mathbb{R}^n$ that includes the origin, such that every vertex of s is either the origin or on a positive axis; i.e. other than the origin, each vertex has exactly one non-zero coordinate and that coordinate is positive. Moreover, we assume that the vertices of s are affinely independent. Further, we define $s \subset \mathbb{R}^d$ as a mass simplex for $\widetilde{\mathcal{A}}$ (or *mass simplex for masses* $m_1, \ldots, m_d$) if each vertex is either the origin or $\frac{1}{m_i}$ on coordinate i and zero otherwise, for $1 \leq i \leq d$.

Note that the last two definitions are consistent; i.e. a mass simplex for masses $m_1, \ldots, m_d$ is a mass simplex. Also, note that a mass simplex for masses $m_1, \ldots, m_d$ can be expressed as the intersection of $d+1$ half-spaces: $\{\mathbf{x} | \boldsymbol{\mu} \cdot \mathbf{x} \leq 1\} \cap \bigcap_{i=1}^{d} \{\mathbf{x} | x_i \geq 0\}$, where $\boldsymbol{\mu} = (m_1, \ldots, m_d)$. From now on we use d to represent the total number of elements in our alphabet $\mathcal{A} = \{a_1, a_2, \ldots, a_d\}$.

These definitions allow us to state the following relationship between counting compositions and counting lattice points:

Lemma 1 (C counts the number of lattice points in the mass simplex) *Suppose C is the cumulative composition counting function over $\widetilde{\mathcal{A}}$ and $\mathcal{P}$ is the mass simplex for $\widetilde{\mathcal{A}}$. Then $C(t) = \mathcal{L}_{\mathcal{P}}(t)$ for all positive real numbers t.*

Proof Let d be the number of elements in $\mathcal{A}$. Fix a positive real number t and let $\mathbf{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{Z}^d$ represent a particular composition in Comp($\mathcal{A}$). We claim that mass($\mathbf{x}$) $\leq t$ if and only if $\mathbf{x} \in t\mathcal{P}$.

Note that mass($\mathbf{x}$) $\leq t$ iff $\sum_{i=1}^{d} m_i x_i \leq t$. Similarly, $\sum_{i=1}^{d} m_i x_i \leq t$ iff $\sum_{i=1}^{d} m_i \frac{x_i}{t} \leq 1$. This last inequality is equivalent to $\frac{\mathbf{x}}{t} \in \mathcal{P}$; i.e. $\mathbf{x} \in t\mathcal{P}$.

Thus, since $\mathbf{x} \in \mathbb{Z}^d$, $C(t) = \#(t\mathcal{P} \cap \mathbb{Z}^d) = \mathcal{L}_{\mathcal{P}}(t)$. □

The problem of counting lattice points in a *rational* polytope (a polytope with rational vertices) was approached by Eugene Ehrhart in [6]. His work gave rise to a class of functions later called Ehrhart quasi-polynomials: A function $Q : \mathbb{Z} \to \mathbb{R}$ is an Ehrhart quasi-polynomial if there exists a positive integer k and polynomials $p_0, \ldots, p_{k-1}$ such that

$$Q(t) = \begin{cases} p_0(t) & \text{if } t \equiv 0 \bmod \lambda \\ p_1(t) & \text{if } t \equiv 1 \bmod \lambda \\ & \vdots \\ p_{\lambda-1}(t) & \text{if } t \equiv \lambda - 1 \bmod \lambda. \end{cases}$$

The degree of Q is defined as $\max\limits_{0 \leq j \leq \lambda-1} \deg\left(p_j\right)$. Note that this is equivalent to saying that there exists an integer d and real numbers $a_{0,0}, \ldots, a_{0,d}, a_{1,0}, \ldots, a_{1,d}, \ldots, a_{\lambda-1,0}, \ldots, a_{\lambda-1,d}$ such that

Explicit form of Ehrhart quasi-polynomial

$$Q(t) = \begin{cases} a_{0,d}t^d + \cdots + a_{0,0}t^0 & \text{if } t \equiv 0 \bmod \lambda \\ a_{1,d}t^d + \cdots + a_{1,0}t^0 & \text{if } t \equiv 1 \bmod \lambda \\ & \vdots \\ a_{\lambda-1,d}t^d + \cdots + a_{\lambda-1,0}t^0 & \text{if } t \equiv \lambda - 1 \bmod \lambda. \end{cases} \tag{1}$$

Therefore, we will often write an Ehrhart polynomial as

$$Q(t) = \sum_{i=0}^{d} a_i(t)t^i$$

where $a_i(t)$ is a function on t which has period λ.

A key result in the study of counting lattice points within polytopes is:

Theorem 2 (Ehrhart's theorem for rational polytopes) *Suppose $\mathscr{P}$ is a rational convex d-dimensional polytope. Then $\mathscr{L}_{\mathscr{P}}(t)$ is an Ehrhart quasi-polynomial in integer variable t of degree d. Its period λ divides the least common multiple of the denominators of the coordinates of the vertices of $\mathscr{P}$.*

A proof of this theorem appears in [7].

By combining Lemma 1 (C counts the number of lattice points in the mass simplex) and Theorem 2 (Ehrhart's theorem for rational polytopes), we can describe the nature of solutions to the Composition Counting Problem.

Corollary 3 (C is an Ehrhart quasi-polynomial) *Suppose C is the cumulative composition counting function over $\widetilde{\mathcal{A}}$ and that $\widetilde{\mathcal{A}}$ is rational (i.e. all masses are rational numbers). Then C is an Ehrhart quasi-polynomial.*

Proof Let $\mathscr{P}$ be the mass simplex for $\widetilde{\mathcal{A}}$. Then by Lemma 1 (C counts the number of lattice points in the mass simplex), $C(t) = \mathscr{L}_{\mathscr{P}}(t)$. However, by Theorem 2 (Ehrhart's theorem for rational polytopes), $\mathscr{L}_{\mathscr{P}}(t)$ is an Ehrhart quasi-polynomial. Therefore, so is $C(t)$. □

Note that, because Ehrhart's theorem addresses only rational polytopes, we must make the same requirement in the Corollary.

2.2 Calculating periods of Ehrhart quasi-polynomials

While Ehrhart's theorem gives us the general form of the solution, we still want to derive the quasi-polynomial for a specific collection of masses. This can be done by calculating the number of compositions $C(M)$ through other means for a finite number of values M and then calculating the coefficients in Eq. (1) (Explicit form of Ehrhart quasi-polynomial) through standard interpolation techniques. However, in order to use that form of the quasi-polynomial, we must first calculate the period λ of the quasi-polynomial.

Several papers address the question of the period of a quasi-polynomial [8–10]. Here, we derive a tighter lower bound for the number of compositions $C(M)$ as well as provide a number-theoretic restriction based on prime-factoring. In order to obtain these results we must develop more mathematical machinery.

Given a function f of one variable, we say that $b(\cdot,\cdot)$ is the $\lambda - M$ difference triangle (of order d) for f if b satisfies

$$b(i,j) = \begin{cases} f(M+\lambda j) & \text{for } i=0 \quad \text{and} \quad j=0,\ldots,d \\ b(i-1,j+1) - b(i-1,j) & \text{for } i=1,\ldots,d \quad \text{and} \quad j=0,\ldots,d-i. \end{cases}$$

The name refers to the fact that one can form a triangle of differences; the first row consists of f evaluated at specific intervals, the second row consists of adjacent differences of the first row, the third row consists of adjacent differences of the second row, and so on.

Note that if the range of f is the integers then the difference triangle consists of integers.

The following lemma will be used to restrict the list of possible periods of a quasi-polynomial.

Lemma 4 (Last term of 1–0 difference triangle) *Consider the polynomial* $f(x) = a_0x^0 + \cdots + a_dx^d$ *and let* $b(\cdot,\cdot)$ *be the 1–0 difference triangle for* f*. Then* $b(d,0) = d!a_d$.

Proof We claim that $b(i,x) = \frac{d!}{(d-i)!}a_dx^{d-i} + g_i(x)$ where $g_i(x)$ is a polynomial of degree $d-i-1$ or less. We will prove this by induction on i.

For $i = 0$ we have

$$\begin{aligned} b(0,x) &= f(x) \\ &= a_dx^d + (a_{d-1}x^{d-1} + \cdots + a_0x^0). \end{aligned}$$

If we define g_0 to be the sum of all the terms of f of degree less than d then we have

$$\begin{aligned} b(0,x) &= a_dx^d + g_0(x) \\ &= \frac{d!}{(d-i)!}a_dx^d + g_i(x). \end{aligned}$$

Assume now that the induction hypothesis holds for i. We want to show that it holds for $i+1$:

$$\begin{aligned} b(i+1,x) &= b(i,x+1) - b(i,x) \\ &= \frac{d!}{(d-i)!}a_d\,(x+1)^{d-i} + g_i(x+1) - \frac{d!}{(d-i)!}a_d\,(x)^{d-i} - g_i(x) \\ &= \frac{d!}{(d-i)!}a_d\left((x+1)^{d-i} - (x)^{d-i}\right) + g_i(x+1) - g_i(x) \end{aligned}$$

$$
\begin{aligned}
&= \frac{d!}{(d-i)!} a_d \left((d-i)x^{d-i-1} + \binom{d-i}{2} x^{d-i-2} + \cdots + \binom{d-i}{d-i} x^0 \right) \\
&\quad + g_i(x+1) - g_i(x) \\
&= \frac{d!}{(d-i-1)!} a_d x^{d-i-1} + \frac{d!}{(d-i)!} a_d \\
&\quad \times \left(\binom{d-i}{2} x^{d-i-2} + \cdots + \binom{d-i}{d-i} x^0 \right) \\
&\quad + g_i(x+1) - g_i(x).
\end{aligned}
$$

Note that everything after the first term is a polynomial of degree $d-i-2$; we replace it with $g_{i+1}(x)$ to get

$$
\begin{aligned}
b(i+1,x) &= \frac{d!}{(d-i-1)!} a_d x^{d-i-1} + g_{i+1}(x) \\
&= \frac{d!}{(d-(i+1))!} a_d x^{d-(i+1)} + g_{i+1}(x),
\end{aligned}
$$

which is the desired result for $i+1$.

Thus, we have shown that $b(i,x) = \frac{d!}{(d-i)!} a_d x^{d-i} + g_i(x)$. Adding the fact that $g_d = 0$, we can conclude for $i = d$ and $x = 0$, that $b(d,0) = d!a_d$. □

Extending the previous lemma, we now calculate the last term in the more general case of a λ-M difference triangle.

Lemma 5 (Last term of λ-M difference triangle) *Consider a polynomial $f(x) = a_0x^0 + \cdots + a_dx^d$. Suppose λ and M are integers, and that $b(\cdot,\cdot)$ is the $\lambda - M$ difference triangle for f. Then $b(d,0) = d!\lambda^d a_d$.*

Proof Define $g(x) = f(M+\lambda x)$ with $g(x) = a'_0x^0 + \cdots + a'_dx^d$. Note that $b(\cdot,\cdot)$ is the 0-1 difference triangle for g. Then, by Lemma 4 (Last term of 1-0 difference triangle), $b(d,0) = d!a'_d$. We can determine a'_d by comparing the highest order terms of f and g:

$$
\begin{aligned}
a'_dx^d + \cdots + a'_0 &= g(x) \\
&= f(M+\lambda x) \\
&= a_d\,(M+\lambda x)^d + \cdots + a_0 \\
&= a_d\lambda^d x^d + \cdots .
\end{aligned}
$$

In other words, $a'_d = a_d\lambda^d$.

Thus,

$$
\begin{aligned}
b(d,0) &= d!a'_d \\
&= d!\lambda^d a_d.
\end{aligned}
$$

□

On a separate line of inquiry, we examine what is already known about the highest order term of $\mathcal{L}_{\mathcal{P}}(t)$.

Lemma 6 (Relationship between volume of polytope and the Ehrhart quasi-polynomial) *Suppose $\mathcal{P}$ is a rational convex d-dimensional polytope. Assume $\mathcal{L}_{\mathcal{P}}(t)$ is an Ehrhart-quasi-polynomial given by $\mathcal{L}_{\mathcal{P}}(t) = \sum_{j=0}^{d} a_j(t)t^j$. Then $a_d(t) = \mathrm{Vol}(\mathcal{P})$.*

Proof (A similar version of this result is described in [7], page 72). We consider computing the volume of $\mathcal{P}$ by counting the number of d-cubes that fit inside of it, as in the computation of a Riemann integral. Each d-cube fills space between the lattice points $\left(\frac{1}{t}\mathbb{Z}\right)^d$, each lattice point being contained in $\mathcal{P}$. Thus,

$$\begin{aligned}\mathrm{Vol}(\mathcal{P}) &= \int\limits_{\mathcal{P}} dx \\ &= \lim_{t\to\infty} \frac{1}{t^d}\#\left(\mathcal{P} \cap \left(\frac{1}{t}\mathbb{Z}\right)^d\right) \\ &= \lim_{t\to\infty} \frac{1}{t^d}\#\left(t\mathcal{P} \cap \mathbb{Z}^d\right) \\ &= \lim_{t\to\infty} \frac{1}{t^d}\sum_{j=0}^{d} a_j(t)t^j \\ &= \lim_{t\to\infty} \frac{a_d(t)t^d}{t^d}.\end{aligned}$$

Since $a_d(t)$ is a periodic function and the volume is a fixed number, $a_d(t)$ must be a constant. Therefore, $a_d(t) = \mathrm{Vol}(\mathcal{P})$. □

In order to use the previous lemma we need to know the volume of a mass simplex.

Lemma 7 (Volume of mass simplex) *Let $\mathcal{P}$ be a mass simplex for positive, real-valued masses $m_1, \ldots, m_d$. Then $Vol(\mathcal{P}) = \frac{1}{d!\prod_{i=1}^{d} m_i}$.*

Proof For $d = 1$, the volume of $\mathcal{P}$ is given by

$$Vol(\mathcal{P}) = \frac{1}{m_1}$$

Now suppose that the induction hypothesis holds for $d = k$. Note that $\mathcal{P}$ is a $k+1$-dimensional cone. Therefore, its volume is given by

$$\begin{aligned}Vol(\mathcal{P}) &= \frac{1}{k+1}(base)(height) \\ &= \frac{1}{k+1}\left(\frac{1}{k!\prod\limits_{i=1}^{k} m_i}\right)\left(\frac{1}{m_{k+1}}\right)\end{aligned}$$

$$= \frac{1}{(k+1)! \prod_{i=1}^{k+1} m_i}$$
$$= \frac{1}{d! \prod_{i=1}^{d} m_i}.$$

□

Combining the results of the previous two lemmas we have

Lemma 8 (Highest order term of C) *Suppose C is the cumulative composition counting function over $\widetilde{\mathfrak{A}}$ and that $\widetilde{\mathfrak{A}}$ is rational. Let d be the number of objects in the alphabet $\mathfrak{A}$. Then C is an Ehrhart quasi-polynomial of degree d, $\sum_{j=0}^{d} a_j(t)t^j$, with $a_d(t) = \frac{1}{d! \prod m_j}$.*

Proof Let $\mathscr{P}$ be the mass simplex for $\widetilde{\mathfrak{A}}$. Then by Lemma 1 (C counts the number of lattice points in the mass simplex), we know that $C(t) = \mathcal{L}_{\mathscr{P}}(t)$. Combining this fact with Lemma 6 (Relationship between volume of polytope and the Ehrhart quasi-polynomial) and Lemma 7 (Volume of mass simplex) we have

$$a_d(t) = Vol(\mathscr{P})$$
$$= \frac{1}{d! \prod_{i=1}^{d} m_i}$$

□

Applying Ehrhart's theorem in a different way to the case of mass simplexes, we are able to place an upper limit on the period, as well as put upper limits on the period's prime factorization.

Lemma 9 (λ divides least common multiple of masses) *Suppose C is the cumulative composition counting function over $\widetilde{\mathfrak{A}}$ and that $\widetilde{\mathfrak{A}}$ is rational. Assume that C is an Ehrhart quasi-polynomial of degree d and period λ and that the masses of $\widetilde{\mathfrak{A}}$ are $m_1, \ldots, m_d$. Then λ divides $\underset{i}{\operatorname{lcm}}(\operatorname{num}(m_i))$.*[1]

Proof Let $\mathscr{P}$ be the mass simplex for $\widetilde{\mathfrak{A}}$. Then the coordinates of the vertices of $\mathscr{P}$ are either zero or $\frac{1}{m_j}$ for $j = 1, \ldots, d$. Since $m_1, m_2, \ldots, m_d$ are positive rational numbers, so are $\frac{1}{m_j}$. By Theorem 2 (Ehrhart's theorem for rational polytopes), $\mathcal{L}_{\mathscr{P}}(t)$ is an Ehrhart quasi-polynomial in t of degree d, whose period λ divides the least common multiple of the denominators of the coordinates of the vertices of $\mathscr{P}$. However, the denominators of $\mathscr{P}$ are the numerators of the masses. Therefore, λ divides $\underset{i}{\operatorname{lcm}}(\operatorname{num}(m_i))$. □

For the following two theorems we denote the number of times that p divides m (i.e., $\max_{n \in \mathbb{N}} \{n : p^n | m\}$) by $n(p, m)$

[1] $\underset{i}{\operatorname{lcm}}(a_i)$ is the least common multiple of the numbers $\{a_i\}$.

We also represent the numerator of the rational number q (where q is in reduced form) as $\mathrm{num}(q)$.

The following theorem greatly restricts the number of possible periods that we might need to check in practice. In particular, it puts bounds on the number of times a prime may be found in the factorization of the period λ.

Theorem 10 (Bounds on divisors of λ) *Suppose C is the cumulative composition counting function over $\widetilde{\boldsymbol{\alpha}}$. Suppose also that C is an Ehrhart quasi-polynomial of period λ. Let d be the number of elements in $\boldsymbol{\alpha}$ and let p be any prime number. Then*

$$\left\lceil \frac{n\left(p, \mathrm{num}\left(\prod m_j\right)\right)}{d} \right\rceil \leq n(p, \lambda) \leq \max_j \left\{n\left(p, \mathrm{num}\left(m_j\right)\right)\right\}.$$

Proof We will handle the two inequalities separately.

Lower bound. Since C is an Ehrhart quasi-polynomial, write $C(t) = \sum_{i=0}^{d} a_i(t)t^i$. By the definition of λ we know that $\sum_{j=0}^{d} a_j(k + \lambda x)t^j = \sum_{j=0}^{d} a_j(k)t^j$ for integers k and x. This allows us to decompose C into λ polynomials:

$$C(t) = \begin{cases} f_0(t) & t \equiv 0 \bmod \lambda \\ & \vdots \\ f_{\lambda-1}(t) & t \equiv \lambda - 1 \bmod \lambda \end{cases}$$

where $f_k(t) = \sum_{j=0}^{d} a_j(k)t^j$ for $k = 0, \ldots, \lambda - 1$. For each $k = 0, \ldots, \lambda - 1$ let $b_k(\cdot, \cdot)$ be the $\lambda - k$ difference triangle of f_k. Then, by Lemma 5 (Last term of $\lambda - M$ difference triangle), $b_k(d, 0) = d!\lambda^d a_d$. (Note that the right-side is independent of k). Furthermore, by Lemma 8 (Highest order term of C), $a_d = \frac{1}{d! \prod m_j}$. Thus,

$$\begin{aligned} b_k(d, 0) &= d!\lambda^d a_d \\ &= \frac{\lambda^d}{\prod m_j} \end{aligned}$$

Note that $b_k(d, 0)$ is an integer. So, if p divides $\mathrm{num}\left(\prod m_j\right)$ a total of $n\left(p, \mathrm{num}\left(\prod m_j\right)\right)$ times, it must divide λ^d at least as many times. Thus, $\frac{n(p,\, \mathrm{num}\,(\prod m_j))}{d} \leq n(p, \lambda)$. Since the right-hand side is an integer, we may round the left-hand side up, yielding: $\left\lceil \frac{n(p,\, \mathrm{num}\,(\prod m_j))}{d} \right\rceil \leq n(p, \lambda)$.

Upper bound. By definition, $p^{n(p,\lambda)} | \lambda$. However, Lemma 9 ($\lambda$ divides least common multiple of masses) tells us that $\lambda | \operatorname{lcm}_j \left\{\mathrm{num}\left(m_j\right)\right\}$. Therefore, $p^{n(p,\lambda)} | \operatorname{lcm}_j \left\{\mathrm{num}\left(m_j\right)\right\}$. Therefore, $n(p, \lambda) \leq \max_j \left\{n(p, \mathrm{num}\left(m_j\right))\right\}$. □

Finally, we summarize the previous work into one theorem by stating that the cumulative composition function C is an Ehrhart quasi-polynomial whose period we can restrict to a small number of possibilities.

Theorem 11 (C is an Ehrhart quasi-polynomial with bounded period) *Suppose C is the cumulative composition counting function over the weighted alphabet $\widetilde{\mathcal{a}}$ with d (positive) rational masses. Then the restriction $C : \mathbb{Z}^+ \to \mathbb{R}$ is a quasi-polynomial of degree d and period λ such that* $\prod_{\{\text{prime}\, p \text{ divides } \operatorname{num}\left(\prod m_j\right)\}}$ *p divides λ. Furthermore, λ divides* $\operatorname{lcm}_i \{\operatorname{num}(m_i)\}$.

Proof Suppose that prime p divides $\operatorname{num}\left(\prod m_j\right)$. Then by Theorem 10 (Bounds on divisors of λ),

$$\begin{aligned} n(p, \lambda) &\geq \left\lceil \frac{n\left(p, \operatorname{num}\left(\prod m_j\right)\right)}{d} \right\rceil \\ &\geq \left\lceil \frac{1}{d} \right\rceil \\ &= 1. \end{aligned}$$

In other words, $p|\lambda$. Since this is true for every prime p that divides $\operatorname{num}\left(\prod m_j\right)$, $\prod_{\{\text{prime } p \text{ divides } \operatorname{num}\left(\prod m_j\right)\}}$ p divides λ, which was the first conclusion.

The second claim follows from Corollary 3 (C is an Ehrhart quasi-polynomial) and Lemma 9 (λ divides least common multiple of masses). □

Note that an Ehrhart quasi-polynomial of period 1 is a polynomial, referred to as an Ehrhart polynomial. In practice, however, the period of the Ehrhart quasi-polynomial for mass simplexes is rarely 1, as demonstrated by the following proposition and corollary.

Proposition 12 (Conditions for C being a polynomial on integers) *Suppose C is the cumulative composition counting function over $\widetilde{\mathcal{a}}$, which has rational masses. If C is a polynomial on $\mathbb{Z}^+$ then* $\operatorname{num}\left(\prod m_j\right) = 1$. *In addition, a partial converse is true: if the numerator of each mass is 1 then C is a polynomial.*

Proof C is a polynomial on $\mathbb{Z}$ if and only if its period λ is 1. Note also that λ is 1 if and only if no prime divides λ. However, by Theorem 11 (C is an Ehrhart quasi-polynomial with bounded period), $\prod_{\{\text{prime } p \text{ divides } \operatorname{num}\left(\prod m_j\right)\}}$ p divides λ. Therefore, since no prime divides λ then no prime divides $\operatorname{num}\left(\prod m_j\right)$. Thus, $\operatorname{num}\left(\prod m_j\right) = 1$.

For the second conclusion, note that if all the masses have a numerator of 1 then $\operatorname{lcm}_j \left\{\operatorname{num}\left(m_j\right)\right\} = 1$. This tells us, again by Theorem 11 (C is an Ehrhart quasi-polynomial with bounded period), that λ must be 1. In other words, C is a polynomial. □

Corollary 13 (C is not usually a polynomial on integers) *Suppose C is the cumulative composition counting function over $\tilde{\mathfrak{a}}$ with integer masses. Then C is a polynomial if and only if $m_1 = m_2 = \cdots = m_d = 1$.*

Proof If C is a polynomial then, by Proposition 12 (Conditions for C being a polynomial on integers), $\operatorname{num}\left(\prod m_j\right) = 1$. Because the masses are integers,

$$\begin{aligned}\prod m_j &= \operatorname{num}\left(\prod m_j\right)\\ &= 1.\end{aligned}$$

However, since the masses are integers, they must all be one.

Now suppose $m_1 = m_2 = \cdots = m_d = 1$. Then the numerator of each mass is one. Therefore, again by Proposition 12 (Conditions for C being a polynomial on integers), C is a polynomial. □

In other words, if we assume that we are using integer masses, C is a polynomial if and only if the integer masses are all 1. The assumption that the masses are integers is critical to this corollary, however. For example, if we have two masses, $m_1 = \frac{1}{2}$ and $m_2 = 2$, then

$$C(M) = \frac{1}{2}M^2 + \frac{3}{2}M + 1$$

which is a polynomial.

Interestingly, the Ehrhart quasi-polynomial for the two mass system of $\frac{1}{3}$ and 3 has a period of 3 on the integers, which is a counter-example of the contra-positive in the first part of Proposition 12 (Conditions for C being a polynomial on integers). It also illustrates that the product of the masses being 1 is not sufficient for C to be a polynomial.

3 Computational results

Theorem 11 (C is an Ehrhart quasi-polynomial with bounded period) describes the structure of the composition counting function. In particular, it describes a type of period and gives bounds for that period. Before we attempt to calculate this formula, however, we should check to see that the solution will be useful. By computing the lower bound of the period for a particular Ehrhart quasi-polynomial, we can state whether it is feasible to describe it using the form of Eq. (1) (Explicit form of Ehrhart quasi-polynomial).

In this section we illustrate the use of the mathematical tools we have developed by applying them to several chemical families. These families are described in the Appendix A: Chemical families.

Table 1 Advantage of Theorem 10 (Bounds on divisors of λ)

Object type	Possible period		Number of candidates		Factors of minimum period
	Minimum	Maximum	Total	Reduced	
Hydrocarbons	84	84	12	1	1
Paired DNA	381,306	381,306	16	1	1 (Relatively prime)
RNA	82,110	246,330	384	2	1, 3
DNA	4,522	614,992	240	8	1, 2, 4, 8, 17, 34, 68, 136 (Relatively prime)
Elemental	168	672	24	3	1, 2, 4
Organic	840	3,360	48	3	1, 2, 4
PEC	43,260	173,040	80	3	1, 2, 4
Amino acid	7.4×10^{25}	9.9×10^{28}	4,718,592	28	$2^{0-6} \times 3^{0-1} \times 7^{0-1}$

By using Theorem 10 the number of candidate periods is often reduced to a manageable number, as shown in the columns labeled "Number of Candidates". "Total" refers to how many numbers divide the least common multiple of the masses. "Reduced" refers to the number of possible periodicities that are available after we apply the lower bound from Theorem 10. The range of periodicities is given by the columns under "Possible Period", while "Factors of Minimum Period" describe the candidate periodicities as a multiple of the minimum possible. For example, the values 1, 2, and 4 in the "Elemental" row describe the fact that the candidate periodicities are 168×1, 168×2, and 168×4 (168, 336, and 672). There are 28 possibilities for amino acids so we describe the candidates in terms of the prime factors. This also shows how to calculate the number of candidates: there are 7 factors of 2 to consider, 2 factors of 3, and 2 factors of 7, for a total of $7 \times 2 \times 2 = 28$ possibilities.

3.1 Periodicities

We start our application of the mathematical results by calculating the period for the various collections of chemical structures.

Lemma 9 (λ divides least common multiple of masses) puts an upper limit on the number of possible periods. In particular, if we assume that we are using integer masses then the period λ divides $\operatorname{lcm}_{i=1,\ldots,d} \{m_i\}$, as represented by the Maximum column in Table 1.

In addition, Theorem 11 (C is an Ehrhart quasi-polynomial with bounded period) gives lower bounds on the number of compositions (see Table 1, the column called "Minimum possible period"). We can improve our bounds yet further by applying Theorem 10 (Bounds on divisors of λ) (from which Theorem 11 is derived)—see Table 1, column labeled "Reduced".

Even for amino acids we have reduced the number of possible periods for the Ehrhart quasi-polynomial from over 4 million possibilities to a mere 28. More importantly, we have also shown that the period is at least 7.4×10^{25}. In other words, if we store all of the coefficients of the quasi-polynomial we will need at least

$$\begin{aligned}(\#\text{terms in a polynomial}) \times (\#\text{ polynomials}) &= 21 \times 7.4 \times 10^{25} \text{ terms} \\ &= 1.4 \times 10^{27} \text{ terms.}\end{aligned}$$

The scale of this number is such that it is safe to say that we cannot expect to compute these polynomials in this way or store them as distinct polynomials.

3.2 Quasi-polynomial interpolation

One method of deriving C is to compute the actual number of compositions in order to derive a collection of terms and then compute the constituent polynomials by interpolation of the data. Suppose that C is an Ehrhart quasi-polynomial of known degree d and known period λ. Suppose also that we can calculate $C(M)$ for integer values of M such that $0 \leq M < (d+1)\lambda$. Since we are given $(d+1)$ terms for each of C's constituent polynomials, which are of degree d, we have enough terms to uniquely identify $C(M)$.

In other words, for this technique to be practical we first need the period of the polynomial (as discussed previously) and next we need to compute $(d+1)\lambda$ terms using the recurrence relationship. Before we start, however, we can check to see if this technique is feasible by calculating the minimum number of required terms $((d+1)\lambda)$.

We can also work in the opposite direction to determine the period: by attempting to determine the coefficients, we check to see if the coefficients do indeed repeat with the period which we are checking.

In order to eliminate additional period candidates we used an Excel spreadsheet, difference triangles, and the fact that we know the highest order term from Lemma 8 (Highest order term of C). While round-off error clearly starts to affect some of our calculations, Table 2 shows that we are able to identify the period for half of the chemical families. We also conjecture that if the masses are pairwise relatively prime then the period must be the maximum possible—this explains the question mark on our "answer" in the DNA row.

In addition to excluding invalid periods, the spreadsheet we created for this purpose calculated the quasi-polynomial when possible (Table 3). Due to the size of the masses in the Paired DNA and DNA problems, we were only able to compute the period by excluding the other possibilities. However, we were able to derive quasi-polynomial coefficients only for Hydrocarbon, Elemental, and Organic chemical families. Their descriptions are too long to include here. With enough computational resources we should be able to compute all of the rest except for the Amino Acid Composition chemical family. We should also note that in every case we could verify, the largest possible period was the actual period.

Was the polynomial worth finding (computationally speaking)? In other words, is it easier to simply store the answers we computed by explicitly counting compositions instead of the quasi-polynomial? The answer, is often "no". For example, for peptide masses less than 10,000 Daltons it is more efficient to store every possible exact answer than it is to store the Ehrhart quasi-polynomial in the form of a list of polynomials.

3.3 Empirical computational complexity

However, even when the period is too large to make such a solution practical it can still be possible to describe it using some other form. For example, suppose that one of the

Table 2 Calculation of Ehrhart quasi-polynomial periods

Object type	Possible period		Items to test	Tested	Answer	Comments
	Minimum	Maximum				
Hydrocarbons	84	84	None	None	84	–
Paired DNA	381,306	381,306	None	None	381,306	–
RNA	82,110	246,330	1	None	?	Our numerical experiments did not allow more than 60,000 calculations; this one test requires over 300,000 – feasible but not practical. This one test would be enough to identify period, though
DNA	4,522	614,992	7	3	614,992?	Masses are relatively prime, which may mean maximum period
Elemental	168	672	2	2	672	168 and 336 give the wrong answers for the last term of the difference pyramid
Organic	840	3,360	2	2	3,360	1,680 gives the right answer most of the time
PEC	43,260	173,040	2	None	?	Impractical
Amino acid	7.4×10^{25}	9.9×10^{28}	27	None	?	Beyond computational feasibility

We determine the period for as many chemical families as possible. Using the candidate values from Table 1 and an Excel spreadsheet designed to compute quasi-polynomials given masses and period, we eliminated candidate periodicities by comparing the derived top order term with that expected from Lemma 8 (Highest order term of C). "Items to test" lists the number of candidate periods that we would need to test to identify the period. "Tested" refers to how many periods we were able to test (eliminate) empirically. "Answer" is the actual period, if known. The phrase "Impractical" (in the last column) means that it takes more than 30,000 calculations to test the hypothesis; this was the limit of the Excel spreadsheet used to make these calculations.

coefficients is a cosine function with period 10^{100}. While the form of Eq. 1 would be prohibitively large, we could simply write the polynomial, replacing the periodic term with a cosine function. In other words, there may be a more concise way to represent the quasi-polynomial.

Verdoolaege et al. [11] usedsuch a representation for the solutions to lattice point problems: assuming that $\{\cdot\}$ is the fractional function (i.e. $\{x\} = (x \mod 1)$), then there exist integers $g_i, h_i, a_{i,j}, b_{i,j}, c_{i,j}, e_{i,j}, n_{i,j}$ such that C is given by

$$C(M) = \sum_{i=0}^{d} \frac{g_i}{h_i} M^i \prod_{j=0}^{e_i} \left\{ \frac{a_{i,j}M + c_{i,j}}{b_{i,j}} \right\}^{n_{i,j}}$$

Verdoolaege et al. also created a software library, called "barvinok", that can find the quasi-polynomial, $\mathcal{L}_{\mathcal{P}}(t)$, in polynomial time of the input (assuming a fixed

Table 3 Determination of Ehrhart Quasi-polynomial solution by interpolation

Object type	Period × dimension		Solvable?	More practical than approx?	Number of exact masses <10,000 Da	More practical than exact answer?
	Min	Max				
Hydrocarbons	168		Yes	Yes	298,453	Yes
Paired DNA	762,612		Hard	No	153	No
RNA	246,330	738,990	Hard	No	50,397	No
DNA	3.2 million		Hard	No	58,446	No
Elemental	3,360		Yes	Yes	1×10^{13}	Yes
Organic	16,800		Yes	No	7×10^{11}	Yes
PEC	259,560	1,038,240	Hard	No	2×10^{11}	Yes
Amino acid	1.6×10^{27}	2.1×10^{30}	No	No	2×10^{11}	No

The storage requirements for the simplistic form of the Ehrhart quasi-polynomial is its period times its dimension +1. If this has few enough terms then we can calculate the coefficients by using the recurrence relation and solving the resulting linear equations. "Solvable?" refers to the question of whether these linear equations can be solved and, thus, we can determine the quasi-polynomial. "Hard" refers to the cases where there are more than 30,000 terms. "More Practical than approx?" compares the usefulness of the Ehrhart quasi-polynomial to using the recursive formula to integer masses. "Number of Exact Masses $< 10,000$ Da" provides a count of the number of chemical compositions of this family with masses less than 10,000 Da, as computed by the recurrence relation. "More Practical than exact answer?" compares the usefulness of the Ehrhart quasi-polynomial to using the recursive formula on exact masses (using elemental composition as an exact integer vector).

Table 4 Complexity of computing quasi-polynomials

	Input		Output			
Object type	Dimensions	File size (bytes)	File size (bytes)	Elapsed time (s)	Number of fractionals	Number of terms
Hydrocarbons	2	141	329	0.00	3	11
Paired DNA	2	143	225	0.00	2	7
RNA	4	193	26,552	0.12	33	603
DNA	4	193	36,843	0.19	46	873
Elemental	5	219	5,764	0.09	9	170
Organic	5	225	20,657	0.13	17	511
PEC	5	226	45,357	0.41	30	2022

This table shows the input complexity (number of masses and file size) and output complexity (output file size, elapsed time to completion, number of fractionals, and number of terms).

dimension d). Note that fixing the dimension is essential to this proof. In fact, they also show that the solution can be exponential in d.

We applied this program to our collection of problems to see if the more compact form of Ehrhart quasi-polynomials could solve the problems that were too complex to list otherwise. Indeed, this was the case for all except amino acids (see Table 4). The relatively small number of chemical species forming the basis allowed all of these

Table 5 Complexity of computing quasi-polynomials—amino acids

Input		Output			
# Amino acids	File size (bytes)	File size (bytes)	Elapsed time (s)	Number of fractionals	Number of terms
1	120	84	0.00	1	3
2	139	402	0.00	4	13
3	162	5,137	0.01	17	136
4	189	35,009	0.18	49	892
5	220	172,239	1.25	82	4,059
6	256	580,272	6.51	122	11,914
7	297	2,779,734	29.8		
8	341	7,095,919	116		
9	390	13,369,407	325		
10	442	43,376,230	1,550		
11	498	130,837,897	6,120		
12	558	334,610,768	26,280		
13	622	511,761,390	94,320		
14	690	1,102,543,683	348,000		
15	762	3,557,762,180	1,760,000		
16	838				
17	918				
18	1,002				
19	1,090				
20	1,182				

Here we describe the complexity of using one or more amino acid masses when computing Ehrhart quasi-polynomials using the software library barvinok. This table has the same specifications as Table 4. The missing values in the output file size reflect our inability to run the experiment due resource requirements - the quasi-polynomial for 15 amino acids took 20 days to derive on a Sun Workstation and we expected the next to take two months. Missing data in the number of fractionals and number of terms were due to the difficulty of parsing the output files into a database—the current parser ran out of memory when processing 7 amino acids.

items (together) to be calculated in under a second on an iMac (3.06 GHz Intel Core 2 Duo, 4 GB RAM).

Unfortunately, the same could not be said for the amino acid problem (Table 5); we were unable to compute the Ehrhart quasi-polynomial for 20 amino acids. Indeed, it took 20 days elapsed time to derive the Ehrhart quasi-polynomial for 15 amino acids. Nonetheless, we can project the resources it would take to compute the Ehrhart quasi-polynomial for all 20 amino acids.

In Fig. 1 we see a strong exponential relationship between the number of masses (amino acids) and the time it took to derive the corresponding Ehrhart quasi-polynomial. If we use the resulting formula to project out to 20 amino acids, we expect that it would take approximately 40 years to compute the Ehrhart quasi-polynomial for 20 amino acids.

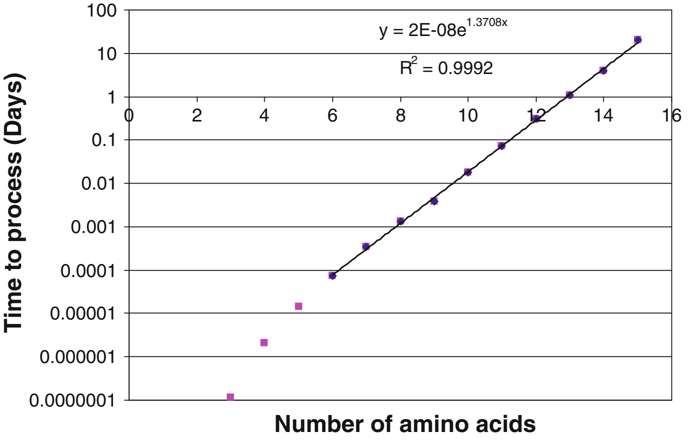

Fig. 1 Time to compute quasi-polynomial—`barvinok`. The equation at the top of the figure describes the exponential regression model for amino acid calculations (for 6–15 amino acids). We do not have data beyond 15 amino acids due to computational resource limitations (time and hard drive space). Using less than 6 amino acids in the regression degraded its performance. We suspect that the program required the use of virtual memory beyond 6 amino acids, degrading its performance

Table 6 Expected resources for the Ehrhart quasi-polynomial representing amino acid compositions

Statistic	Method	Calculation	Estimate at 20 amino acids
Run time	AA -> Runtime	$2 \times 10^{-8} \times 3.94^{AA}$ (days)	40 years
File size of output	AA ->Filesize	0.0059×2.422^{AA}(MB)	286 TB
Number of terms	AA ->Filesize -> #terms	$0.0466 \left(0.0059 \times 2.422^{AA}\right)^{0.9386}$	2.6 million
Number of fractionals	AA -> #fractionals	$0.8214 \times AA^{2.8123}$	3,744
Number of computations	AA ->Filesize -> #computations	$49282 \left(0.0059 \times 2.422^{AA}\right) + 57.444$ $= 290 \times 2.422^{AA} + 57.444$	14 billion

Note that this table assumes that we are using integer masses for the amino acids

By modeling the data observed in Table 5 we were able to estimate resource use for solving the amino acid problem. We summarize our projections to 20 amino acids in Fig. 3. This table also shows the modeling equations. The conclusion is that, while it would be feasible to calculate this Ehrhart quasi-polynomial, given a very long time and what would be considered an astronomical amount of resources at present, its utility would be extremely limited: with 2.6 million terms, it would be better to explicitly count the number of compositions and store all of the values of interest (Table 6).

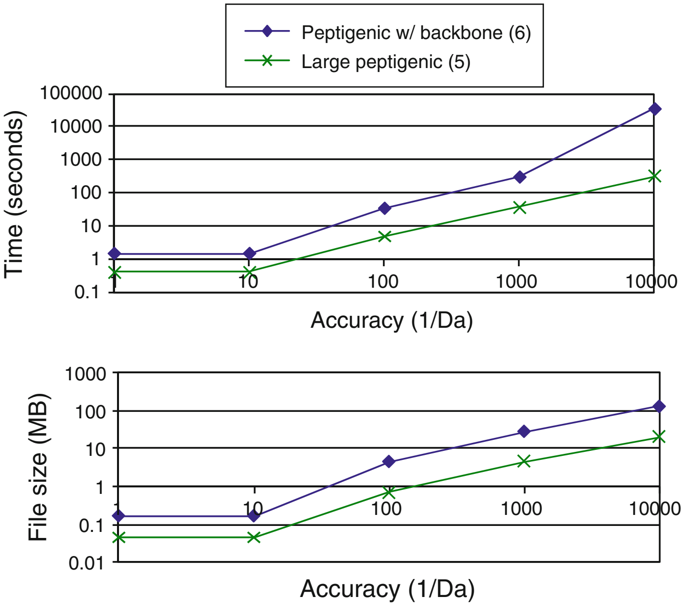

Fig. 2 Increased complexity of higher accuracy—barvinok. We compare the different levels of accuracy for four different experiments, measuring the effects of increased accuracy on time to completion (*top*) and output file size (*bottom*). "Peptigenic w/ backbone" is similar to PEC except that we add another object (the backbone of the peptide) that is common to every amino acid residue. This also means that the sulfur-containing object must be altered so that it does not include the backbone. "Large peptigenic" is the same as PEC

3.4 Effects of accuracy

So far, we have only applied barvinok to integer masses. Ideally, however, we would like to use it for high accuracy calculations. Therefore, we need to determine how complexity changes when we change mass accuracy.

We analyzed accuracy performance on three of the chemical families and one derived from the others (see Fig. 2). They appear to have similar behavior across the five scales that we measured (1, 0.1 , 0.01, 0.001, and 0.0001 Da) but we cannot draw any quantitative conclusions from this data.

Figure 3, on the other hand, suggests that, at least for the PEC data, accuracy is inversely related to both time to completion and file size. In other words, if we want accuracy to move from 1 to 0.01 Daltons, it will take 100 times longer and use 100 times more space. This matches our intuition regarding the effects of adding accuracy requirements.

4 Discussion

We have presented theoretical and empirical methods forsolving the composition problem, i.e., the problem of calculating the number of chemical compositions with a spe-

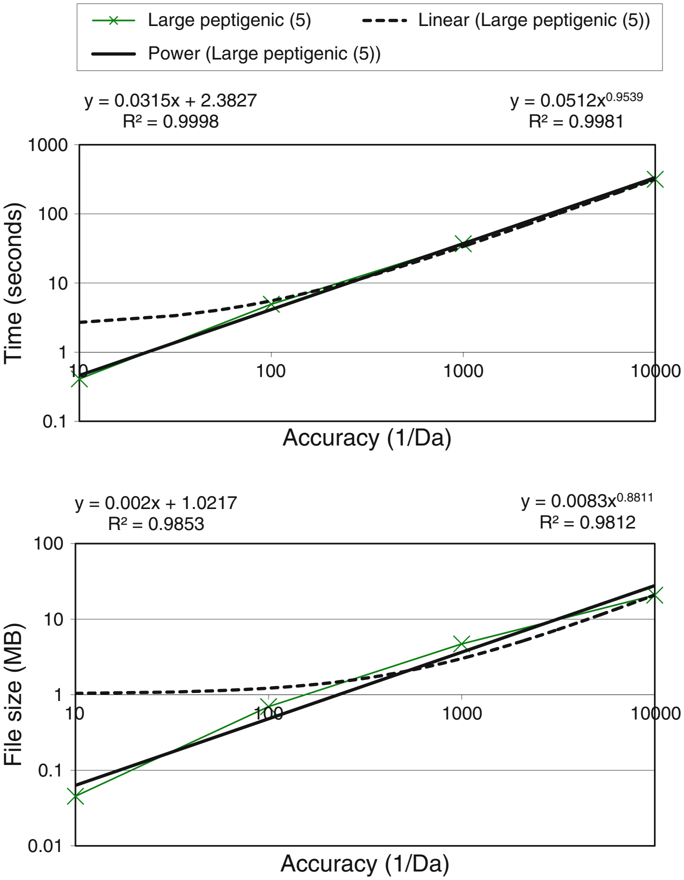

Fig. 3 Quantitative analysis of accuracy complexity—`barvinok`. Analysis was performed on the PEC data. While a power law matched the data better (*solid line*), a linear relationship also modeled the data well (*dotted line*)

cific mass. We introduced the relationship between counting compositions and Ehrhart quasi-polynomials, a relationship which derives from a geometric interpretation of the composition problem. In addition we provide strong restrictions on the period of the corresponding Ehrhart quasi-polynomial, greatly reducing the number of periods that need to be tested for this class of problems. We also applied two different methods to compute Ehrhart quasi-polynomials for seven biologically relevant chemical families (e.g. RNA, DNA, hydrocarbons, organic molecules) and showed, theoretically and empirically, that neither of these approaches are practical or even desirable for an eighth chemical family, theoretical peptides (amino acid compositions).

In addition, period information is provided by both Zaslavsky's Theorem [10,12], describing the period of the convolution of two quasi-polynomials, and McMullen's Theorem [13], describing the period of the individual quasi-polynomial coefficients. There is even a polynomial time algorithm for calculating the period. However, apart from providing methods for calculating period (which can be difficult), none of these theorems provide a simple lower limit to the period [9]. As we described in Sect. 2, Theorem 10 (Bounds on divisors of λ) provides a simple lower bound on period, albeit on a restricted subset of polytopes, mass simplexes. This theorem also restricts the number of candidate periods, greatly simplifying the process of checking individual periods. We also conjecture that this result can be extended to all simplexes and, thus, to all polytopes. It is interesting to note that the lower limit that we provide will often be the same as the upper limit; others have noted that the upper limit is often the actual period [8].

We tested the utility of our new bounds on the period by applying them to eight biologically relevant chemical composition problems. In every case the lower bound provided important information about the problem, either solving it explicitly or making it easy to prove that finding the solution was not feasible and not practical even if found. In particular, the space required to store the formula for the number of amino acid compositions of integer mass was much larger than the amount of storage needed to store the specific counts for each unit of mass up to the size of the largest protein encountered in nature. Thus, even when identifying the quasi-polynomial was not practical, the ability to restrict the candidate list of periods was useful.

There have been important efforts on understanding the computational complexity of finding the Ehrhart quasi-polynomial specified by the number of lattice points in a polytope. While Barvinok showed that the problem is equivalent to the Knapsack Problem (which is NP-complete), he also was able to describe a polynomial time algorithm if one restricts the polytope in question to a simplex of fixed dimension [14]. Verdoolaege et al. extended this result to any polytope of fixed dimension, combined it with several other theoretical results [5,15] and implemented the algorithm,calling the software library `barvinok` [11]. They also mention that, from a practical point of view, problems with more than six dimensions are often difficult to solve, which matches our experiences as well; `barvinok` performed well on all of our chemical problems except the amino acid composition problem, which is 20-dimensional.

Much of our analysis assumes that we are dealing with *integer* masses. The field of mass spectrometry, however, rarely uses integer masses anymore and it is well understood that high mass accuracy is required (though not sufficient) to properly identify an amino acid composition [16,17]. We investigated the use of `barvinok` on problems with different mass accuracies. As expected, as accuracy increased, so did the size of the solution; `barvinok` uses exact solutions (computing with fractions with an arbitrary number of digits). For elemental compositions (five or six elements), it appears that the formula for 10^{-5} Dalton mass accuracy is practical, although large.

Thus we have shown that the actual formula for the number of elemental compositions of a given mass could be computed (provided the number of elements is small). Such an algorithm may be of value to some identification scoring algorithms [18]. In addition, while we have used a mass simplex throughout this paper, some ad hoc rules (see [16]) could be defined in terms of a different polytope.

Acknowledgments We would like to thank Josh Coon for useful discussions and comments and Sven Verdoolaege for technical assistance with the `barvinok` software library.

Appendix A: Chemical families

In this appendix we describe the chemical families that are used in in this paper. First we describe why we may want to explore the mass distribution of the family and then describe it in detail, along with any assumptions that had to be made to define the family.

A.1 Hydrocarbons

The simplest infinite family of compositions is that of hydrocarbons - molecules made up of hydrogen and carbon. Mass spectrometry of hydrocarbons is a major area of oil research.

Object name	Chemical composition	Mass
Carbon	C	12
Hydrogen surrogate	CH_2	14.01565006

Note that there is no non-radical hydrocarbon made of an odd number of hydrogen atoms, a fact that is a special case of one of Senior's Rules [19–21]. Also, for this family we expect at least one C for each pair of hydrogens after the first pair. Thus, we can use CH_2 as a surrogate for H. This means that we can represent all hydrocarbons as $H_2(CH_2)_aC_b$, although this is not standard notation.

The integer mass solution to this is fairly simple since we need only look at the least common multiple of 12 and 14 to evaluate combinations involving both the hydrogen surrogate and carbon.

A.2 Double-stranded DNA

Similar to Hydrocarbons, this is a simple collection of two compositional components, a DNA strand with its complement. Interestingly, the two components differ by only one Dalton, arrived by replacing CH with N. This is a nice illustration of one of Senior's rules—an even mass number corresponds to an even number of nitrogen atoms (at least for organic molecules composed of C, H, N, O, or S).

Mathematically this is a very interesting case because the difference between this is so small relative to their total mass. We can represent their masses as m and $m + \Delta m$. If we select the units so that m is 1 then Δm is 0.001612774, which gives us an interesting distribution on the accurate masses. In particular, we must have 620 base-pairs before we have an overlap of distributions (i.e., the distribution of masses of 620 base-pairs overlaps that of 621 base-pairs).

Object name	Chemical composition	Mass
A=T	$C_{20}H_{25}N_7O_{12}P_2$	617.10364
G=C	$C_{19}H_{24}N_8O_{12}P_2$	618.09889

A.3 RNA

RNA and DNA have several interesting properties. In particular, there are only four of the building blocks and they are linearly independent (mass uniquely identifies which components are present). They also have a remarkably small mass defect, meaning that they are very close to an integer value, about $1/10^{\text{th}}$ that of proteins, per mass.

Object name	Chemical composition	Mass
A	$C_{10}H_{12}N_5O_6P$	329.05252
C	$C_9H_{12}N_3O_7P$	305.04128
G	$C_{10}H_{12}N_5O_7P$	345.04743
U	$C_9H_{11}N_2O_8P$	306.02530

A.4 DNA

Object name	Chemical composition	Mass
A	$C_{10}H_{12}N_5O_5P$	313.05761
C	$C_9H_{12}N_3O_6P$	289.04637
G	$C_{10}H_{12}N_5O_6P$	329.05252
T	$C_{10}H_{13}N_2O_7P$	304.04603

A.5 Elemental

We may want to use a collection of elements (say, C, H, N, O, and S), considering all possible compositions.

Object name	Chemical composition	Mass
Carbon	C	12
Hydrogen	H	1.0078250321
Nitrogen	N	14.0030740048
Oxygen	O	15.99491461956
Sulfur	S	31.972071001

A.6 Organic

The Elemental chemical family is actually quite primitive, in the sense that it includes many impossible compositions. For example, H_3 is included but it is not possible.

On the other hand, we can combine certain parts to ensure that the resulting molecule is reasonable. For example, Senior's Rules applies to the above list of elements: the sum of nitrogens and hydrogens must be even. This implies that every N is accompanied by an H and, when there are insufficient nitrogen atoms to cover the hydrogens, the remainder must be lumped by twos with a C.

(These rules satisfy many of the "Seven Golden Rules of Mass Spectrometry", as given in [20].)

Object name	Chemical composition	Mass
Carbon	C	12
Hydrogen part	CH_2	14.01565006
Nitrogen part	NH	15.010899
Oxygen	O	15.99491461956
Sulfur	S	31.972071001

A.7 Peptigenic elemental composition (PEC)

The designation of "peptigenic elemental composition" refers to elemental compositions (i.e. the counts of particular atoms within a molecule) that can be generated from amino acid compositions. Similar to the Organic chemical family, we group elements into components that are commonly found in an amino acid and, thus, in a peptide or protein. Note that, while we call this peptigenic, there are many combinations of these objects which do not correspond to molecules formed of amino acids. For example, every non-sulfuric amino acid includes one each of Carbon, Hydrogen Part, Nitrogen Part, and Oxygen part (elemental composition ONC_3H_5). This composition of parts is the peptide backbone.

In spite of this caveat, the number of compositions using these units approximates the number of peptigenic elemental compositions well for larger masses.

Object name	Chemical composition	Mass
Carbon	C	12
Hydrogen part	CH_2	14.01565006
Nitrogen part	NH	15.010899
Oxygen	O	15.99491461956
Sulfur part (Cysteine)	$SONC_3H_5$	103.0091848

A.8 Amino acid composition (AAC)

In mass spectrometry it can be useful to know how many amino acid compositions are in a particular mass range.These represent amino acid residues in a protein; i.e. these are the contributions of any particular amino acid to a protein.

Note also that this is the only chemical family represented in this paper whose components are not linearly independent. For example, two Glycines have the same chemical composition as one Asparagine (these are isomers).

Object name	Chemical composition	Mass
Glycine	C_2H_3ON	57.021464
Alanine	C_3H_5ON	71.037114
Serine	$C_3H_5O_2N$	87.032028
Proline	C_5H_7ON	97.052764
Valine	C_5H_9ON	99.068414
Threonine	$C_4H_7O_2N$	101.047678
Cysteine	C_3H_5ONS	103.009185
Iso-leucine	$C_6H_{11}ON$	113.084064
Leucine	$C_6H_{11}ON$	113.084064
Asparagine	$C_4H_6O_2N_2$	114.042928
Aspartic acid	$C_4H_5O_3N$	115.026943
Glutamine	$C_5H_8O_2N_2$	128.058578
Lysine	$C_6H_{12}ON_2$	128.094963
Glutamic acid	$C_5H_7O_3N$	129.042593

Object name	Chemical composition	Mass
Methionine	C_5H_9ONS	131.040485
Histidine	$C_6H_7ON_3$	137.058912
Phenylalanine	C_9H_9ON	147.068414
Arginine	$C_6H_{12}ON_4$	156.101111
Tyrosine	$C_9H_9O_2N$	163.063329
Tryptophan	$C_{11}H_{10}ON_2$	186.079313

References

1. S. Hubler, G. Craciun, Mass distributions of linear chain polymers. J. Math. Chem. (2012). doi: 10.1007/s10910-012-9983-z
2. S.L. Hubler, G. Craciun, Periodic patterns in distributions of peptide masses. BioSystems (2012). http://dx.doi.org/10.1016/j.biosystems.2012.04.008
3. S.L. Hubler, *Mathematical Analysis of Mass Spectrometry Data*, PhD Thesis, in Mathematics. (University of Wisconsin-Madison, Madison, WI, 2010)
4. S. Verdoolaege, *Barvinok: User Guide*. (2011) September 7, 2011 [cited 2011 12/1/2011]; barvinok-0.34-26-g2523709: [Available from: http://www.kotnet.org/~skimo/barvinok/barvinok.pdf].
5. A.I. Barvinok, J.E. Pommersheim, in *An algorithmic theory of lattice points in polyhedra*, ed. by L.J. Billera New Perspectives in Geometric Combinatorics. (MSRI Publications, Cambridge, 1999)
6. E. Ehrhart, Sur les polyèdres rationnels homothétiques à n dimensions. C. R. Acad. Sci. Paris **254**, 616–618 (1962)
7. M. Beck, S. Robins, in *Computing the continuous discretely—integer-point enumeration of polyhedra, 1st edn*, ed. by S. Axler, K.A. Ribet Undergraduate Texts in Mathematics (Springer, New York, 2007), p. 250
8. T.B. McAllister, K.M. Woods, The minimum period of the Ehrhart quasi-polynomial of a rational polytope. J. Combinatorial Theory A **109**(2), 345–352 (2005)
9. K. Woods, Computing the period of an Ehrhart quasi-polynomial. Electron. J. Combinatorics **12**(1), (2005)
10. T. Zaslavsky, Periodicity in quasipolynomial convolution. Electronic Journal of Combinatorics **11**(2), (2004)
11. S. Verdoolaege et al., *Computation and Manipulation of Enumerators of Integer Projections of Parametric Polytopes* (Department of Computer Science, K.U. Leuven, 2005), p. 103
12. M. Beck, S.V. Sam, K.M. Woods, Maximal periods of (Ehrhart) quasi-polynomials. J. Combinatorial Theory A **115**(3), 517–525 (2008)
13. P. Mcmullen, Lattice invariant valuations on rational polytopes. Archiv. Der. Mathematik **31**(5), 509–516 (1978)
14. A.I. Barvinok, Computing the Ehrhart quasi-polynomial of a rational simplex. Math. Comput. **75**, 1449–1466 (2006)
15. P. Clauss, V. Loechner, Parametric analysis of polyhedral iteration spaces. J. VLSI Signal Process. Syst. **19**(2), 179–194 (1998)
16. B. Spengler, Accurate mass as a bioinformatic parameter in data-to-knowledge conversion: Fourier transform ion cyclotron resonance mass spectrometry for peptide de novo sequencing. Eur. J. Mass Spectrom. **13**(1), 83–87 (2007)
17. S.L. Hubler et al., Valence parity renders z(center dot)-type ions chemically distinct. J. Am. Chem. Soc. **130**(20), 6388–6394 (2008)
18. G. Alves, Y.-K. Yu, Statistical characterization of a 1D random potential problem–With applications in score statistics of MS-based peptide sequencing. Phys. A Stat. Mech. Appl. **387**(26), 6538–6544 (2008)
19. J. Meija, Mathematical tools in analytical mass spectrometry. Anal. Bioanal. Chem. **385**(3), 486–499 (2006)
20. T. Kind, O. Fiehn, Seven golden rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry. BMC Bioinform **8**, 105–124 (2007)
21. J.K. Senior, Partitions and their representative graphs. Am. J. Math. **73**(3), 663–689 (1951)